

Greenplum Database 4.3.6.1 Release Notes

Rev: A03

Updated: January, 2017

Welcome to Pivotal Greenplum Database 4.3.6.1

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.

Warning: Do not upgrade from Greenplum Database 4.2.x to Greenplum Database 4.3.6.1. When upgrading from Greenplum Database 4.2.x, upgrade to Greenplum Database 4.3.6.2 or later. When Greenplum Database 4.2.x is configured with mirrors, a known `gpmigrator_mirror` issue prevents the successful upgrade of a Greenplum Database system to 4.3.6.1.

For information about the issue, see the Pivotal Knowledge Base article [gpmigrator_mirror fails with ERROR: pg_controldata: "Database cluster state: shutting down"](#)

Note: This document contains pertinent release information about Greenplum Database 4.3.6.1. 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](#).

About Greenplum Database 4.3.6.1

Greenplum Database 4.3.6.1 is a patch release that resolves known issues, and introduces a number of significant new features, as well as performance and stability enhancements. Please refer to the following sections for more information about this release.

- [Product Enhancements](#)
- [Changed and Deprecated Features](#)
- [Supported Platforms](#)
- [Resolved Issues in Greenplum Database 4.3.6.x](#)
- [Known Issues in Greenplum Database 4.3.6.x](#)
- [Upgrading to Greenplum Database 4.3.6.x](#)
- [Greenplum Database Tools Compatibility](#)
- [Greenplum Database Extensions Compatibility](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.3.6.1 Documentation](#)

Product Enhancements

Greenplum Database 4.3.6.1 includes enhancements in these areas:

- [External Table Support for Avro and Parquet File Formats on HDFS](#)

- *Partitioned Tables Support for External Tables as a Partition*
- *Connection Pooling for Client Connections*
- *Managing COPY Commands with Resource Queues*
- *Pivotal Query Optimizer*
- *Analyzing Greenplum Database Catalog Tables*
- *Transferring Database Objects*
- *Incremental Resynchronization Process Performance*
- *Migration and Mirror Migration*
- *gpsupport Utility for Troubleshooting and Log Collection*

External Table Support for Avro and Parquet File Formats on HDFS

The Greenplum Database external table protocol `gp_hdfs` supports Avro and Parquet file formats on a Hadoop file system (HDFS). You can create a Greenplum Database external table that uses the `gp_hdfs` protocol to read and write Avro or Parquet format files on HDFS.

- An *Avro* file stores both the data definition (schema) and the data together in one file making it easy for programs to dynamically understand the information stored in an Avro file. The Avro schema is in JSON format, the data is in a binary format making it compact and efficient.
- A *Parquet* file is designed to take advantage of compressed, efficient columnar data representation available to projects in the Hadoop ecosystem. Parquet supports complex nested data structures and uses Dremel record shredding and assembly algorithms. Parquet supports very efficient compression and encoding schemes. Parquet allows compression schemes to be specified on a per-column level, and supports adding more encodings as they are invented and implemented.

For information about the changes to the `CREATE EXTERNAL TABLE` command, see [CREATE EXTERNAL TABLE Command Changes](#)

For information about Greenplum Database support for the Avro and Parquet file formats, see the *Greenplum Database Administrator Guide*. For information about the `CREATE EXTERNAL TABLE` command, see the *Greenplum Database Reference Guide*.

Partitioned Tables Support for External Tables as a Partition

For a Greenplum Database partitioned table, you can exchange a leaf child partition of the partitioned table with a readable external table. The external table data can reside on a host file system, an NFS mount, or a Hadoop file system (HDFS).

For example, if you have a partitioned table that is created with monthly partitions and most of the queries against the table only access the newer data, you can copy the older, less accessed data to external tables and exchange older partitions with the external tables. For queries that only access the newer data, you could create queries that use partition elimination to prevent scanning the older, unneeded partitions.

Exchanging a leaf child partition with an external table is not supported in these cases:

- The partitioned table is created with the `SUBPARTITION` clause or if a partition has a subpartition.
- The partitioned table contains a column with a check constraint or a `NOT NULL` constraint.

For information about changes to the `ALTER TABLE` command to support external table partitions, see [ALTER TABLE Command Changes](#)

For information about exchanging a leaf child partition of the partitioned table with a readable external table, see the *Greenplum Database Administrator Guide*. For information about the `ALTER TABLE` command, see the *Greenplum Database Reference Guide*.

Connection Pooling for Client Connections

The Greenplum Database installation includes the PgBouncer 1.6.1 connection pooler to manage external connections to the Greenplum Database master. The PgBouncer utility controls connections to the

Greenplum Database master and maintains a cache of database connections so that the connections can be reused when future requests to the database are required.

You can use the `stunnel` utility to encrypt connections to the Greenplum Database master. The `stunnel` utility uses OpenSSL to encrypt client connections when using PgBouncer connection pooler to connect to the Greenplum Database.

For information about using PgBouncer and `stunnel`, see *Greenplum Database Administration Guide*. For PgBouncer reference information see the *Greenplum Database Utility Guide*. For `stunnel` reference information, see the `stunnel` documentation at <https://www.stunnel.org>.

Managing COPY Commands with Resource Queues

When a Greenplum Database user who is not a superuser runs a `COPY` command, the command can be controlled by a resource queue. The resource queue must be configured with the `ACTIVE_STATEMENTS` parameter that specifies a maximum limit on the number of queries that can be executed by roles assigned to that queue. Greenplum Database does not apply a cost value or memory value to a `COPY` command, resource queues with only cost or memory limits do not affect the running of `COPY` commands.

A non-superuser can run these types of `COPY` commands:

- `COPY FROM` command where the source is `stdin`
- `COPY TO` command where the destination is `stdout`

For information about the `COPY` command, see the *Greenplum Database Reference Guide*. For information about the resource queues, see the *Greenplum Database Administrator Guide*.

Pivotal Query Optimizer

For queries against partitioned tables, the Pivotal Query Optimizer supports uniform multi-level partitioned tables. A multi-level partitioned table is a partitioned table that was created with the `SUBPARTITION` clause. A uniform partitioned table must meet these requirements.

- The partitioned table structure is uniform. Each partition node at the same level must have the same hierarchical structure.
- The partition key constraints must be consistent and uniform. At each subpartition level, the sets of constraints on the child tables created for each branch must match.

For information about uniform multi-level partitioned tables and the Pivotal Query Optimizer, see the *Greenplum Database Administrator Guide*.

Analyzing Greenplum Database Catalog Tables

The Greenplum Database `analyzedb` utility can gather statistics on catalog tables. If you do not specify a table, set of tables, or schema, the utility collects the statistics as needed on all system catalog tables and user-defined tables in the database.

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

Transferring Database Objects

The Greenplum Database `gptransfer` utility has been enhanced. Enhancement include:

- New options for to make `gptransfer` more flexible
- Enhanced logging during the transfer process
- Ability to use regular expressions to specify database and file names

For information about the changes to the `gptransfer` utility, see *gptransfer Utility Changes*.

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

Incremental Resynchronization Process Performance

Greenplum Database performance of full and incremental resynchronization of segment mirrors has been improved. The Greenplum Database server configuration parameter `filerep_mirrorvalidation_during_resync` controls file validation during incremental resynchronization of segment mirrors.

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

Migration and Mirror Migration

If you have performed a catalog check with `gpmigrator` or `gpmigrator_mirror` utility `--check-only` option in preparation for migrating from Greenplum Database 4.2.x, you can specify the `--skip-check` option to skip the catalog check during the migration.

For information the changes to the utilities, see *gpmigrator and gpmigrator_mirror Utilities Changes*

For information about the `gpmigrator` and `gpmigrator_mirror` utilities, see the *Greenplum Database Utility Guide*.

gpsupport Utility for Troubleshooting and Log Collection

You can use the `gpsupport` utility to collect Greenplum Database information when troubleshooting and diagnosing issues with Greenplum Database installations. The utility `gpsupport v1.2` includes these enhancements:

- Specify a single date in the format `YYYY-MM-DD`. For any commands that operate on a specific date range, you can specify a single date with `singleDate`.
- The `diagnose system` command determines the platform on which you are running Greenplum Database and validates various platform-specific configuration settings. This command is similar to the Greenplum Database `gpcheck` utility.

For information about the `gpsupport` utility, see the documentation for the *gpsupport utility*.

Changed and Deprecated Features

- *Changed Features in Greenplum Database 4.3.6.1*
- *New Parameters*
- *Deprecated Features*

Changed Features in Greenplum Database 4.3.6.1

These feature changes have been introduced in Greenplum Database 4.3.6.1:

- *ALTER TABLE Command Changes*
- *CREATE EXTERNAL TABLE Command Changes*
- *EXPLAIN Plan Output Changes*
- *analyzedb Utility Changes*
- *gpdrestore Utility Changes*
- *gpinitstandby Utility Changes*
- *gpmigrator and gpmigrator_mirror Utilities Changes*
- *gptransfer Utility Changes*
- *JDBC Drivers Added to Connectivity Tools*
- *Greenplum Database PL/Perl Extension Change*

ALTER TABLE Command Changes

- The `ALTER TABLE` command supports the `EXCHANGE PARTITION` clause to exchange a readable external table into a partitioned table hierarchy in place of an existing leaf child partition. You must also specify the `WITHOUT VALIDATION` clause to skip table validation against the `CHECK` constraint of the partition you are exchanging.

```
EXCHANGE PARTITION { partition_name | FOR (RANK(number)) |
  FOR (value) } WITH TABLE table_name WITHOUT VALIDATION
```

Warning: When you specify the `WITHOUT VALIDATION` clause, you must ensure that the data in table that you are exchanging for an existing child leaf partition is valid against the `CHECK` constraints on the partition. Otherwise, queries against the partitioned table might return incorrect results.

Exchanging a leaf child partition with an external table is not supported in these cases:

- The partitioned table is created with the `SUBPARTITION` clause or if a partition has a subpartition.
- The partitioned table contains a column with a check constraint or a `NOT NULL` constraint.
- The `ALTER TABLE` command no longer supports the `EXCHANGE DEFAULT PARTITION` clause to exchange the default partition of a partitioned table with an existing Greenplum Database table.

For information about `ALTER TABLE` command, see the *Greenplum Database Reference Guide*.

CREATE EXTERNAL TABLE Command Changes

The `FORMAT` clause of the `CREATE EXTERNAL TABLE` command to create a readable or writable external table supports the values `AVRO` and `PARQUET` for Avro or Parquet format files on HDFS. The `gphdfs` protocol must be specified to create external tables that read or write Avro or Parquet format files. For example, this is the syntax for a readable external table.

```
CREATE EXTERNAL TABLE tablename (column_spec) LOCATION
  ('gphdfs://location') FORMAT 'AVRO'
```

For information about `CREATE EXTERNAL TABLE`, see the *Greenplum Database Reference Guide*.

EXPLAIN Plan Output Changes

The Greenplum Database `EXPLAIN` command displays the setting of the server configuration parameter `OPTIMIZER` for a query and whether the Pivotal Query Optimizer or the legacy query optimizer generated the explain plan.

- When the Pivotal Query Optimizer generates the query plan, the setting `optimizer=on` and the Pivotal Query Optimizer version are displayed at the end of the query plan. For example.

```
Settings: optimizer=on
Optimizer status: PQO version 1.584
```

- When Greenplum Database falls back to the legacy optimizer to generate the plan, the setting `optimizer=on` and `legacy query planner` are displayed at the end of the query plan. For example.

```
Settings: optimizer=on
Optimizer status: legacy query planner
```

- When the server configuration parameter `OPTIMIZER` is `off`, these lines are displayed at the end of a query plan.

```
Settings: optimizer=off
```

Optimizer status: **legacy query planner**

For information about EXPLAIN plan output, see the *Greenplum Database Administrator Guide*.

analyzedb Utility Changes

The Greenplum Database `analyzedb` utility supports analyzing system catalog tables. You can specify the system catalog table to analyze. If you do not specify a table, set of tables, or schema, the utility collects the statistics as needed on all system catalog tables and user-defined tables in the database.

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

gpdbretrieve Utility Changes

The Greenplum Database `gpdbretrieve` utility `-T` option supports restoring schemas from a backup created with `gpcrondump`.

To restore the tables for a specific schema, you can specify the schema name with the `*` wildcard character. For example `-T mytest.*` restores the tables for the schema `mytest`. System catalog schemas are not supported.

If you specify the `--truncate` option with the `-t schema.*` option, all existing tables within the database schema are truncated before tables are restored from the backup.

gpinitstandby Utility Changes

If the Greenplum Database `gpinitstandby` utility fails to initialize the standby master, the standby master data directory is not cleaned up after the failure because it contains log files that can help in determining the reason for the failure. If an initialization failure occurs, a summary report file is generated in the standby host directory `/tmp`. The report file lists the directories on standby host that require clean up.

If the Greenplum Database `gpinitstandby` utility fails to initialize the standby master, you must delete the files in the standby master data directory before running `gpinitstandby` again.

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

gpmigrator and gpmigrator_mirror Utilities Changes

When migrating from Greenplum Database 4.2.x, the Greenplum Database `gpmigrator` and `gpmigrator_mirror` utilities have been enhanced to improve migration. The `--skip-check` option has been added to permit you to skip the catalog check portion of the migration process.

Pivotal recommends performing a database catalog check and fixing catalog issues before migrating Greenplum Database. By default, the `gpmigrator` and `gpmigrator_mirror` utilities perform a catalog check as part of the migration process. Performing a catalog check with the Greenplum Database `gpcheckcat` utility before starting your migration and then invoking `gpmigrator` or `gpmigrator_mirror` is recommended. As an alternative, you can perform a catalog check with `gpmigrator` or `gpmigrator_mirror` utility option `--check-only`. After you have performed a catalog check, you can specify the `--skip-check` option to skip the catalog check during migration with `gpmigrator` or `gpmigrator_mirror`.

For information about the `gpmigrator` and `gpmigrator_mirror` utilities, see the *Greenplum Database Utility Guide*.

gptransfer Utility Changes

These changes have been made to the Greenplum Database `gptransfer` utility:

- The CSV (comma separated values) format is default file format for the external tables that `gptransfer` creates to copy data between databases.
- The `-d`, `-t` and `-f` options to specify tables to transfer are mutually exclusive.

- For a list of files being transferred, `gptransfer` starts the transfer in the order listed.
- The `gptransfer` logging has been enhanced:
 - Status of tables being copied is displayed during the transfer process.
 - Copied table names are logged.
 - The number of tables copied is displayed and logged
- The `gptransfer` utility supports these options.
 - The `--delimiter` option lets you specify a single ASCII character that separates columns within each row of data. Delimiter to use for writable external tables created by `gptransfer`.
 - The `--format` option lets you specify the format of the writable external tables that are created by `gptransfer` to transfer data. You can specify `CSV` for comma separated values, or `TEXT` for plain text. The default value is `CSV`.
 - The `--quote` option lets you specify the quotation character for writable external tables, when `gptransfer` creates writable external tables with the `CSV` format.
 - The `-F` and `-T` options let you specify tables from the source database system to exclude from transfer. The `-F` option lets you specify a file containing list of fully qualified table names. The `-T` option lets you specify a list of fully qualified table names.
- To specify a set of databases or tables you can use the Python regular expression syntax. The regular expression pattern must be enclosed in slashes (`/RE_pattern/`). If you use a regular expression, the name must be enclosed in double quotes (`"`). This example `-d "demo/.*/"` specifies all databases in the Greenplum Database installation that begin with `demo`.

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

JDBC Drivers Added to Connectivity Tools

These JDBC 4 drivers have been added to the Greenplum Database Connectivity Tools

- `postgresql-9.4-1201.jdbc4.jar`.
- `postgresql-9.4-1201.jdbc41.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-9.4-1201.jdbc4.jar`.
- For Java 1.7 or 1.8, use the JDBC41 driver `postgresql-9.4-1201.jdbc41.jar`.

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

Greenplum Database PL/Perl Extension Change

The PL/Perl extension has been updated to support future enhancements. The PL/Perl Extension Package version has been increased from `pv1.2` to `pv1.3`.

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

New Parameters

These server configuration parameters have been introduced in Greenplum Database 4.3.6.1. For information about Greenplum Database server configuration parameters, see the *Greenplum Database Reference Guide*.

`filerep_mirrorvalidation_during_resync`

The default setting `false` improves Greenplum Database performance during incremental resynchronization of segment mirrors. Setting the value to `true` enables checking for the existence of files

for all relations on the segment mirror during incremental resynchronization. Checking for files degrades performance of the incremental resynchronization process but provides a minimal check of database objects.

Value Range	Default	Set Classifications
Boolean	false	master session reload

gp_autostats_mode_in_functions

Specifies the mode for triggering automatic statistics collection with `ANALYZE` for statements in procedural language functions. The `none` option disables statistics collection. The `on_no_stats` option triggers statistics collection for `CREATE TABLE AS SELECT`, `INSERT`, or `COPY` operations that are executed in functions on any table that has no existing statistics.

The `on_change` option triggers statistics collection only when the number of rows affected exceeds the threshold defined by `gp_autostats_on_change_threshold`. Operations in functions that can trigger automatic statistics collection with `on_change` are:

```
CREATE TABLE AS SELECT
```

```
UPDATE
```

```
DELETE
```

```
INSERT
```

```
COPY
```

Value Range	Default	Set Classifications
none	none	master
on_change		session
on_no_stats		reload

Deprecated Features

- The Greenplum Database interconnect types TCP and UDP are deprecated. In the next major release, only the UDPIFC interconnect type will be supported by Greenplum Database.

For information about Greenplum Database interconnect types, see the Greenplum Database server configuration parameter `gp_interconnect_type` in the *Greenplum Database Reference Guide*.

- The `ALTER TABLE` command no longer supports the `EXCHANGE DEFAULT PARTITION` clause to exchange the default partition of a partitioned table with an existing Greenplum Database table. If the clause is specified to exchange a default partition, an error is returned.

For information about `ALTER TABLE` command, see the *Greenplum Database Reference Guide*.

Please send any questions or comments about the deprecated items to Pivotal Support at <https://support.pivotal.io>.

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 4.3.6.1 runs on the following platforms:

- 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 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
4.3.6.1	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

Greenplum Database	Data Domain Boost	DDOS
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 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 to Pivotal Support at <https://support.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 4.3.6.x

The table below lists issues that are now resolved in Greenplum Database 4.3.6.x.

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

Table 2: Resolved Issues in 4.3.6.x

Issue Number	Category	Resolved In	Description
25926	DDL and Utility Statements	4.3.6.1	<p>When a user dropped a Greenplum Database table, the table was dropped only on the Greenplum Database master, not on the Greenplum Database segments, in this situation. The table is qualified by a schema name that is the same as the Greenplum Database user, the variable <code>\$user</code> listed in the Greenplum Database <code>search_path</code> to include the user ID in the search path, and the <code>DROP TABLE</code> command issued by the user does not specify the table schema.</p> <p>For example, the user is <code>gpuser1</code>, the table is <code>gpuser1.table1</code>, and the <code>search_path</code> contains "<code>\$user</code>". The table <code>table1</code> is dropped only on the Greenplum Database master when this command is run by <code>gpuser1</code>.</p> <pre>DROP TABLE table1 ;</pre> <p>This issue has been resolved.</p> <p>This issue occurs with Greenplum Database 4.3.6.0. If you installed Greenplum Database 4.3.6.0 and have encountered this issue, please contact Pivotal Support at https://support.pivotal.io for instructions on how to check for and remove the orphan tables on the Greenplum Database segments.</p>
25898	Storage: Transaction Management	4.3.6.1	<p>Greenplum Database performance was degraded due to some Greenplum Database catalog queries that were hanging because Greenplum Database signal handlers did not properly stop the processes associated with the queries.</p> <p>The ability of Greenplum Database signal handlers to stop processes has been enhanced.</p>
25682	Management Scripts: gpssh	4.3.6.1	<p>In some cases, the Greenplum Database utility <code>gpssh</code> failed to connect to some hosts specified either with the <code>-h</code> option or in the file specified by <code>-f</code> option. The utility successfully connected to other specified hosts.</p> <p>This issue has been resolved.</p>
25885	DDL and Utility Statements	4.3.6.1	<p>After exchanging the default partition of a partitioned table with a table that contains data that violates the constraints defined on the default partition, some queries against the partitioned table returned incorrect results.</p> <p>This issue has been resolved. For the <code>ALTER TABLE</code> command, the <code>EXCHANGE DEFAULT PARTITION</code> clause is no longer supported. See <i>Deprecated Features</i>.</p>

Issue Number	Category	Resolved In	Description
25865	Security	4.3.6.1	<p>Greenplum Database software has been updated to resolve these security issues:</p> <ul style="list-style-type: none"> • <i>CVE-2015-0243</i>. Memory errors in functions in the <code>pgcrypto</code> extension. • <i>CVE-2015-0242</i>. Buffer overrun in replacement <code>printf</code> family of functions. • <i>CVE-2015-0241</i>. Buffer overruns in "to_char" functions. • <i>CVE-2013-1900</i>. Random numbers generated by <code>contrib/pgcrypto</code> functions may be easy for another database user to guess.
25845	Backup and Restore	4.3.6.1	<p>When restoring Greenplum Database database objects with the Greenplum Database utility <code>gpdbrestore</code>, tables for a specific schema could not be specified.</p> <p>This issue has been resolved. See <i>gpdbrestore Utility Changes</i>.</p>
25839	Storage: Vacuum	4.3.6.1	<p>For append-optimized tables that contain indexes, the <code>VACUUM</code> command scanned the table index to reclaim storage space from index data even if a vacuum operation was not performed on the append-optimized table. This caused a performance issue.</p> <p>This issue has been resolved. The <code>VACUUM</code> command performs a vacuum operation on indexes of append-optimized tables only when performing a vacuum operation on the append-optimized table.</p>
25830	Query Execution	4.3.6.1	<p>When Pivotal Query Optimizer is enabled, some queries that contain the <code>LEAST</code> or <code>GREATEST</code> function and also contain a subquery that is defined inside the <code>WITH</code> clause might have fallen back to the Greenplum Database legacy optimizer and produced incorrect results.</p> <p>This has been resolved. The query is executed by the Pivotal Query Optimizer and returns the correct results.</p>
25806	Query Optimizer	4.3.6.1	<p>For some queries, the Pivotal Query Optimizer was not able to use multi-column indexes because of incorrect metadata.</p> <p>This issue has been resolved.</p>
25794 25684	Catalog and Metadata	4.3.6.1	<p>Greenplum Database requires a larger amount of memory during a Bitmap Index Scan depending on the number of rescans of the indexed relations. This memory requirement might cause some queries to run out of memory.</p> <p>Memory management involving rescans of the indexed relations has been improved.</p>

Issue Number	Category	Resolved In	Description
25787	Catalog and Metadata	4.3.6.1	If columns were dropped from a table and then <code>ANALYZE</code> was run on the table, the <code>gp_toolkit</code> view <code>gp_stats_missing</code> reported that the table did not have statistics. This issue has been resolved.
25781	Backup and Restore	4.3.6.1	The Greenplum Database <code>gpdbrestore</code> utility failed when the <code>-T</code> or <code>--table-file</code> option was specified to restore database data if the data that was being restored contained a function definition that contains lines that begin with a <code>SET</code> command (without preceding spaces). This issue has been resolved.
25744	Backup and Restore	4.3.6.1	When the Greenplum Database <code>gpdbrestore</code> utility was used to restore specific database tables from a backup on Data Domain server, other objects such as sequences were also incorrectly restored. This issue has been resolved. For information about the utility, see <i>Greenplum Database Utility Guide</i> .
25735	Upgrade / Downgrade	4.3.6.1	When upgrading from Greenplum Database 4.2.x, databases that contained 4.2.x append-only tables were not assigned the correct owner when they were converted to 4.3.x append optimized tables. This issue has been resolved when upgrading from Greenplum Database 4.2.x to Greenplum Database 4.3.6.x. When upgrading from Greenplum Database 4.3.x for example, 4.3.5.3, to Greenplum Database 4.3.6.x, the 4.3.6.x installation contains an SQL script, <code>fix_visimap_owner.sql</code> , that fixes this ownership issue. For information about using the script see <i>Upgrading from 4.3.x to 4.3.6.x</i> .
25724	Query Planner	4.3.6.1	For some queries that contained subqueries, the Greenplum Database planner generated an incorrect plan. When executing the plan, an error occurred when information from the outer query was not passed to the inner subquery. The error caused a Greenplum Database PANIC.
25710	DDL and Utility Statements	4.3.6.1	When default table storage options were specified with the Greenplum Database server configuration parameter <code>gp_default_storage_options</code> , the storage options were applied to Greenplum Database external tables.
25681	DDL and Utility Statements	4.3.6.1	When default table storage options were specified with the Greenplum Database server configuration parameter <code>gp_default_storage_options</code> , the storage options were applied to system tables.

Issue Number	Category	Resolved In	Description
25669	Backup and Restore	4.3.6.1	After an append-optimized table was restored with the Greenplum Database <code>gpdbrestore</code> utility, the table description was not restored.
25650	Storage: Segment Mirroring	4.3.6.1	Performance of the Greenplum Database <code>gprecoverseg</code> utility for incremental and full recovery of a Greenplum Database segment has been improved. For incremental resynchronization of segment mirrors the Greenplum Database server configuration parameter <code>filerep_mirrorvalidation_during_resync</code> controls the of checking for the existence of files during incremental recovery. See <i>filerep_mirrorvalidation_during_resync</i> .
25647	Storage: Access Methods	4.3.6.1	The processing of bit commits and abort transactions has been improved by eliminating look ups and contention on a global data structure during the operations.
25590	Interconnect	4.3.6.1	When no sockets were available on the Greenplum Database master, the message displayed was incorrect.
25589	Interconnect	4.3.6.1	When a Greenplum Database master instance failed over to the standby master, some processes that were controlled by the failed master were not shut down properly.
25588	Management Scripts: expansion	4.3.6.1	The Greenplum Database <code>gpexpand</code> utility performance has been enhanced.
25584	Query Execution	4.3.6.1	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.
25569	Management Scripts: gptransfer	4.3.6.1	When the Greenplum Database <code>gptransfer</code> utility <code>--schema-only</code> option was specified to create only database schemas in the target database, only one schema was created in the database. If the source database contained multiple schemas only one schema was created in the target database.
25564	Backup and Restore	4.3.6.1	When the Greenplum Database <code>gpdbrestore</code> utility options <code>--redirect</code> and <code>--truncate</code> were specified to restore database tables from a back up to a different database and then truncate the data from restored tables, the utility truncated the tables that were in the original database that was backed up.

Issue Number	Category	Resolved In	Description
25551	Loaders: Copy/ ExternalTabs	4.3.6.1	In some cases, the Greenplum Database <code>COPY</code> command returned this error when copying data from a file that contained lines that used a mixture of newline types. ERROR: COPY metadata not found. This issue has been resolved.
25547	OS Abstraction	4.3.6.1	In some cases, Greenplum Database hung because the Greenplum Database logger process could not communicate with an SMTP server. The SMTP communication software has been improved.
25516	DDL and Utility Statements	4.3.6.1	When shutting down a Greenplum Database, the Greenplum Database utility <code>gpstop</code> sometimes incorrectly returned a failure error code when it performed a forceful termination of Greenplum Database processes. This caused other Greenplum Database utilities that use <code>gpstop</code> , such as <code>gpexpand</code> , to fail because of the incorrect failure error code from <code>gpstop</code> . The <code>gpstop</code> utility now returns the correct code.
25474	GPHDFS	4.3.6.1	When the Hadoop file system replication factor is 1, a Hadoop file created for an Greenplum Database external writable table with the <code>gphdfs</code> protocol was created with replication factor of 3.
25385	Client Access Methods and Tools	4.3.6.1	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.
25113	Storage: Access Methods	4.3.6.1	Greenplum Database did not manage the partition of partitioned tables efficiently. On a partitioned table with a large number of partitions, the frequent addition and removal of partitions caused the partitions ID numbers to incorrectly increase to more than the maximum value. This prohibited data from being added to the partition. This issue has been resolved. Greenplum Database manages the partitions of partitioned tables more efficiently.
25081	Interconnect	4.3.6.1	In some cases, when a <code>COPY</code> command that contains a sub-select returns an error, Greenplum Database generated a segmentation fault.
24020	Backup and Restore	4.3.6.1	For the Greenplum Database <code>pgcrondump</code> utility, not all error messages were propagated from backup processes to utility during the backup process. This caused some failed backups to be reported as successful.

Issue Number	Category	Resolved In	Description
23963	Management Scripts: gptransfer	4.3.6.1	The Greenplum Database <code>gptransfer</code> utility would fail if it was run a second time after the <code>gptransfer</code> utility previously failed. The second failure occurred when the utility detected temporary database objects that were created in the target database when it previously failed. This issue has been resolved. The <code>gptransfer</code> utility detects previously created temporary objects and deletes them before starting a transfer process.
23953 25662 25148	Management Scripts: gptransfer	4.3.6.1	The Greenplum Database <code>gptransfer</code> utility failed if the column data contained newline characters and in some cases when handling Null data and no data (""). These issues have been resolved. The default <code>gptransfer</code> utility has been enhanced to use the CSV format to transfer data. The CSV format can handle data that contains newline characters and Null data. For information about changes to <code>gptransfer</code> , see gptransfer Utility Changes .

Known Issues in Greenplum Database 4.3.6.x

This section lists the known issues in Greenplum Database 4.3.6.x. 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 4.3.6.x

Issue	Category	Description
25924	Query Execution	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. Note: For queries executed by the legacy query optimizer, generating plans with bitmap index/scans is enabled.
25829	Dispatch	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.
25737	Catalog and Metadata	Greenplum Database does not support the <code>FILTER</code> clause within aggregate expressions.

Issue	Category	Description
25754	Management Scripts: expansion	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. Workaround: Create the input file manually.
25833	Management Scripts: gpexpand	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.
15835	DDL and Utility Statements	For multi-level partitioned tables that have these characteristics: <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. Splitting a subpartition with the <code>ALTER TABLE SPLIT PARTITION</code> command returns an error and rolls back the transaction.
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	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. Workaround: Manually stop the processes that were started by <code>gpcheckperf</code> .
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 the 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.

Issue	Category	Description
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.
25326	Interconnect	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. The parameter should be set to <code>on</code> only on the Greenplum Database master.
25280	Management Scripts: gpstart/gpstop	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> . Workaround: Unset the environment variable <code>LANG</code> before running the <code>gpstop</code> utility. For example: <pre>\$ unset LANG</pre>
25246	Management Scripts: gpconfig	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. <pre>\$ gpconfig -c gp_email_to -v 'test@example.com'</pre> The improperly set parameter causes Greenplum Database to fail when it is restarted. Workaround: Enclose the value for <code>gp_email_to</code> or <code>gp_email_from</code> with double quotes. <pre>\$ gpconfig -c gp_email_to -v "'test@example.com'"</pre>
25168	Locking, Signals, Processes	When the server configuration parameter <code>client_min_messages</code> is set to either set to <code>PANIC</code> or <code>FATAL</code> and a <code>PANIC</code> or <code>FATAL</code> level message is encountered, Greenplum Database hangs. The <code>client_min_messages</code> parameter should not be set a value higher than <code>ERROR</code> .
24588	Management Scripts: gpconfig	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>). Workaround: The SQL command <code>SHOW</code> displays the correct <code>gp_enable_gpperfmon</code> value.

Issue	Category	Description
24031	gphdfs	<p>If a readable external table is created with <code>FORMAT 'CSV'</code> and uses the gphdfs protocol, reading a record fails if the record spans multiple lines and the record is stored in multiple HDFS blocks.</p> <p>Workaround: Remove line separators from within the record so that the record does not span multiple lines.</p>
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>

Issue	Category	Description
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"> • <i>table.column</i> is NULL. • <i>subquery</i> returns the empty result.
21838	Backup and Restore	<p>When restoring sets of tables with the Greenplum Database utility <code>gpdrestore</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	<p>SSL is only supported on the master host. It is not supported on segment hosts.</p>
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>gpccrondump</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>

Issue	Category	Description
19493 19464 19426	Backup and Restore	<p>The <code>gpcrondump</code> and <code>gpdbrestore</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>gpdbrestore</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>
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>

Issue	Category	Description
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>
18414	Loaders	<p>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).</p>
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>

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

Issue	Category	Description
16064	Backup and Restore	Restoring a compressed dump with the <code>--ddboost</code> option displays incorrect dump parameter information. When using <code>gpdbrestore --ddboost</code> to restore a compressed dump, the restore parameters incorrectly show “Restore compressed dump = Off”. This error occurs even if <code>gpdbrestore</code> passes the <code>--gp-c</code> option to use <code>gunzip</code> for in-line de-compression.
15899	Backup and Restore	When running <code>gpdbrestore</code> with the list (<code>-L</code>) option, external tables do not appear; this has no functional impact on the restore job.

Upgrading to Greenplum Database 4.3.6.x

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.6.x. 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 4.3.6.x, you must install and use a Greenplum Database extension packages (`gpkg` 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 4.3.6.x](#)
- [Upgrading from 4.3.x to 4.3.6.x on Pivotal DCA Systems](#)
- [Upgrading from 4.2.x.x to 4.3.6.x](#)
- [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:

```
gpccrondump --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 4.3.6.x

An upgrade from 4.3.x to 4.3.6.x 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 4.3.6.x on Pivotal DCA Systems](#). This section is for upgrading to Greenplum Database 4.3.6.x 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 4.3.6.x 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-4.3.6.1
```

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-4.3.6.1/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-4.3.6.1 /usr/local/greenplum-db
```

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

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

7. Run the `gpsegininstall` utility to install the 4.3.6.x 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 4.3.6.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.

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 4.3.6.x 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.

Upgrading from 4.3.x to 4.3.6.x on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.6.x 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 4.3.6.x on DCA systems. This section is only for installing Greenplum Database 4.3.6.x 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 4.3.6.x 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 4.3.6.1.

```
# ./greenplum-db-appliance-4.3.6.1-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 4.3.6.x

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.6.x. 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 an 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-4.3.6.1-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.6.1-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-4.3.6.1`), 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 4.3.6.1 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.6.1/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.6.1 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 4.3.6.1 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 4.3.6.1.

```
# ./greenplum-db-appliance-4.3.6.1-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-4.2.6.3/greenplum_path.sh
```

On a DCA system, the path to the file 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 4.3.6.1 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.6.1/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.6.1/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.6.1 installation.
 - a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.6.1 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.6.1 /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-4.3.6.1 /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-4.3.6.1 /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-4.3.6.1 /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-4.3.6.1
```

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-4.3.6.1`.

- As `gpadmin`, run the 4.3.6.x 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-4.2.6.3
  /usr/local/greenplum-db-4.3.6.1
```

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-4.3.6.1`.

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

- The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.3.6.1 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 4.3.6.x system)

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

```
$ gpinitstandby -s standby_hostname
```

- 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*.
- 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.
- 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.
- 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 4.3.6.x 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 4.3.6.x. Detailed upgrade instructions are provided in [Upgrading from 4.2.x.x to 4.3.6.x](#).

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 4.3.6.x.

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 4.3.6.x](#).

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.6.x.

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 4.3.6.x](#).

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.6.x.

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 4.3.6.x](#).

Troubleshooting a Failed Upgrade

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

Be prepared to provide the following information:

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

Greenplum Database Tools Compatibility

Client Tools

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

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

Table 4: Greenplum Database Tools Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql)	4.3	4.3
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC 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.

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 1.2 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 2.1 for Greenplum Database 4.3.x.x	PL/R	8.3.0.15
	R	3.1.0

Greenplum Database Extension	Extension Components	
	Name	Version
PL/R 1.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.2 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 6.x 5.5.8 on RHEL 5.x, SUSE 10
PL/Perl 1.1 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
PL/Perl 1.0 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
Pgcrypto 1.2 for Greenplum Database 4.3.x.x	Pgcrypto	Based on PostgreSQL 8.3
MADlib 1.5 for Greenplum Database 4.3.x.x	MADlib	Based on MADlib version 1.8

Note: Greenplum Database 4.3.6.x does not support the PostGIS 1.0 extension package.

Greenplum Database 4.3.6.x supports these minimum Greenplum Database extensions package versions.

Table 6: Greenplum Database 4.3.6.x Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1 and release <code>gpdb4.3orca</code>
PL/Java	1.2 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 4.3.6.x, 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 4.3.6.x.

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 Pivotal HD 1.0 ¹	gphd-2.0
Greenplum HD	Greenplum HD 1.2	gphd-1.2
	Greenplum HD 1.1	gphd-1.1 (default)
Cloudera	CDH 5.2, 5.3	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	hdp2
MapR ³	MapR 4.x	gpmr-1.2
	MapR 1.x, 2.x, 3.x	gpmr-1.0

Hadoop Distribution	Version	gp_hadoop_target_version
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 4.3.6.1 Documentation

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

Table 8: Greenplum Database Documentation

Title	Revision
<i>Greenplum Database 4.3.6.1 Release Notes</i>	A03
<i>Greenplum Database 4.3 Installation Guide</i>	A09
<i>Greenplum Database 4.3 Administrator Guide</i>	A11
<i>Greenplum Database 4.3 Reference Guide</i>	A11
<i>Greenplum Database 4.3 Utility Guide</i>	A12
<i>Greenplum Database 4.3 Client Tools for UNIX</i>	A05
<i>Greenplum Database 4.3 Client Tools for Windows</i>	A04
<i>Greenplum Database 4.3 Connectivity Tools for UNIX</i>	A04
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A03
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A08
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A07
<i>Greenplum Command Center 1.3 Administrator Guide</i>	A04

Copyright

[Privacy Policy](#) | [Terms of Use](#)

Copyright © 2017 Pivotal Software, Inc. All rights reserved.

Pivotal Software, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." PIVOTAL SOFTWARE, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.