

# Greenplum Database 4.3.5.0 Release Notes

Rev: A03

Updated: July, 2015

## Welcome to Pivotal Greenplum Database 4.3.5.0

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

**Note:** This document contains pertinent release information about Greenplum Database 4.3.5.0. For previous versions of the release notes for Greenplum Database, go to [Pivotal Documentation](#) or EMC [Support Zone](#). For information about Greenplum Database end of life, see [Greenplum Database end of life policy](#).

## About Greenplum Database 4.3.5.0

Greenplum Database 4.3.5.0 is a minor release that introduces performance and stability enhancements. Please refer to the following sections for more information about this release.

- [Product Enhancements](#)
- [Changed Features](#)
- [Supported Platforms](#)
- [Resolved Issues in Greenplum Database 4.3.5.x](#)
- [Known Issues in Greenplum Database 4.3.5.x](#)
- [Upgrading to Greenplum Database 4.3.5.0](#)
- [Greenplum Database Tools Compatibility](#)
- [Greenplum Database Extensions Compatibility](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.3.5.0 Documentation](#)

## Product Enhancements

Greenplum Database 4.3.5.0 includes enhancements in these areas:

- [Pivotal Query Optimizer](#)
- [Utility to ANALYZE Incrementally and Concurrently](#)

- *Append-optimized Compaction Information*

## Pivotal Query Optimizer

In Greenplum Database 4.3.5.0, the Pivotal Query Optimizer co-exists with the legacy query optimizer. Both query optimizers, generate query plans that are executed by Greenplum Database. The Pivotal Query Optimizer extends the planning and optimization capabilities of Greenplum Database. The Pivotal Query Optimizer is extensible, verifiable, and achieves better optimization using multi-core architectures.

Pivotal Query Optimizer includes enhancements for specific types of queries and operations:

- Queries against partitioned tables
- Queries that contain sub-queries
- Queries that contain a common table expression (CTE)
- DML operations

Pivotal Query Optimizer includes contains these optimization enhancements:

- Improved join ordering
- Join-Aggregate reordering
- Sort order optimization
- Data skew estimates included in query optimization

By default, Greenplum Database uses the legacy query optimizer. When the Pivotal Query Optimizer is enabled, Greenplum Database uses the Pivotal Query Optimizer to generate an execution plan for a query when possible. If the Pivotal Query Optimizer cannot be used, the legacy query optimizer is used.

**Notes:** These notes apply when the Pivotal Query Optimizer is enabled:

- If you intend to execute queries on partitioned tables with the Pivotal Query Optimizer enabled, the server configuration parameter `optimizer_analyze_root_partition` must be set to `on` and you must collect statistics on the partitioned table root partition.
- For Greenplum Command Center monitoring performance, Pivotal recommends the default setting for Pivotal Query Optimizer (`off`) for the `gpperfmon` database that is used by Greenplum Command Center. Enabling Pivotal Query Optimizer for the `gpperfmon` database is not supported. To ensure that the Pivotal Query Optimizer is disabled for the `gpperfmon` database, run this command on the system where the database is installed:

```
ALTER DATABASE gpperfmon SET OPTIMIZER = OFF
```

For information about Pivotal Query Optimizer usage, features, and limitations, see "Querying Data" in the *Greenplum Database Administrator Guide*.

## Statistics Collection on the Root Partition of a Partitioned Table

The `ANALYZE` command collects statistics about the contents of tables in the database. For Greenplum Database 4.3.5.0, the new `ROOTPARTITION` clause lets you analyze only the root partition of a partitioned table without analyzing all the individual partitions of a partitioned table. This clause is useful when the individual partitions have been analyzed and do not need to be analyzed again. This is the syntax of the `ANALYZE` command.

```
ANALYZE [VERBOSE] [ROOTPARTITION [ALL] ]
[table [ (column [, ...] ) ]]
```

### ROOTPARTITION

Collect statistics only on the root table of partitioned tables. When you specify `ROOTPARTITION`, you must specify either `ALL` or the name of a partitioned table.

If you specify `ALL` with `ROOTPARTITION`, Greenplum Database collects statistics for the root table of all partition tables in the database. If there are no partitioned tables in the database, a message is returned. For tables that are not partitioned tables, statistics are not collected.

If you specify a table name with `ROOTPARTITION`, and the table is not partitioned table, a warning is returned. No statistics are collected for the table.

For tables that are not partitioned tables, statistics are not collected. For information about the other clauses for the `ANALYZE` command, see the *Greenplum Database Reference Guide*.

For a partitioned table, specifying which portion of the table to analyze (the root table or child leaf tables) can be useful if the partitioned table has large number of partitions that have been analyzed and only a few leaf child tables have changed. Partitioned tables, child tables and their inheritance level relationships are tracked in the system view `pg_partitions`.

The `ROOTPARTITION` clause is not valid with `VACUUM ANALYZE`. The command `VACUUM ANALYZE ROOTPARTITION` returns an error.

For the partitioned table `sales_curr_yr`, this example command collects statistics only on the root table of the partitioned table.

```
ANALYZE ROOTPARTITION sales_curr_yr;
```

This example `ANALYZE` command collects statistics on all root tables of the partitioned tables in the database.

```
ANALYZE ROOTPARTITION ALL;
```

**Note:** You can also use the Greenplum Database utility `analyzedb` to update table statistics. For information about the `analyzedb` utility, see the next section.

## Utility to ANALYZE Incrementally and Concurrently

In Greenplum Database 4.3.5.0, the new utility `analyzedb` can perform `ANALYZE` operations on tables incrementally and in parallel. For append optimized tables, the utility can limit updating statistics on the table only if the table was changed after statistics were previously collected for the table.

While performing the `ANALYZE` operations, `analyzedb` creates a snapshot of table metadata snapshot and stores it on disk on the master host. If you run `analyzedb` to `ANALYZE` an append-optimized table, the utility checks the metadata taken during a previous `analyzedb` operation to determine if the table has been modified. An `ANALYZE` operation is performed only if the table has been modified. If a table or partition has not been modified since the last time it was analyzed, `analyzedb` automatically skips the table or partition because it already contains up-to-date statistics.

For a partitioned, append-optimized table, `analyzedb` checks the partitioned table root partition and leaf partitions. If needed, the utility updates statistics for partitions that have changed and, if necessary, the root partition.

To perform the `ANALYZE` operations, `analyzedb` creates concurrent sessions to analyze tables in parallel. For each session, `analyzedb` issues an `ANALYZE` command to the database and specifies different table names.

This example command specifies a file that contains a list of tables. The command collects statistics on the tables listed in the file `analyze-tables` in the database named `mytest`.

```
$ analyzedb -d mytest -f analyze-tables
```

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

## Append-optimized Compaction Information

In Greenplum Database 4.3.5.0, the new the function `__gp_aovisimap_compaction_info(oid)` displays compaction information for an append-optimized table. The information is for the on-disk data files on Greenplum Database segments that store the table data. You can use the information to determine the data files that will be compacted by a `VACUUM` operation on an append-optimized table.

**Note:** Until a `VACUUM` operation deletes the row from the data file, deleted or updated data rows occupy physical space on disk even though they are hidden to new transactions. The configuration parameter `gp_appendonly_compaction` controls the functionality of the `VACUUM` command.

## Changed Features

- *Changed Features in Greenplum Database 4.3.5.0*
- *New Configuration Parameters*
- *Changed Server Configuration Parameters*

### Changed Features in Greenplum Database 4.3.5.0

These are changes to existing features:

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

To use extension packages with Greenplum Database 4.3.5.0, you must install and use Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0. 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.

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

- For partitioned tables, statistics can be collected on the root partition. The root partition statistics are required when you run a query against a partition with Pivotal Query Optimizer. See *Statistics Collection on the Root Partition of a Partitioned Table*.
- The functionality of server configuration parameter `gp_create_table_random_default_distribution` has been enhanced to support the table creation by the Pivotal Query optimizer. The server configuration parameter can control the default distribution policy.
- When executing an SQL command with the Pivotal Query Optimizer, Greenplum Database issues a warning if the command performance could be improved by collecting statistics on a column or set of columns referenced by the command. The warning is issued on the command line and information is added to the Greenplum Database log file. For information about collecting statistics on table columns, see the `ANALYZE` command in the *Greenplum Database Reference Guide*.
- For Greenplum Command Center monitoring performance, Pivotal recommends the default setting for Pivotal Query Optimizer (`off`) for the `gpperfmon` database that is used by Greenplum Command Center. Enabling Pivotal Query Optimizer for the `gpperfmon` database is not supported. To ensure that the Pivotal Query Optimizer is disabled for the `gpperfmon` database, run this command on the system where the database is installed. See *Pivotal Query Optimizer*.
- Greenplum Database 4.3.5.0 does not support the PostGIS 1.0 extension package.

### New Configuration Parameters

These server configuration parameters have been introduced in Greenplum Database 4.3.5.0.

## optimizer

Enables the Pivotal Query Optimizer when running SQL queries. The default is `off`, Greenplum Database uses only the legacy query optimizer.

The Pivotal Query Optimizer co-exists with the legacy query optimizer. When the Pivotal Query Optimizer is enabled, Greenplum Database uses the Pivotal Query Optimizer to generate an execution plan for a query when possible. If the Pivotal Query Optimizer cannot be used, the legacy query optimizer is used.

Value Range	Default	Set Classifications
Boolean	off	master session reload

## optimizer\_analyze\_root\_partition

For a partitioned table, collects statistics for the root partition when the `ANALYZE` command is run on the table. The Pivotal Query Optimizer uses the root partition statistics. The legacy query optimizer does not use these statistics. If you set the value of the server configuration parameter `optimizer` to `on`, set the value of this parameter to `on` and run the command `ANALYZE` or `ANALYZE ROOTPARTITION` on partitioned tables to ensure the proper statistics have been collected.

Value Range	Default	Set Classifications
Boolean	off	master session reload

## optimizer\_control

Controls whether the server configuration parameter `optimizer` can be changed with `SET`, the `RESET` command, or the Greenplum Database utility `gpconfig`. If the `optimizer_control` parameter value is `on`, users can set the `optimizer` parameter. If the `optimizer_control` parameter value is `off`, the `optimizer` parameter cannot be changed.

Value Range	Default	Set Classifications
Boolean	on	master system restart superuser

## optimizer\_minidump

The Pivotal Query Optimizer generates minidump files to describe the optimization context for a given query. The minidump files are used by Pivotal support to analyze Greenplum Database issues. The minidump file is located under the master data directory and uses the following naming format:

```
Minidump_date_time.mdp
```

Setting this parameter to `ALWAYS` generates a minidump for all queries. Pivotal recommends that you set this parameter to `ONERROR` in production environments to minimize costs.

Value Range	Default	Set Classifications
ONERROR ALWAYS	ONERROR	master session reload

## Changed Server Configuration Parameters

In Greenplum Database 4.3.5.0, Greenplum Database table distribution policy and functionality of the server configuration parameter `gp_create_table_random_default_distribution` has changed.

### `gp_create_table_random_default_distribution`

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

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

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

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

For a `CREATE TABLE AS` command that does not contain a distribution clause:

- If the legacy query optimizer creates the table, and the value of the parameter is `off`, the table distribution policy is determined based on the command.
- If the legacy query optimizer creates the table, and the value of the parameter is `on`, the table distribution policy is random.
- If the Pivotal Query Optimizer creates the table, the table distribution policy is random. The parameter value has no affect.

Value Range	Default	Set Classifications
boolean	off	master system reload

## 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 4.3.5.0 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

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

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
4.3.5.0	3.0.0.3	5.5.0.x 5.4 (all versions) 5.3 (all versions)
4.3.4.1 4.3.4.0	3.0.0.3	5.5.0.x 5.4 (all versions) 5.3 (all versions)
4.3.3.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.2.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.1.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.0.0	2.4.2.2	5.0.1.0, 5.1, and 5.2

**Note:** In addition to the DDOS versions listed in the previous table, Greenplum Database 4.3.4.0 and later supports all minor patch versions (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.

## Resolved Issues in Greenplum Database 4.3.5.x

The table below lists issues that are now resolved in Greenplum Database 4.3.5.x.

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

**Table 2: Resolved Issues in 4.3.5.0**

Issue Number	Category	Description
90561896	Management Scripts: recoverseg	In Greenplum Database 4.3.4.1, the Greenplum Database <code>gprecoverseg</code> utility checked persistent tables by default. In some cases, this check reported false positives for catalog corruptions.  For this release, the persistent table checks have been removed from <code>gprecoverseg</code> . To check for persistent table issues, use the Greenplum Database <code>gpcheckcat</code> utility.  The behavior in Greenplum Database has been reverted to the behavior in 4.3.4.0 and earlier.
89931274	Security	Greenplum Database software has been updated to use OpenSSL 0.9.8ze. For information about major changes in OpenSSL 0.9.8ze, see <a href="http://www.openssl.org/news/openssl-0.9.8-notes.html">http://www.openssl.org/news/openssl-0.9.8-notes.html</a> .
87808098	Loaders	In some cases, the Greenplum Database utility <code>gpfdist</code> issued an error message when a network connection to an ETL (extract, transform, and load) host was disconnected due to an issue with the host or the connection with the host.  The message has been enhanced to identify the cause of the error.
25423	Storage: Access Methods	In some cases, running the Greenplum Database utility <code>gpcrondump</code> caused a PANIC on some Greenplum Database segments.
25422	Management Scripts: General	The Greenplum Database utility <code>gprecoverseg</code> returns an error when the <code>PGPORT</code> environment variable is not set.
25417	Monitoring: gpperfmon server	In some cases when the Greenplum Command Center is installed, the <code>gpsmon</code> process timed out after one hour and returned an error that no request were received after 3600 seconds.
25339	Query Execution	In some cases, an out of memory error occurred during the evaluation of a per-row SQL function that required executing a nested query plan.



Issue Number	Category	Description
25335	Catalog and Metadata, Global Persistent Objects	In some cases, rebuilding a persistent table in a Greenplum database failed if the table was created with a non-default table space.
25311 25350	Query Optimizer	During query optimization, some queries with a large number of conjunctive predicates could consume a large amount of memory.  This issue has been resolved.
25305	Backup and Restore	The Greenplum Database utility <code>gpmfr</code> failed when all the files that were being backed up were less than 1000 bytes.
25297	Query Optimizer	Some queries returned wrong results when an <code>IN</code> clause contained a nested expression.  This issue has been resolved.
25296	Query Optimizer	Some queries that contained aggregate functions were terminated by an error because of incorrect plans within a window function.  This issue has been resolved.
25292 25361	Query Optimizer	Inefficient plans were generated for queries that contained the function <code>unnest (ARRAY[...])</code> . The plan generated by the Pivotal Query Optimizer for this type query has been improved.
25288	Query Execution	A Greenplum Database PANIC occurred when deleting data from a table with the <code>DELETE</code> command if the contains a <code>USING</code> clause.
25279	Management Scripts: gpstart/gpstop	In some cases, the Greenplum Database utility <code>gpstop</code> issued the warning <code>No leftover gpmon process found</code> . These warning messages have be changed to informational messages.
25252	Query Optimizer	Some queries that required partition elimination with a <code>NOT IN</code> predicate caused a crash.  This issue has been resolved.
25175	Query Execution	A Greenplum Database PANIC occurred when using the <code>COPY</code> command to copy data into a table that contains no columns.
25170	Storage: Vacuum	In some cases, running the Greenplum Database utility <code>vacuumdb</code> caused a Greenplum Database PANIC due to issues with a system table that is used to track append optimized file segments.
25160	Query Execution	In some cases, running a query and concurrently performing a DDL operation on the same data returned this error.  <code>ERROR", "XX000", "could not open relation</code>

Issue Number	Category	Description
25124	Dispatch	<p>If a cursor was declared in a transaction, and then a <code>SET</code> command was issued in the same transaction before the cursor was closed, a Greenplum Database crash occurred.</p> <p>In Greenplum Database 4.3.5.0, an error is returned if the <code>SET</code> command is issued while a cursor is declared and not closed in a transaction. In the transaction, the cursor must be closed before the <code>SET</code> command can be issued.</p>
25081	Interconnect	In some cases, when a <code>COPY</code> command that contains a sub-select returns an error, Greenplum Database generated a segmentation fault.
24953	Management Scripts: gptoolkit	<p>For append-optimized tables, display information about the amount of bloat (table disk space is occupied by deleted or obsolete rows) in the on-disk data files that are used by the tables.</p> <p>The Greenplum Database function <code>__gp_aovisimap_compaction_info</code> displays append-optimized table on-disk storage and bloat information. See <a href="#">Append-optimized Compaction Information</a></p>
24944	DDL and Utility Statements	The <code>set_config()</code> function changed the sever configuration parameter only on the Greenplum Database master, not on the Greenplum Database segment instances.
24621	Backup and Restore, Functions and Languages	The Greenplum Database function <code>to_date()</code> did not validate the range of the input date.
24591	Backup and Restore	<p>In some cases, the Greenplum Database utility <code>gpcrondump</code> failed with the error <code>Cannot allocate memory</code>.</p> <p>The memory management of the Greenplum Database utility has been enhanced to minimize occurrence of the error.</p>
24557	Query Optimizer	<p>Some queries with aggregate functions that contained outer references returned the error message: <code>aggref found in non-Agg plan node</code>.</p> <p>This issue has been resolved.</p>
24263	Query Optimizer	Some queries with predicates on the join key of a left outer join did not push down a predicate. The plan generated by the Pivotal Query Optimizer for this type query has been improved.
23801	DDL and Utility Statements	<p>For a table with a primary key, the <code>ALTER TABLE</code> command could change the distribution policy (the columns specified with the <code>DISTRIBUTION KEY</code> clause) to a non-primary key. Specifying a <code>DISTRIBUTION KEY</code> to a non-primary key column is not supported.</p> <p>This issue has been resolved.</p>

Issue Number	Category	Description
18673	DDL and Utility Statements	In some cases, SQL commands that were executed concurrently with an <code>ALTER TABLE</code> command that contains a <code>SPLIT PARTITION</code> clause on a partitioned table returned this error:  <pre>ERROR: Relation decrement reference count found relation <i>relation-id</i> with bad count</pre>

## Known Issues in Greenplum Database 4.3.5.x

This section lists the known issues in Greenplum Database 4.3.4.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 4.3.5.x**

Issue	Category	Description
90799642	Query Optimizer	For queries that include <code>DISTINCT</code> aggregates expressed as window functions, the query might return wrong results because the <code>DISTINCT</code> qualifier is incorrectly dropped in the window operator.
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).
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>&lt;</code> and <code>,</code> <code>&gt;</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> .  <b>Workaround:</b> 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@my-email.com'</pre> The improperly set parameter causes Greenplum Database to fail when it is restarted.  <b>Workaround:</b> 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@my-email.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> ).  <b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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"> <li>• <i>table.column</i> is NULL.</li> <li>• <i>subquery</i> returns the empty result.</li> </ul>
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><b>Workaround:</b> 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>gpcrondump</code> options <code>--ddboost-host</code> and <code>--ddboost-user</code>.</p>
18247	DDL and Utility Statements	<p><code>TRUNCATE</code> command does not remove rows from a sub-table of a partitioned table. If you specify a sub-table of a partitioned table with the <code>TRUNCATE</code> command, the command does not remove rows from the sub-table and its child tables.</p> <p><b>Workaround:</b> 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><b>Workaround:</b> 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"> <li>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.</li> <li>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.</li> </ul> <p><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> Reinstall <code>gp_toolkit</code>.</p>
18710	Management Scripts Suite	<p>Greenplum Management utilities cannot parse IPv6 IP addresses.</p> <p><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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><b>Workaround:</b> 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 gunzip 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 4.3.5.0

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.5.0. The minimum recommended upgrade path for this release is from Greenplum Database version 4.2.x.x. If you have an earlier major version of the database, you must first upgrade to version 4.2.x.x.

### Prerequisites

Before starting the upgrade process, Pivotal recommends performing the following checks. Pivotal recommends running the `gpcheckcat` utility a few weeks before the upgrade during a maintenance period. If necessary, you can resolve any issues found by the utility before the scheduled upgrade.

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

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

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

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

**Important:** If you intend to use an extension package with Greenplum Database 4.3.5.0, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0. 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.

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

- [Upgrading from 4.3.x to 4.3.5.0](#)
- [Upgrading from 4.3.x to 4.3.5.0 on Pivotal DCA Systems](#)
- [Upgrading from 4.2.x.x to 4.3.5.0](#)
- [For Users Running Greenplum Database 4.1.x.x](#)

- *For Users Running Greenplum Database 4.0.x.x*
- *For Users Running Greenplum Database 3.3.x.x*
- *Migrating a Greenplum Database That Contains Append-Only Tables*

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

```
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 4.3.5.0

An upgrade from 4.3.x to 4.3.5.0 involves stopping Greenplum Database, updating the Greenplum Database software binaries, upgrading and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 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.

**Important:** If you are upgrading from Greenplum Database 4.3.x on a Pivotal DCA system, see *Upgrading from 4.3.x to 4.3.5.0 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.5.0 on non-DCA systems.

**Note:** If you are upgrading from Greenplum Database between 4.3.0 and 4.3.2, run the `fix_ao_upgrade.py` utility to check Greenplum Database for the upgrade issue and fix the upgrade issue (See step 11). The utility is in this Greenplum Database 4.3.5.x directory: `$GPHOME/share/postgresql/upgrade`

For information about the utility, see *fix\_ao\_upgrade.py Utility*.

**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 4.3.5.0 on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example:

```
/usr/local/greenplum-db-4.3.5.0
```

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

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

to:

```
source /usr/local/greenplum-db-4.3.5.0/greenplum_path.sh
```

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

```
$ rm /usr/local/greenplum-db
$ ln -s /usr/local/greenplum-db-4.3.5.0 /usr/local/greenplum-db
```

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

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

7. Run the `gpsegininstall` utility to install the 4.3.5.0 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

8. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 (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.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.

## 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 `gadmin`.

#### **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 4.3.5.0 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.5.0 on a Pivotal DCA system involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 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.

**Important:** Skip this section if you are *not* installing Greenplum Database 4.3.5.0 on DCA systems. This section is only for installing Greenplum Database 4.3.5.0 on DCA systems.

**Note:** If you are upgrading from Greenplum Database between 4.3.0 and 4.3.2, run the `fix_ao_upgrade.py` utility to check Greenplum Database for the upgrade issue and fix the upgrade issue (See step 8). The utility is in this Greenplum Database 4.3.5.x 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 4.3.5.0 on the Greenplum Database master host and specify the file `hostfile` that lists all hosts in the cluster. If necessary, copy `hostfile` to the directory containing the installer before running the installer.

This example command runs the installer for Greenplum Database 4.3.5.0.

```
# ./greenplum-db-appliance-4.3.5.0-build-1-RHEL5-x86_64.bin hostfile
```

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

6. Install Greenplum Database extension packages.

**Important:** Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 (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 4.3.5.0

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.5.0. For users running versions prior to 4.2.x.x of Greenplum Database, see the following:

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

## Planning Your Upgrade

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

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

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

For example:

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

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

**Important:** If you intend to use an extension packages with Greenplum Database 4.3.5.0, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0. 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.

## 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-4.3.5.0-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.5.0-PLATFORM.bin
```

4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press `ENTER` to accept the default install path (for example: `/usr/local/greenplum-db-4.3.5.0`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer installs the Greenplum Database software and creates a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Source the path file from your new 4.3.5.0 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.5.0/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.5.0 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

### ***Install the Greenplum Database 4.3 Software Binaries on DCA Systems***

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

1. Download or copy the installer file to the Greenplum Database master host.
2. As root, run the Pivotal DCA installer for 4.3.5.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 4.3.5.0.

```
# ./greenplum-db-appliance-4.3.5.0-build-1-RHEL5-x86_64.bin hostfile
```

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

### ***Upgrade Execution***

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

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

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

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

2. (*optional but strongly recommended*) Back up all databases in your Greenplum Database system using `gpcrondump`. See the *Greenplum Database Administrator Guide* for more information on how to do backups using `gpcrondump`. Make sure to secure your backup files in a location outside of your Greenplum data directories.

3. If your system has a standby master host configured, remove the standby master from your system configuration. For example:

```
$ gpinitstandby -r
```

4. Perform a clean shutdown of your current Greenplum Database 4.2.x.x system. This example uses the `-a` option to disable confirmation prompts:

```
$ gpstop -a
```

5. Source the path file from your new 4.3.5.0 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.5.0/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.5.0/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.5.0 installation.

- a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.5.0 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.5.0 /usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

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

```
# ln -s /usr/local/GP-4.3.5.0 /usr/local/greenplum-db
```

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

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.3.4.1 /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-4.3.5.0 /usr/local/greenplum-db
```

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

```
$ gpmigrator_mirror --check-only
  /usr/local/greenplum-db-4.2.6.3
  /usr/local/greenplum-db-4.3.5.0
```

On a DCA system, the old `GPHOME` location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new `GPHOME` location would be similar to `/usr/local/GP-4.3.5.0`.

- As `gpadmin`, run the 4.3.5.0 version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ gpmigrator_mirror /usr/local/greenplum-db-4.2.6.3
  /usr/local/greenplum-db-4.3.5.0
```

On a DCA system, the old `GPHOME` location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new `GPHOME` location would be similar to `/usr/local/GP-4.3.5.0`.

**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 4.3.5.x 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 4.3.5.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 4.3.5 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. (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 4.3 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](#).

- (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
```

- Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.4.1 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 4.3.5.x. Detailed upgrade instructions are provided in *Upgrading from 4.2.x.x to 4.3.5.0*.

Pre-Upgrade Preparation (on your current system)
* 4.2.x.x system is up and available
<input type="checkbox"/> Log in to your master host as the <code>gpadmin</code> user (your Greenplum superuser).
<input type="checkbox"/> (Optional) Run <code>VACUUM</code> on all databases,
<input type="checkbox"/> (Optional) Remove old server log files from <code>pg_log</code> in your master and segment data directories.
<input type="checkbox"/> Check for and recover any failed segments ( <code>gpstate</code> , <code>gprecoverseg</code> ).
<input type="checkbox"/> Copy or preserve any additional folders or files (such as backup folders).
<input type="checkbox"/> Install the Greenplum Database 4.3 binaries on all Greenplum hosts.
<input type="checkbox"/> Inform all database users of the upgrade and lockout time frame.
Upgrade Execution
* The system will be locked down to all user activity during the upgrade process
<input type="checkbox"/> Backup your current databases.
<input type="checkbox"/> Remove the standby master ( <code>gpinitstandby -r</code> ).
<input type="checkbox"/> Do a clean shutdown of your current system ( <code>gpstop</code> ).
<input type="checkbox"/> Update your environment to source the new Greenplum Database 4.3.5 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 4.3.5 system will be up and running.
Post-Upgrade (on your 4.3.5 system)
* The 4.3.5 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 4.3.5 installation.
<input type="checkbox"/> Download and install any Greenplum Database extensions.
<input type="checkbox"/> (Optional) Install the latest Command Center Console and update your environment to point to the latest Command Center binaries.
<input type="checkbox"/> Inform all database users of the completed upgrade.

### **For Users Running Greenplum Database 4.1.x.x**

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.4.1.

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 4.3.5.0](#).

### **For Users Running Greenplum Database 4.0.x.x**

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.4.1.

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 4.3.5.0](#).

### **For Users Running Greenplum Database 3.3.x.x**

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.4.1.

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](#)).



- Follow the upgrade instructions in these release notes for *Upgrading from 4.2.x.x to 4.3.5.0*.

## Troubleshooting a Failed Upgrade

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

**Be prepared to provide the following information:**

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

## Greenplum Database Tools Compatibility

### Client Tools

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

Tool packages are available from *Pivotal Network*.

**Table 4: Greenplum Database Tools Compatibility**

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

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

- AIX 5.3L (32-bit)
- AIX 5.3L and AIX 6.1 (64-bit)
- Apple OSX on Intel processors (32-bit)
- HP-UX 11i v3 (B.11.31) Intel Itanium (Client and Load Tools only)
- Red Hat Enterprise Linux i386 (RHEL 5)
- Red Hat Enterprise Linux x86\_64 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
PL/R 1.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	2.13.0

Greenplum Database Extension	Extension Components	
	Name	Version
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 4.3.5.0 does not support the PostGIS 1.0 extension package.

Greenplum Database 4.3.5.0 supports these minimum Greenplum Database extensions package versions.

**Table 6: Greenplum Database 4.3.5.0 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 due to the introduction of Pivotal Query Optimizer. Also, extension packages for Greenplum Database 4.3.5.0 are not compatible with Greenplum Database 4.3.4.x and earlier.

To use extension packages with Greenplum Database 4.3.5.0, you must install and use Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0. 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.

## 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 2.0 Pivotal HD 1.0 <sup>1</sup>	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 <sup>2</sup> - CDH 4.7	cdh4.1
Hortonworks Data Platform	HDP 2.1, 2.2	hdp2
MapR <sup>3</sup>	MapR 4.x	gpmr-1.2
	MapR 1.x, 2.x, 3.x	gpmr-1.0

### Notes:

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

## Greenplum Database 4.3.5.0 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 4.3.5.0 Release Notes</i>	A03
<i>Greenplum Database 4.3 Installation Guide</i>	A07
<i>Greenplum Database 4.3 Administrator Guide</i>	A08
<i>Greenplum Database 4.3 Reference Guide</i>	A08
<i>Greenplum Database 4.3 Utility Guide</i>	A08
<i>Greenplum Database 4.3 Client Tools for UNIX</i>	A04
<i>Greenplum Database 4.3 Client Tools for Windows</i>	A04
<i>Greenplum Database 4.3 Connectivity Tools for UNIX</i>	A03
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A03
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A06
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A05
<i>Greenplum Command Center 1.3 Administrator Guide</i>	A01