Pivotal

RELEASE NOTES

Greenplum Database 4.3.5.3 Release Notes

Rev: A02

Updated: August, 2015

Welcome to Pivotal Greenplum Database 4.3.5.3

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 4.3.5.3. 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 4.3.5.3

Greenplum Database 4.3.5.3 is a patch release that resolves known issues and includes product enhancements. Please refer to the following sections for more information about this release.

Product Enhancements

Greenplum Database 4.3.5.3 includes these enhancements:

- Greenplum Database supports `INSERT` operations into the leaf child partition table of a partitioned table.

To insert data into a partitioned table, you can specify a leaf child table of the partitioned table in an `INSERT` command. Leaf child tables are the lowest-level tables in the hierarchy of child tables created by Greenplum Database for use by the partitioned table. An error is returned if the data is not valid for the specified leaf child table. Specifying a child table that is not a leaf child table in the `INSERT` command is not supported.

Execution of other DML commands such as `UPDATE` and `DELETE` on any child table of a partitioned table is not supported. These commands must be executed on the root partitioned table, the table created with the `CREATE TABLE` command.

For information about partitioned tables, see the *Greenplum Database Administrator Guide*. For information about the `INSERT` command, see the *Greenplum Database Reference Guide*.

- The Greenplum Database utility `analyzedb` supports analyzing database catalog tables. For information about `analyzedb`, see the *Greenplum Database Utility Guide*.

Changed and Deprecated Features

For Greenplum Database 4.3.5.3, there are changed and deprecated features:

Changed Feature

Greenplum Database logging has been improved when falling back to the legacy query optimizer from the Pivotal Query Optimizer.

- Greenplum Database no longer adds extraneous log messages to the log file.
- The log file messages contain a `NOTICE` indicating the reason the Pivotal Query Optimizer did not execute the query. This is an example notice that contained in a log message.

```
NOTICE,""Feature not supported by the Pivotal Query Optimizer: Queries on
 master-only tables""
```

In previous releases, the information was in a `ERROR` string.

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

Deprecated Features

- For the `LOG ERRORS [INTO` *`error_table`*`]` clause of the Greenplum Database commands `COPY` and `CREATE EXTERNAL TABLE`, the optional clause `INTO` *`error_table`* that creates error tables is deprecated and will be removed in the next major release. When you specify the `LOG ERRORS` clause, Greenplum Database captures table row formatting error information in the within Greenplum Database, not in a user table.

  For information about `COPY` and `CREATE EXTERNAL TABLE` commands, see the *Greenplum Database Reference Guide*.
- Support for the Greenplum MapReduce product is deprecated. Greenplum MapReduce will not be supported in a future release.

  Note: Please send any questions or comments about the deprecated features to *gpdb@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 4.3.5.3 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:

- 6.*xxx*
- 7.*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
4.3.5.3 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)
4.3.4.1 4.3.4.0	3.0.0.3	5.5.0.*x* 5.4 (all versions) 5.3 (all versions)
4.3.3.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.2.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.1.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.0.0	2.4.2.2	5.0.1.0, 5.1, and 5.2

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

Greenplum Database support on DCA:

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

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

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

Supported Platform Notes

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

- The only file system supported for running Greenplum Database is the XFS file system. All other file systems are explicitly *not* supported by Pivotal.
- Greenplum Database is supported on all 1U and 2U commodity servers with local storage. Special purpose hardware that is not commodity *may* be supported at the full discretion of Pivotal Product Management based on the general similarity of the hardware to commodity servers.
- Greenplum Database is supported on network or shared storage if the shared storage is presented as a block device to the servers running Greenplum Database and the XFS file system is mounted on the block device. Network file systems are *not* supported. When using network or shared storage, Greenplum Database mirroring must be used in the same way as with local storage, and no modifications may be made to the mirroring scheme or the recovery scheme of the segments. Other features of the shared storage such as de-duplication and/or replication are not directly supported by Pivotal, but may be used with support of the storage vendor as long as they do not interfere with the expected operation of Greenplum Database at the discretion of Pivotal.
- Greenplum Database is supported when running on virtualized systems, as long as the storage is presented as block devices and the XFS file system is mounted for the storage of the segment directories.
- A minimum of 10-gigabit network is required for a system configuration to be supported by Pivotal.
- Greenplum Database is supported on Amazon Web Services (AWS) servers using either Amazon instance store (Amazon uses the volume names `ephemeral[0-20]`) or Amazon Elastic Block Store (Amazon EBS) storage. If using Amazon EBS storage the storage should be RAID of Amazon EBS volumes and mounted with the XFS file system for it to be a supported configuration.

Resolved Issues in Greenplum Database 4.3.5.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.x

Issue Number	Category	Resolved In	Description
25766	Query Optimizer	4.3.5.3	For some queries, the Pivotal Query Optimizer incorrectly converted a left outer join operation to an inner join operation during query optimization when a predicate contained a `COALESCE` function with constant arguments. This issue has been resolved.
25743	Management Scripts: analyzedb	4.3.5.3	The Greenplum Database utility `analyzedb` returned an error if the utility encountered a table or schema that contained uppercase and lowercase characters. This issue has been resolved.
25726	Query Optimizer	4.3.5.3	For some queries that have both outer joins and a predicate that contained `COALESCE`, the Pivotal Query Optimizer returned incorrect results. This issue has been resolved.
25722	Query Optimizer	4.3.5.3	For some queries that contained computed columns in a subquery that were not used in the main query, Pivotal Query Optimizer generated an execution plan that contained the unused computed columns. Now, Pivotal Query Optimizer generates a more efficient plan that does not contain unused computed columns.
25721	Query Optimizer	4.3.5.3	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. See *Changed Feature*.
25707	Query Execution	4.3.5.3	Pivotal Query Optimizer returned incorrect results for queries against partitioned tables that were partitioned by timestamp, and the query predicate for the partition selection was of type `date`.
25700	Query Optimizer	4.3.5.3	If the `WHERE` clause of a query compares a literal to a non-integer distribution column, and if the datatype of the literal differs from that of the distribution column, the Pivotal Query Optimizer might have generated a query execution plan that performed a dispatch to an incorrect segment. This dispatching problem caused incorrect results to be returned. This issue has been resolved.

Issue Number	Category	Resolved In	Description
25697	Query Optimizer	4.3.5.3	For queries that contain a `UNION` or `UNION ALL` over multiple subexpressions, Pivotal Query Optimizer generated an execution plan with cascaded `UNION` or `UNION ALL` operators. Now, a more efficient plan is generated with a single n-ary `UNION` or `UNION ALL` operator
25666	Query Execution	4.3.5.3	Greenplum Database supports `INSERT` operations into the leaf child partition table of a partitioned table. See *Product Enhancements*.
24438	Query Optimizer	4.3.5.3	For queries that contain a `GROUP BY` clause that was used to group the results of a `UNION` or `UNION ALL` clause over more than two subexpressions, Pivotal Query Optimizer generated an execution plan that contained cascaded `UNION` or `UNION ALL` operators. Now, the `GROUP BY` operation is pushed below the `UNION` or `UNION ALL` operator.
25667	Security	4.3.5.2	Greenplum Database software has been updated to use OpenSSL 0.9.8zg. For information about major changes in OpenSSL 0.9.8zg, see *http://www.openssl.org/news/openssl-0.9.8-notes.html*.
25661	Query Execution	4.3.5.2	Greenplum Database could not successfully execute some parametrized queries that included indexed columns where the indexed columns were queried but the parameters did not reference the indexed columns. This issue has been resolved.
25643	Query Optimizer	4.3.5.2	For some queries that contained a computed column in a `GROUP BY` clause, Greenplum Database generated an execution plan that incorrectly pulled the computed column above the `GROUP BY` operation. This caused a Greenplum Database PANIC. This issue has been resolved.
25637	Query Execution	4.3.5.2	Queries returned incorrect results if the query execution plan rescanned a bitmap index under a subplan where the bitmap was filtered using parameters from the outer plan.
25590	Interconnect	4.3.5.2	When no sockets were available on the Greenplum Database master, the message displayed was incorrect.
25589	Interconnect	4.3.5.2	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.

Issue Number	Category	Resolved In	Description
25588	Management Scripts: expansion	4.3.5.2	The Greenplum Database `gpexpand` utility performance has been enhanced.
25579	Query Optimizer	4.3.5.2	Some queries on partitioned tables caused a Greenplum Database PANIC if the query contains both of the following features: • An `IN` predicate that includes the table partitioning key • A subquery whose output column is the same partitioning key from the outer query This issue has been resolved.
25576	Query Optimizer	4.3.5.2	For some star join queries with a large number of dimensions, the Pivotal Query Optimizer ran out of memory during the recursive processing of intermediate expressions. This issue has been resolved by reducing the number of recursive steps performed on large expressions.
25516	Management Scripts:	4.3.5.2	When shutting down a Greenplum Database, the Greenplum Database utility `gpstop` 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 `gpstop`, such as `gpexpand`, to fail because of the incorrect failure error code from `gpstop`. The `gpstop` utility now returns the correct code.
90922060	Query Execution	4.3.5.1	For queries where aggregates with distinct arguments are used as window functions, the query plan could have produced wrong results due to a limitation of the Window operator. This issue has been resolved.
84334744	Query Optimizer	4.3.5.1	For some queries, an error occurs when the query contains a `WITH` clause (a common table expression) that references a column multiple times. In this example the column `a` is referenced multiple times: `WITH x AS (SELECT a AS a1, a AS a2 FROM t1) ...` This issue has been resolved.
25537	Query Execution	4.3.5.1	In some cases when the Pivotal Query Optimizer is enabled, a memory management issue occurred during a DynamicTableScan or DynamicIndexScan operation and caused a Greenplum Database PANIC.

Issue Number	Category	Resolved In	Description
25502	Storage: Access Methods	4.3.5.1	A Greenplum Database segment encountered an out of memory issue that caused a segmentation fault during the cleanup process. The segmentation fault caused a failover to a segment mirror. The out of memory issue no longer results in a failover.
25490	Query Planner	4.3.5.1	Some SQL queries with nested subqueries returned this error: ERROR: Failed to locate datatype for paramid 2
25485	Monitoring: gpperfmon server	4.3.5.1	If the password for the Greenplum Command Center user `gpmon` is not listed in the `.pgpass` file, the `gpmmon` process hung and users could not access Greenplum Command Center.
25484	DDL and Utility Statements, Storage: Transaction Management	4.3.5.1	Some queries were not handled properly by the Greenplum Database query dispatcher. This caused a PANIC on a Greenplum Database segment.
25476	Loaders: gpfdist	4.3.5.1	When running the Greenplum Database utility `gpfdist`, the size of log files increased quickly in some situations. In Greenplum Database 4.3.5.1, the `gpfdist` utility supports the `-s` option to help minimize the increase of `gpfdist` log files. The `-s` option enables simplified logging. Only messages with WARN level and higher are written to the log file.
25472	Management Scripts: gpcheckcat	4.3.5.1	In some cases, the Greenplum Database utility `gpcheckcat` required a significant amount of time when it detected inconsistent OID (object ID) information and generated output about the inconsistencies. In Greenplum Database 4.3.5.1, the `gpcheckcat` utility has been enhanced to reduce the time required when inconsistent OID information is detected. The utility creates verification files that contain a query that generates information about the inconsistent OIDs.
25464	Functions and Languages	4.3.5.1	Greenplum Database calculated the incorrect time for some queries due to an old version of a timezone file. The Greenplum Database timezone files have been updated.

Issue Number	Category	Resolved In	Description
25455	Storage: Segment Mirroring	4.3.5.1	When running Greenplum Database utility `gprecoverseg` to perform an incremental segment recovery (the `-F` option was not specified), performance was poor if the database contained a large number of objects and a large number of `DROP` operations were performed. The performance of `gprecoverseg` has been improved. **Note:** The performance of `gprecoverseg` is not affected if the `-F` option is specified to perform a full segment recovery.
25421	Management Scripts: General	4.3.5.1	The Greenplum Database upgrade script `fix_ao_upgrade.py` failed on databases that do not contain the schema *public*. This issue has been resolved.
25395	DDL and Utility Statements	4.3.5.1	When a Greenplum Database superuser issued a `REVOKE` command that did not cause a change the database, a warning was not issued. Now, Greenplum Database displays a message that no privileges were revoked.
25386	Catalog and Metadata	4.3.5.1	The Greenplum Database utility `gpcheckcat` incorrectly issued messages for inconsistencies in the catalog for partitioned append-optimized tables. The `gpcheckcat` utility no longer issues these messages.
25357	DDL and Utility Statements	4.3.5.1	In some cases, when running an ETL program against Greenplum Database, SQL DML statements such as `SELECT`, `DELETE`, `INSERT`, and `UPDATE` returned the warning `unrecognized node type: 701`.
25306	Backup and Restore	4.3.5.1	The Greenplum Database utility `pg_dump` did not back up aggregate functions correctly.
25158	Management Scripts: General	4.3.5.1	In some cases when Greenplum Database email alerts were enabled, the logger process crashed when trying to send an email. This resulted in fragmented `pg_log` files. This issue has been resolved.

Issue Number	Category	Resolved In	Description
13685	Catalog and Metadata	4.3.5.1	In some cases, when an `ALTER TABLE` command that contained an `EXCHANGE PARTITION` clause was run to exchange table partitions, Greenplum Database did not use the same OID (object ID) for the related entry in the *pg_constraint* system catalog table on all the Greenplum Database segments. The Greenplum Database `gpcheckcat` utility reported the inconstancy as an error when some database catalog checks were performed. A consistent OID is now used.
11575	Catalog and Metadata	4.3.5.1	In some cases, when a `CREATE INDEX` command that contained a `WHERE` clause was run to create a partial index, Greenplum Database did not use the same OID (object ID) for the related entry in the *pg_index* system catalog table on all the Greenplum Database segments. The Greenplum Database `gpcheckcat` utility reported the inconstancy as an error when some database catalog checks were performed. A consistent OID is now used.
11289	Catalog and Metadata	4.3.5.1	In some cases, when performing `CREATE TABLE` or `ALTER TABLE` operations that include a default column value, Greenplum Database did not use the same OID (object ID) for the related entry the *pg_attrdef* system catalog table on all the Greenplum Database segments. The Greenplum Database `gpcheckcat` utility reported the inconstancy as an error when some database catalog checks were performed. A consistent OID is now used.
90561896	Management Scripts: recoverseg	4.3.5.0	In Greenplum Database 4.3.4.1, the Greenplum Database `gprecoverseg` 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 `gprecoverseg`. To check for persistent table issues, use the Greenplum Database `gpcheckcat` utility. The behavior in Greenplum Database has been reverted to the behavior in 4.3.4.0 and earlier.
89931274	Security	4.3.5.0	Greenplum Database software has been updated to use OpenSSL 0.9.8ze. For information about major changes in OpenSSL 0.9.8ze, see *http://www.openssl.org/news/openssl-0.9.8-notes.html*.

Issue Number	Category	Resolved In	Description
87808098	Loaders	4.3.5.0	In some cases, the Greenplum Database utility `gpfdist` 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	4.3.5.0	In some cases, running the Greenplum Database utility `gpcrondump` caused a PANIC on some Greenplum Database segments.
25422	Management Scripts: General	4.3.5.0	The Greenplum Database utility `gprecoverseg` returns an error when the `PGPORT` environment variable is not set.
25417	Monitoring: gpperfmon server	4.3.5.0	In some cases when the Greenplum Command Center is installed, the `gpsmon` process timed out after one hour and returned an error that no request were received after 3600 seconds.
25339	Query Execution	4.3.5.0	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.
25335	Catalog and Metadata, Global Persistent Objects	4.3.5.0	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	4.3.5.0	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	4.3.5.0	The Greenplum Database utility `gpmfr` failed when all the files that were being backed up were less than 1000 bytes.
25297	Query Optimizer	4.3.5.0	Some queries returned wrong results when an `IN` clause contained a nested expression. This issue has been resolved.
25296	Query Optimizer	4.3.5.0	Some queries that contained aggregate functions were terminated by an error because of incorrect plans within a window function. This issue has been resolved.

Issue Number	Category	Resolved In	Description
25292 25361	Query Optimizer	4.3.5.0	Inefficient plans were generated for queries that contained the function `unnest(ARRAY[...])`. The plan generated by the Pivotal Query Optimizer for this type query has been improved.
25288	Query Execution	4.3.5.0	A Greenplum Database PANIC occurred when deleting data from a table with the `DELETE` command if the contains a `USING` clause.
25279	Management Scripts: gpstart/ gpstop	4.3.5.0	In some cases, the Greenplum Database utility `gpstop` issued the warning `No leftover gpmmon process found`. These warning messages have be changed to informational messages.
25252	Query Optimizer	4.3.5.0	Some queries that required partition elimination with a `NOT IN` predicate caused a crash. This issue has been resolved.
25175	Query Execution	4.3.5.0	A Greenplum Database PANIC occurred when using the `COPY` command to copy data into a table that contains no columns.
25170	Storage: Vacuum	4.3.5.0	In some cases, running the Greenplum Database utility `vacuumdb` caused a Greenplum Database PANIC due to issues with a system table that is used to track append optimized file segments.
25160	Query Execution	4.3.5.0	In some cases, running a query and concurrently performing a DDL operation on the same data returned this error. `ERROR", "XX000","could not open relation`
25124	Dispatch	4.3.5.0	If a cursor was declared in a transaction, and then a `SET` command was issued in the same transaction before the cursor was closed, a Greenplum Database crash occurred. In Greenplum Database 4.3.5.0, an error is returned if the `SET` command is issued while a cursor is declared and not closed in a transaction. In the transaction, the cursor must be closed before the `SET` command can be issued.
25081	Interconnect	4.3.5.0	In some cases, when a `COPY` command that contains a sub-select returns an error, Greenplum Database generated a segmentation fault.

Issue Number	Category	Resolved In	Description
24953	Management Scripts: gptoolkit	4.3.5.0	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. The Greenplum Database function `__gp_aovisimap_compaction_info` displays append-optimized table on-disk storage and bloat information. See "The gp_toolkit Administrative Schema" in the *Greenplum Database Reference Guide*.
24944	DDL and Utility Statements	4.3.5.0	The `set_config()` 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	4.3.5.0	The Greenplum Database function `to_date()` did not validate the range of the input date.
24591	Backup and Restore	4.3.5.0	In some cases, the Greenplum Database utility `gpcrondump` failed with the error `Cannot allocate memory`. The memory management of the Greenplum Database utility has been enhanced to minimize occurrence of the error.
24557	Query Optimizer	4.3.5.0	Some queries with aggregate functions that contained outer references returned the error message: `aggref found in non-Agg plan node`. This issue has been resolved.
24263	Query Optimizer	4.3.5.0	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	4.3.5.0	For a table with a primary key, the `ALTER TABLE` command could change the distribution policy (the columns specified with the `DISTRIBUTION KEY` clause) to a non-primary key. Specifying a `DISTRIBUTION KEY` to a non-primary key column is not supported. This issue has been resolved.

Issue Number	Category	Resolved In	Description
18673	DDL and Utility Statements	4.3.5.0	In some cases, SQL commands that were executed concurrently with an `ALTER TABLE` command that contains a `SPLIT PARTITION` clause on a partitioned table returned this error: `ERROR: Relation decrement reference count found relation relation-id with bad count`

Known Issues in Greenplum Database 4.3.5.x

This section lists the known issues in Greenplum Database 4.3.5.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
25684	Query Execution	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.
12019	Management Scripts: checkperf	When the Greenplum Database `gpcheckperf` utility is run with the option `-f host_file` and the host that is running `gpcheckperf` is listed in *host_file*, processes that were started `gpcheckperf` might not be cleaned up after the utility completes. **Workaround**: Manually stop the processes that were started by `gpcheckperf`.
90799642	Query Optimizer	For queries that include `DISTINCT` aggregates expressed as window functions, the query might return wrong results because the `DISTINCT` qualifier is incorrectly dropped in the window operator.
25147	Query Optimizer	When changing a table definition with the `ALTER TABLE` command, the `REORGANIZE` clause cannot be specified when the distribution policy of the table is being changed to random distribution (with the `DISTRIBUTED RANDOMLY` 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.

Issue	Category	Description
23571	Query Optimizer	For queries that contain inequality conditions such as `!=` , `<` and , `>`, 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.
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 `log_hostname` to `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 `on` only on the Greenplum Database master.
25280	Management Scripts: gpstart/ gpstop	The Greenplum Database utility `gpstop`, the utility returns an error if it is run and the system environment variable `LANG` is set, for example, `export LANG=ja_JP.UTF-8`. **Workaround:** Unset the environment variable `LANG` before running the `gpstop` utility. For example: `$ unset LANG`

Issue	Category	Description

25246	Management Scripts: gpconfig	When you set the server configuration parameters `gp_email_to` and `gp_email_from` with the Greenplum Database utility `gpconfig`, the utility removes the single quotes from the values.
		`$ gpconfig -c gp_email_to -v 'test@my-email.com'`
		The improperly set parameter causes Greenplum Database to fail when it is restarted.
		Workaround: Enclose the value for `gp_email_to` or `gp_email_from` with double quotes.
		`$ gpconfig -c gp_email_to -v "'test@my-email.com'"`

25168	Locking, Signals, Processes	When the server configuration parameter `client_min_messages` is set to either set to `PANIC` or `FATAL` and a `PANIC` or `FATAL` level message is encountered, Greenplum Database hangs. The `client_min_messages` parameter should not be set a value higher than `ERROR`.
24588	Management Scripts: gpconfig	The Greenplum Database `gpconfig` utility does not display the correct information for the server configuration parameter `gp_enable_gpperfmon`. The parameter displays the state of the Greenplum Command Center data collection agents (`gpperfmon`). **Workaround:** The SQL command `SHOW` displays the correct `gp_enable_gpperfmon` value.
24031	gphdfs	If a readable external table is created with `FORMAT 'CSV'` and uses the gphdfs protocol, reading a record fails if the record spans multiple lines and the record is stored in multiple HDFS blocks. **Workaround:** Remove line separators from within the record so that the record does not span multiple lines.
23824	Authentication	In some cases, LDAP client utility tools cannot be used after running the source command: `source $GPHOME/greenplum_path.sh` because the LDAP libraries included with Greenplum Database are not compatible with the LDAP client utility tools that are installed with operating system. **Workaround:** The LDAP tools can be used without running the source command in the environment.
23525	Query Planner	Some SQL queries that contain sub-selects fail with this error. `ERROR: Failed to locate datatype for paramid 0`

Issue	Category	Description
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the `gp_adjust_priority()` function. The attempt to execute this request might silently fail. The return value of the `gp_adjust_priority()` 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 `gp_adjust_priority()` call.
23492	Backup and Restore,	A backup from a Greenplum Database 4.3.x system that is created with a Greenplum Database back up utility, for example gpcrondump, cannot be restored to a Greenplum Database 4.2.x system with the `psql` utility or the corresponding restore utility, for example gpdbrestore.
23521	Client Access Methods and Tools	Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database. **Workaround:** For Hadoop distributions based on Hadoop 2.2 or later that are supported by Greenplum Database, the classpath environment variable and other directory paths defined in `$GPHOME/lib/hadoop/hadoop_env.sh` must be to be modified so that the paths point to the appropriate JAR files.

20453	Query Planner	For SQL queries of either of the following forms:
		`SELECT columns FROM table WHERE table.column NOT IN subquery;`
		`SELECT columns FROM table WHERE table.column = ALL subquery;`
		tuples that satisfy both of the following conditions are not included in the result set:
		• *table.column* is `NULL`.
		• *subquery* returns the empty result.

21838	Backup and Restore	When restoring sets of tables with the Greenplum Database utility gpdbrestore, 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. **Workaround:** Before restoring a set of tables, create the schemas for the tables in the database.
21129	DDL and Utility Statements	SSL is only supported on the master host. It is not supported on segment hosts.
20822	Backup and Restore	Special characters such as `!`, `$`, `#`, and `@` cannot be used in the password for the Data Domain Boost user when specifying the Data Domain Boost credentials with the `gpcrondump` options `--ddboost-host` and `--ddboost-user`.

Issue	Category	Description
18247	DDL and Utility Statements	`TRUNCATE` 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 `TRUNCATE` command, the command does not remove rows from the sub-table and its child tables. **Workaround:** Use the `ALTER TABLE` command with the `TRUNCATE PARTITION` clause to remove rows from the sub-table and its child tables.
19705	Loaders: gpload	`gpload` fails on Windows XP with Python 2.6. **Workaround:** Install Python 2.5 on the system where gpload is installed.
19493 19464 19426	Backup and Restore	The `gpcrondump` and `gpdbrestore` utilities do not handle errors returned by DD Boost or Data Domain correctly. These are two examples: • If invalid Data Domain credentials are specified when setting the Data Domain Boost credentials with the `gpcrondump` utility, the error message does not indicate that invalid credentials were specified. • Restoring a Greenplum database from a Data Domain system with `gpdbrestore` and the `--ddboost` option indicates success even though segment failures occured during the restore. **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.
15692 17192	Backup and Restore	Greenplum Database's implementation of RSA lock box for Data Domain Boost changes backup and restore requirements for customers running SUSE. The current implementation of the RSA lock box for Data Domain Boost login credential encryption only supports customers running on Red Hat Enterprise Linux. Workaround: If you run Greenplum Database on SUSE, use NFS as your backup solution. See the *Greenplum Database Administrator Guide* for information on setting up a NFS backup.
18850	Backup and Restore	Data Domain Boost credentials cannot be set up in some environments due to the absence of certain libraries (for example, `libstdc++`) expected to reside on the platform. **Workaround:** Install the missing libraries manually on the system.
18851	Backup and Restore	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.

Issue	Category	Description
18713	Catalog and Metadata	Drop language plpgsql cascade results in a loss of `gp_toolkit` functionality. **Workaround:** Reinstall `gp_toolkit`.
18710	Management Scripts Suite	Greenplum Management utilities cannot parse IPv6 IP addresses. **Workaround:** Always specify IPv6 hostnames rather than IP addresses
18703	Loaders	The bytenum field (byte offset in the load file where the error occurred) in the error log when using gpfdist with data in text format errors is not populated, making it difficult to find the location of an error in the source file.
12468	Management Scripts Suite	`gpexpand --rollback` fails if an error occurs during expansion such that it leaves the database down `gpstart` also fails as it detects that expansion is in progress and suggests to run `gpexpand --rollback` which will not work because the database is down. **Workaround:** Run `gpstart -m` to start the master and then run rollback.
18785	Loaders	Running `gpload` with the `--ssl` option and the relative path of the source file results in an error that states the source file is missing. **Workaround:** Provide the full path in the yaml file or add the loaded data file to the certificate folder.
18414	Loaders	Unable to define external tables with fixed width format and empty line delimiter when file size is larger than `gpfdist` chunk (by default, 32K).
17285	Backup and Restore	NFS backup with `gpcrondump -c` can fail. In circumstances where you haven't backed up to a local disk before, backups to NFS using `gpcrondump` with the `-c` option can fail. On fresh systems where a backup has not been previously invoked there are no dump files to cleanup and the `-c` flag will have no effect. **Workaround:** Do not run `gpcrondump` with the `-c` option the first time a backup is invoked from a system.
17837	Upgrade/ Downgrade	Major version upgrades internally depend on the `gp_toolkit` system schema. The alteration or absence of this schema may cause upgrades to error out during preliminary checks. **Workaround:** To enable the upgrade process to proceed, you need to reinstall the `gp_toolkit` schema in all affected databases by applying the SQL file found here: `$GPHOME/share/postgresql/gp_toolkit.sql`.

Issue	Category	Description
17513	Management Scripts Suite	Running more than one `gpfilespace` command concurrently with itself to move either temporary files (`--movetempfilespace`) or transaction files (`--movetransfilespace`) to a new filespace can in some circumstances cause OID inconsistencies. **Workaround:** Do not run more than one `gpfilespace` command concurrently with itself. If an OID inconsistency is introduced `gpfilespace --movetempfilespace` or `gpfilespace --movetransfilespace` can be used to revert to the default filespace.
17780	DDL/DML: Partitioning	`ALTER TABLE ADD PARTITION` inheritance issue When performing an `ALTER TABLE ADD PARTITION` operation, the resulting parts may not correctly inherit the storage properties of the parent table in cases such as adding a default partition or more complex subpartitioning. This issue can be avoided by explicitly dictating the storage properties during the `ADD PARTITION` invocation. For leaf partitions that are already afflicted, the issue can be rectified through use of `EXCHANGE PARTITION`.
17795	Management Scripts Suite	Under some circumstances, `gppkg` on SUSE is unable to correctly interpret error messages returned by rpm. On SUSE, `gppkg` is unable to operate correctly under circumstances that require a non-trivial interpretation of underlying rpm commands. This includes scenarios that result from overlapping packages, partial installs, and partial uninstalls.
17604	Security	A Red Hat Enterprise Linux (RHEL) 6.x security configuration file limits the number of processes that can run on gpadmin. RHEL 6.x contains a security file (/etc/security/limits.d/90-nproc.conf) that limits available processes running on gpadmin to 1064. **Workaround:** Remove this file or increase the processes to 131072.
17334	Management Scripts Suite	You may see warning messages that interfere with the operation of management scripts when logging in. Greenplum recommends that you edit the `/etc/motd` file and add the warning message to it. This will send the messages to are redirected to stdout and not stderr. You must encode these warning messages in UTF-8 format.
17221	Resource Management	Resource queue deadlocks may be encountered if a cursor is associated with a query invoking a function within another function.
17113	Management Scripts Suite	Filespaces are inconsistent when the Greenplum database is down. 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.

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

Upgrading to Greenplum Database 4.3.5.x

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.5.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 4.3.5.x, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later.

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

- *Upgrading from 4.3.x to 4.3.5.x*
- *Upgrading from 4.3.x to 4.3.5.x on Pivotal DCA Systems*
- *Upgrading from 4.2.x.x to 4.3.5.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:

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

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

Upgrading from 4.3.x to 4.3.5.x

An upgrade from 4.3.x to 4.3.5.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 4.3.5.x on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.5.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 4.3.4.1 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.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-4.3.5.3
```

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-4.3.5.3/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-4.3.5.3 /usr/local/greenplum-db
```

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

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

7. Run the `gpseginstall` utility to install the 4.3.5.x binaries on all the segment hosts specified in the *hostfile*. For example:

```
$ gpseginstall -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. 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 4.3.5.3 as follows:

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

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

fix_ao_upgrade.py Utility

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

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

Syntax

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

Options

-r | --report

Report inconsistencies without making any changes.

-h *master_host* | --host=*master_host*

Greenplum Database master hostname or IP address.

-p *master_port* | --port=*master_port*

Greenplum Database master port.

-u *user* | --user=*user*

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

v | --verbose

Verbose output that includes table names.

--help

Show the help message and exit.

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

Upgrading from 4.3.x to 4.3.5.x on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.5.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 4.3.5.x on DCA systems. This section is only for installing Greenplum Database 4.3.5.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 4.3.4.1 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.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.x.

```
# ./greenplum-db-appliance-4.3.5.3-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 4.3.5.x

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.5.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 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-4.3.5.3-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.5.3-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-4.3.5.3`), 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.x installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.5.3/greenplum_path.sh
```

8. Run the `gpseginstall` utility to install the 4.3.5.x binaries on all the segment hosts specified in the *hostfile*. For example:

```
$ gpseginstall -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 4.3.5.3.

```
# ./greenplum-db-appliance-4.3.5.3-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.x installation. For example:

   ```
   $ source /usr/local/greenplum-db-4.3.5.3/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.x installation.
   a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.5.3 installation directory. For example (as root):

      ```
      # rm -rf /usr/local/greenplum-db
      # ln -s /usr/local/greenplum-db-4.3.5.3 /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-4.3.5.3 /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-4.3.5.3 /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-4.3.5.3 /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-4.3.5.3
```

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-4.3.5.3`.

8. As `gpadmin`, run the 4.3.5.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-4.3.5.3
```

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-4.3.5.3`.

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

9. The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.3.5.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)

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

```
$ gpinitstandby -s standby_hostname
```

2. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the compatible Greenplum Database 4.3.5 Load Tools package. Application Packages are available at *Pivotal Network*. For information about `gpfdist`, see the *Greenplum Database 4.3 Administrator Guide*.
3. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later. (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.
4. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, download the corresponding packages from *Pivotal Network*, and install using this utility. See the *Greenplum Database Utility Guide* for `gppkg` usage details.
5. If you want to utilize the Greenplum Command Center management tool, install the latest Command Center Console and update your environment variable to point to the latest Command Center binaries (source the `gpperfmon_path.sh` file from your new installation). See the Greenplum Command Center documentation for information about installing and configuring Greenplum Command Center.

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

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

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

```
$ gpstate
```

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

Pre-Upgrade Preparation (on your current system) * 4.2.x.x system is up and available	
☐	Log in to your master host as the `gpadmin` user (your Greenplum superuser).
☐	(Optional) Run `VACUUM` on all databases.
☐	(Optional) Remove old server log files from `pg_log` in your master and segment data directories.
☐	Check for and recover any failed segments (`gpstate`, `gprecoverseg`).
☐	Copy or preserve any additional folders or files (such as backup folders).
☐	Install the Greenplum Database 4.3 binaries on all Greenplum hosts.
☐	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	
☐	Backup your current databases.
☐	Remove the standby master (`gpinitstandby -r`).
☐	Do a clean shutdown of your current system (`gpstop`).
☐	Update your environment to source the new Greenplum Database 4.3.x installation.

☐	Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
☐	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	
☐	Reinitialize your standby master host (`gpinitstandby`).
☐	Upgrade `gpfdist` on all of your ETL hosts.
☐	Rebuild any custom modules against your 4.3.x installation.
☐	Download and install any Greenplum Database extensions.
☐	(Optional) Install the latest Greenplum Command Center and update your environment to point to the latest Command Center binaries.
☐	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 4.3.5.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 4.3.5.x*.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.5.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 4.3.5.x*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.5.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 4.3.5.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) 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

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 `gpdb4.3orca`
PL/Java	1.2 and release `gpdb4.3orca`
PL/Perl	1.2 and release `gpdb4.3orca`
PL/R	2.1 and release `gpdb4.3orca`
Pgcrypto	1.2 and release `gpdb4.3orca`
MADlib	1.9.3 and release `gpdb4.3orca`

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 `ossv`*version*. 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.

`pv`*pkg-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.

`gpdb`*rel-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`

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 4.3.5.3 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
Greenplum Database 4.3.5.3 Release Notes	A02
Greenplum Database 4.3 Installation Guide	A08
Greenplum Database 4.3 Administrator Guide	A10
Greenplum Database 4.3 Reference Guide	A10
Greenplum Database 4.3 Utility Guide	A11
Greenplum Database 4.3 Client Tools for UNIX	A05
Greenplum Database 4.3 Client Tools for Windows	A04
Greenplum Database 4.3 Connectivity Tools for UNIX	A03
Greenplum Database 4.3 Connectivity Tools for Windows	A03
Greenplum Database 4.3 Load Tools for UNIX	A07
Greenplum Database 4.3 Load Tools for Windows	A06
Greenplum Command Center 1.3 Administrator Guide	A04