Pivotal

RELEASE NOTES

Greenplum Database 4.3.3 Release Notes

Rev: A03

Updated: March, 2015

Welcome to Pivotal Greenplum Database 4.3.3

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 4.3.3. For previous versions of the release notes for Greenplum Database, go to *Pivotal Documentation* or EMC *Support Zone*.

About Greenplum Database 4.3.3

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

- *Product Enhancements*
- *Changed and Deprecated Features*
- *Downloading Greenplum Database*
- *Supported Platforms*
- *Resolved Issues in Greenplum Database 4.3.3*
- *Known Issues in Greenplum Database 4.3.3*
- *Upgrading to Greenplum Database 4.3.3*
- *Greenplum Database Tools Compatibility*
- *Greenplum Database Extensions Compatibility*
- *Hadoop Distribution Compatibility*
- *Greenplum Database 4.3.3 Documentation*

Product Enhancements

Greenplum Database 4.3.3 includes enhancements in these areas:

- *Managing Greenplum Database Objects and Data*

- *New Server Parameter to Control Reading External Data Error Limit*
- *Greenplum Database Extension Enhancements*
- *The gpload Utility Supports Table Schema Names*
- *External Table Support for Hadoop Distributions*

Managing Greenplum Database Objects and Data

- The `GRANT` and `REVOKE` commands support the `TRUNCATE` privilege on a table.

  You can use the `GRANT` and `REVOKE` commands to allow or prohibit a Greenplum Database user or role from removing all the rows in a table with the `TRUNCATE` command.

  For information about `GRANT` and `REVOKE`, and `TRUNCATE`, see the *Greenplum Database Reference Guide*.
- Capturing errors that occur from reading data from external data sources does not require an error table.

  You can capture formatting errors that occur from reading external data with the `COPY` command or with `CREATE EXTERNAL TABLE` by specifying the clause `LOG ERRORS INTO` *`error_table`*. In Greenplum Database 4.3.3, the clause `INTO` *`error-table`* is optional. If not specified, the errors are stored internally, not in a Greenplum Database error table.

  For errors that are stored internally in a log, use the built-in SQL function `gp_read_error_log()` to read the error log data. Use the built-in SQL function `gp_truncate_error_log()` to delete the error log data.

  For information about `COPY` and `CREATE EXTERNAL TABLE`, see the *Greenplum Database Reference Guide*. For information about external tables, see "Loading and Unloading Data" in the *Greenplum Database Administrator Guide*.

  > **Note:** For the `COPY` and `CREATE EXTERNAL TABLE` commands, the `INTO` *`error-table`* clause is deprecated.
- Greenplum Database 4.3.3 enables delta compression for compressed columns in append-optimized, column-oriented tables.

  For columns of type `BIGINT`, `INTEGER`, `DATE`, `TIME`, or `TIMESTAMP`, delta compression is also applied if the `COMPRESSTYPE` option is set to `RLE-TYPE` compression in an append-optimized, column-oriented table. The delta compression algorithm is based on the delta between consecutive column values and is designed to improve compression when data is loaded in sorted order or is when the compression is applied to column data that is in sorted order.

  For information about table compression, see the `CREATE TABLE` command in the *Greenplum Database Reference Guide*, and "Using Compression (Append-Optimized Tables Only)" in the *Greenplum Database Administrator Guide*

New Server Parameter to Control Reading External Data Error Limit

Greenplum Database stops processing input rows when you import data with the `COPY` command or from an external table if the first 1000 rows processed contain formatting errors. If a valid row is processed within the first *n* rows, Greenplum Database continues processing input rows. With Greenplum Database 4.3.3, you can change the default limit by setting the Greenplum Database server configuration parameter `gp_initial_bad_row_limit`.

For information about the server configuration parameter, see *New Server Configuration Parameter*. For information about `COPY` and `CREATE EXTERNAL TABLE`, see the *Greenplum Database Reference Guide*.

Greenplum Database Extension Enhancements

- The Greenplum Database PL/R 2.0 extension package uses R 3.1.0.

Greenplum Database 4.3.3 supports the PL/R 2.0 extension package that uses R 3.1.0. This page on the R web site describes the enhancements and changes to versions of R:

http://cran.r-project.org/src/base/NEWS.html

See the information in section *CHANGES IN R 3.1.0* and earlier sections for information about changes in R 3.1.0 and earlier.

For information about the PL/R Extension, see "Greenplum PL/R Extension" in the *Greenplum Database Reference Guide*. For information about extension compatibility, see *Greenplum Database Extensions Compatibility*.

- The Greenplum Database 4.3.3 installation includes the Greenplum Database Fuzzy String Match Extension.

  The Greenplum Fuzzy String Match extension provides functions to determine similarities and distance between strings based on various algorithms. The Fuzzy String Match extension is based on the PostgreSQL fuzzystrmatch module.

  For information about the Fuzzy String Match Extension, see "Greenplum Fuzzy String Match Extension" in the *Greenplum Database Reference Guide*.

Backup and Restore Enhancements

For Greenplum Database 4.3.3, the `gpcrondump` and `gpdbrestore` utilities have been enhanced:

- Support for Symantec NetBackup.

  For Greenplum Database running Red Hat Enterprise Linux, you can configure Greenplum Database to perform backup and restore operations with Symantec NetBackup. To perform a back up or restore with NetBackup, you configure Greenplum Database and NetBackup and then run a Greenplum Database `gpcrondump` or `gpdbrestore` utility.
- Incremental backup support for NetBackup and Data Domain.

  The `gpcrondump` and `gpdbrestore` utilities support incremental backup when you use a Symantec NetBackup system or a Data Domain system.

  For incremental back up sets, a full backup and associated incremental backups, the backup set must be on a single device. For example, a backup set must all be on a file system. The backup set cannot have some backups on the local file system and others on a Data Domain system or a NetBackup system.
- The `gpcrondump` utility supports customized email notification for backup operations.

  The `gpcrondump` utility can be configured to send out status email notifications after a backup operation completes. You can customize the email Subject and From lines of the email notifications that gpcrondump sends after a back up completes for a database.
- The `gpdbrestore` utility analyzes only restored tables.

  In previous releases, the `gpdbrestore` utility ran the `ANALYZE` command on all tables in the database. You can disable analyzing restored tables with the `gpdbrestore` option `--noanalyze`.

For information about the Greenplum Database utilities `gpcrondump` and `gpdbrestore`, see the *Greenplum Database Utility Guide*. For information about backing up and restoring Greenplum Database, see "Backing Up and Restoring Databases " in the *Greenplum Database Administrator Guide*.

The gpload Utility Supports Table Schema Names

The Greenplum Database utility `gpload` supports specifying a schema name for the external table objects that are created when a load job is run. You specify the schema name in the YAML file that controls the load job with the `EXTERNAL:SCHEMA` property. See the *Greenplum Database Utility Guide* for information about the `gpload` utility and the YAML control file.

External Table Support for Hadoop Distributions

With Greenplum Database external tables created with the `CREATE EXTERNAL TABLE` command, you can specify the `gphdfs` protocol to access external files on an Hadoop file system (HDFS) as if they are regular database tables. For Greenplum Database 4.3.3, the `gphdfs` protocol has been enhanced to support these Hadoop distributions:

- Pivotal Hadoop 2.0
- Hortonworks Data Platform (HDP) 2.1
- Cloudera 4.x and 5.x

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

Changed and Deprecated Features

- *Changed Features*
- *New Server Configuration Parameter*
- *Changed Server Configuration Parameter*
- *Deprecated Features*

Changed Features

- The Greenplum Database PL/R extension package has been updated to use R 3.1.0. The package version is `pv2.0`. For information about the PL/R extension enhancement, see *Greenplum Database Extension Enhancements*. For information about Greenplum Database PL/R extension package naming, see *Package File Naming Convention*.
- The Greenplum Database commands `COPY` and `CREATE EXTERNAL TABLE` have been enhanced. An error table is no longer required to capture formatting errors. For more information, see *Managing Greenplum Database Objects and Data*.
- The `gphdfs` protocol can access external files on an Hadoop file system (HDFS) as if they are regular database tables. The protocol has been enhanced to support additional Hadoop distributions. For more information, see *External Table Support for Hadoop Distributions*.

New Server Configuration Parameter

For Greenplum Database 4.3.3, the parameter `gp_initial_bad_row_limit` controls how Greenplum Database processes input rows when errors occur while reading data from external data sources.

Table 1: New Server Configuration Parameter in 4.3.3

Parameter Name	Value Range	Default Value	Description	Set Classifications
gp_initial_bad_row_limit	0 - `INT_MAX`	`1000`	For the parameter value *n*, Greenplum Database stops processing input rows when you import data with the `COPY` command or from an external table if the first *n* rows processed contain formatting errors. If a valid row is processed within the first *n* rows, Greenplum Database continues processing input rows. Setting the value to 0 disables this limit. The `SEGMENT REJECT LIMIT` clause can also be specified for the `COPY` command or the external table definition to limit the number of rejected rows. `INT_MAX` is the largest value that can be stored as an integer on your system.	master session reload

Changed Server Configuration Parameter

For Greenplum Database 4.3.3, the parameter `gp_hadoop_target_version` supports the value `hdp2` for the Hortonworks distribution of HDFS.

Table 2: Changed Server Configuration Parameter in 4.3.3

Parameter Name	Value Range	Default Value	Description	Set Classifications
gp_hadoop_target_version	gphd-1.0 gphd-1.1 gphd-1.2 gphd-2.0 gpmr-1.0 gpmr-1.2 hdp2 cdh3u2 cdh4.1	gphd-1.1	The installed version of Greenplum Hadoop target.	local session reload

For information about supported Hadoop distributions, see *Hadoop Distribution Compatibility*.

Deprecated Features

Pivotal plans to deprecate the following items:

- For the `COPY` and `CREATE EXERNAL TABLE` commands, the `INTO` *`error-table`* clause is deprecated and will not be supported in a future release. Only internal error logs will be supported.
- The `gpsnmpd` utility is deprecated.
- The `gpdetective` utility is deprecated.
- The Greenplum Database utilities `gp_dump` and `gp_restore` are not supported and have been removed from the documentation.

Please send any questions or comments about the deprecated items to *gpdb@gopivotal.com*.

Downloading Greenplum Database

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

- Greenplum Database 4.3.x software is available from *Pivotal Network*.
- Current release Greenplum Database documentation is available from the *Pivotal Documentation* site.

  Previous release versions of Greenplum Database documentation, as well as other Greenplum Database documents, are available from *Support Zone*

Supported Platforms

Greenplum Database 4.3.3 runs on the following platforms:

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

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

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

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

Table 3: Data Domain Boost Compatibility

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

Greenplum Database support on DCA:

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

Resolved Issues in Greenplum Database 4.3.3

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

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

Table 4: Resolved Issues in 4.3.3

Issue Number	Category	Description
24479	Backup and Restore	A table could not be restored (with the `gpdbrestore -T` option) from a back up that is on a Data Domain Boost system and that was created with the `gpcrondump --ddboost` options.
24478	Management Scripts: expansion	The Greenplum Database `gpexpand` utility failed when an error table for an external table was present in Greenplum Database. The utility displayed this message: DETAIL: ALTER TABLE is not allowed on error tables
24326	Query Execution, Storage Access Methods	If either a non-partitioned append-only table or an individual append-only part of a partitioned table had more than 127 million rows on a segment, a query that uses an index to access the table data could return duplicate rows. This issue has been fixed.

Issue Number	Category	Description
24317	Security	Greenplum Database software has been updated to use OpenSSL 0.9.8zb in response to the OpenSSL Security Advisory [6 Aug 2014]. For information about the advisory, see *https://www.openssl.org/news/secadv_20140806.txt*.
24248	GPHDFS	The Greenplum Database external table protocol gphdfs supports the Cloudera 4.x and 5.x HDFS distributions. See *External Table Support for Hadoop Distributions*.
24237	DDL and Utility Statements	Temporary tables were not cleaned up properly in the following situation. A user defined function (UDF) was created with a security definer and that included statements to create the temporary table. The UDF was executed by a regular user who was given EXECUTE permission on the function. This caused the temporary table to stay in the database after the session was disconnected.
24182	Management Scripts: General	Greenplum Database timezone information has been updated to match world-wide timezones. For in formation about timezones, see *http://www.iana.org/time-zones*.
24168	Vacuum	For an append-optimized table that did not contain any data, the `VACUUM` command did not update the value of relfrozenxid in the catalog table *pg_class*.
24158	Upgrade / Downgrade	When upgrading Greenplum Database from a 4.2.x release to a 4.3.x release prior to 4.3.2.1, append-only tables were not correctly converted to append-optimized tables. In some cases, the incorrect conversion prevented the `VACUUM` command from reclaiming storage occupied by deleted tuples. For information about the upgrade issue, see *Upgrading from 4.3.x to 4.3.3*.
24119	Query Execution	In some cases, a segmentation fault occurred when a `DECLARE CURSOR WITH HOLD` command was run by an ODBC driver.
24116 21042	Loaders: gpfdist	The Greenplum Database `gpfdist` utility failed with a SIGSEGV error when the utility received a empty request with two consecutive return characters "\n\n".
24089	Loaders: Copy/ ExternalTabs	Multibyte characters were not handled properly when writing to an external table that uses the gb18030 encoding from a Greenplum database that was created with UTF8 encoding. In some cases, this error was encountered. ERROR: The size of the value cannot be bigger than the field size value
24079	GPHDFS	The Greenplum Database external table protocol gphdfs supports the Pivotal 2.0 distribution. See *External Table Support for Hadoop Distributions*.

Issue Number	Category	Description
24068	Postgis	When using PostGIS, In some cases a closed curved polygon that was converted to a linear polygon was not closed due to a linear approximation precision issue with PostGIS 2.0.3.
24067	Loaders: gpfdist, Loaders: gpload	In some cases when network load was heavy, the Greenplum Database utility gpfdist intermittently failed with this error: gpfdist closed connection to server
24055	Vacuum	The `VACUUM FULL` command transaction processing has been enhanced ensure proper operation with other concurrent operations.
24011	Catalog and Metadata, Vacuum	In some cases, when a `VACUUM FULL` command was cancelled, incorrect handling of Greenplum Database transaction log caused a PANIC signal to be issued and prevented Greenplum Database from performing a crash recovery of a segment mirror.
24001	Backup and Restore	During a backup operation, the Greenplum Database utility gpcrondump held an EXCLUSIVE lock on the catalog table pg_class longer than required.
23955	Query Execution	In some query plans, where a window operator is under the right child of a nested loops join, wrong results could have been generated because of improper cleanup of the operator's internal state.
23925	Management Scripts: expansion, Management Scripts: General	The Greenplum Database utilities gpactivatestandby and gpexpand used SSH to connect to localhost (the Greenplum Database host where the utility was run). Using SSH was redundant as the command was already on the local host and has been eliminated.
23894	Backup and Restore	Performing a back up to a Data Domain system failed when the Greenplum Database utility gpcrondump command specified the `--ddboost` options because gpcrondump performed a disk space check.
23864	Catalog and Metadata	Running the `REINDEX` command on a database while other workloads are concurrently running could create inconsistencies in the database catalog.
23850	Management Scripts	In some cases after expanding a Greenplum Database system, running `gpinitstandby -n` failed to resynchronize the data between the primary and standby master host.
23850	Management Scripts: General	In some cases after expanding a Greenplum Database system, running `gpinitstandby -n` failed to resynchronize the data between the primary and standby master host.
23842	Replication: Segment Mirroring	In some rare cases, if a restart occurred while the gprecoverseg utility was running, some tables and a persistent table were detected having less data on a mirror segment that corresponds to a primary segment.

Issue Number	Category	Description
23802	Query Execution	Greenplum Database did not manage temporary workfiles (spill files) properly. In some cases, this caused a query that required workfiles to fail with a message that stated that a Greenplum Database segment had reached the maximum configured workfile usage limit.
23753	Backup and Restore	The emails sent by the Greenplum Database `gpcrondump` utility could not be customized. Now the utility supports customized email notification for backup operations. See *Backup and Restore Enhancements*.
23730	Backup and Restore, Management Scripts: master mirroring	When configuring a Greenplum Database system with a standby master, the `gpinitstandby` utility did not correctly update the `pg_hba.conf` file on Greenplum Database segment hosts.
23729	Backup and Restore, DDL and Utility Statements	When the `-b` option was specified with the `gpcrondump` utility to disable a disk space check, a check was still performed.
23717	Locking, Signals, Processes	During Greenplum Database shutdown, a signal-unsafe function call was called from a signal handler function. The signal-unsafe function was replaced.
23699	Monitoring: gpperfmon server	Greenplum Database failed when the gpperfmon log files were not encoded in UTF8. This issue has been resolved.
23637	Backup and Restore	When restoring a Greenplum database with the Greenplum Database `gpcrondump` utility, the utility performed an `ANALYZE` operation on the entire database. Now the `gpcrondump` utility analyzes only the restored tables. See *Backup and Restore Enhancements*.
23568	Backup and Restore	When backing up a Greenplum database with the Greenplum Database `gpcrondump` utility and specifying an NFS directory with the `-u` option, the gpcrondump utility created an empty `db_dumps` directory in the master and segment data directories.
23558	Backup and Restore	When restoring a backup from a Data Domain system using `--ddboost` options, the Greenplum Database `gpdbrestore` utility failed because it could not find C data and post data files.
23286	Dispatch	In some cases, Greenplum Database did not handle the processing of cancelled distributed queries properly. This issue has been resolved.
22974	Loaders: Copy/ ExternalTabs	When reading data from external sources, Greenplum Database stopped reading data if the first 1000 rows processed contain formatting errors. Now the limit is configurable. See *New Server Parameter to Control Reading External Data Error Limit*.

Issue Number	Category	Description
20504	Query Execution	FOR loops in PL/pgSQL did not close the sequence generator if further access was still required.
18562	DDL and Utility Statements	A transaction lock did not block reader processes from proceeding when a writer process was holding the same lock. In some cases this caused a race condition to occur. Now, Greenplum Database blocks reader processes when a writer process holds the same lock to prevent race conditions from occurring.
17264	Replication: Segment Mirroring	In some cases, Greenplum Database continuously logged this message when sending file replication process statistics to Greenplum Database perfmon process: Error when sending file rep stats to perfmon
16450	Backup and Restore	When running the Greenplum Database utility `pg_dumpall` with the option `--resource-queues` to create scripts that contain resource queue definitions, the utility generated incorrect scripts when the resource queue definition contained the `memory_limit` option.
16059	Resource Management	Some SQL statements that executed a PL/pgSQL function that contained an insert, update, or delete operation did not allocate memory correctly. This caused the following issues: • Decreased performance • Large number of spill files were created on disk This issue has been resolved.

Known Issues in Greenplum Database 4.3.3

This section lists the known issues in Greenplum Database 4.3.3. 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 5: All Known Issues in 4.3.3

Issue	Category	Description
24383	gphdfs	Greenplum Database external tables do not support using the gphdfs protocol and MapR to access HDFS data.
24264	Catalog and Metadata	The commands `REINDEX TABLE` *`table_name`* and `REINDEX INDEX` *`index_name`* do not re-index child partition indexes of a partitioned table. **Workaround:** Run `REINDEX` on the child tables of the partitioned table.

Issue	Category	Description
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.
22798	Management Scripts: expansion, Management Scripts: master mirroring	If it is not possible to use SSH to connect from the Greenplum Database master host to 'localhost' a failure occurs when running the Greenplum Database `gpactivatestandby` or `gpexpand` utility because of an SSH failure. **Workaround:** Enable SSH to 'localhost' on the master host to work around this issue.
23646	DML	Running an `UPDATE` command after a `DROP COLUMN` and `ADD PARTITION` command on a partitioned table causes a Greenplum Database segment instance failure.
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.
23924	Backup and Restore	In some cases, performing some operations on an append-optimized table and then performing a full backup with the gpcrondump utility to a Data Domain system with DDBoost fails with this error: `ERROR: relation "file_name" does not exist`
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`
22792	Build	Greenplum Database is not certified on Red Hat Enterprise Linux 5.10.

Issue	Category	Description
22215	Build	Greenplum Database is not certified with these connectivity drivers: Data Direct v 7.022; PowerExchange for Greenplum 9.5.1 32-bit Microstrategy ODBC for Greenplum Wire Protocol 6.10.01.80 Open source ODBC 9.01.0100 and JDBC 9.1.902 Type 4 SAS/ACCESS 9.3 driver provided with SAS software2
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the `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.
21917	Replication: Segment Mirroring	In some rare cases after the Greenplum Database utility `gprecoverseg` was run, some append-optimized tables and a persistent table were detected having less data on a mirror segment corresponding to a primary segment.

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.

Issue	Category	Description
21724	Query Planner	Greenplum Database executes an SQL query in two stages if a scalar subquery is involved. The output of the first stage plan is fed into the second stage plan as a external parameter. If the first stage plan generates zero tuples and directly contributes to the output of the second stage plan, incorrect results might be returned.
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`.
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.
19788	Replication: Resync, Transaction Management	In some rare circumstances, performing a full recovery with `gprecoverseg` fails due to inconsistent LSN. **Workaround:** Stop and restart the database. Then perform a full recovery with `gprecoverseg`.
19705	Loaders: gpload	`gpload` fails on Windows XP with Python 2.6. **Workaround:** Install Python 2.5 on the system where gpload is installed.

Issue	Category	Description
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.
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.

Issue	Category	Description
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).
14640	Backup and Restore	`gpdbrestore` outputting incorrect non-zero error message. When performing single table restore, `gpdbrestore` gives warning messages about non-zero tables however prints out zero rows.
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`.
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.

Issue	Category	Description
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.
17415	Installer	When you run `gppkg -q -info<some gppkg>`, the system shows the GPDB version as main build dev.
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.
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.

Issue	Category	Description

16519	Backup and Restore	Limited data restore functionality and/or restore performance issues can occur when restoring tables from a full database backup where the default backup directory was not used.
		
		In order to restore from backup files not located in the default directory you can use the `-R` option to point to another host and directory. This is not possible however, if you want to point to a different directory on the same host (NFS for example).
		
		Workaround: Define a symbolic link from the default dump directory to the directory used for backup, as shown in the following example:
		
		1. Perform a full Database Backup to a specific NFS directory:
		`$ gpcrondump -x <db_name> -z -u /backup/DCA-93 -a`
		
		2. Create a file listing the segment servers:
		`$ vi /home/gpadmin/segments`
		`sdw1`
		`sdw2`
		`sdw3`
		`…`
		
		3. Remove the relevant date folder from db_dumps directories on the master and segments:
		`$ rm -r /data/master/gpseg-1/db_dumps/20120619`
		`$ gpssh -f segments 'rm -r /data1/primary/gpseg*/db_dumps/20120619'`
		`$ gpssh -f segments 'rm -r /data2/primary/gpseg*/db_dumps/20120619'`
		
		4. Create a symbolic link between the master and segment directories and the directory to which you backed up in step 1. Only the master and sdw1 was shown here, write a script for the remaining segments:
		`$ ln -s /backup/DCA-93/db_dumps/20120619/data/master/gpseg-1/db_dumps/20120619`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data1/primary/gpseg0/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data1/primary/gpseg1/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data1/primary/gpseg2/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data2/primary/gpseg3/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data2/primary/gpseg4/db_dumps/20120619'`

Issue	Category	Description

		$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619/data2/primary/gpseg5/db_dumps/20120619'
		5. Restore from backup files:
		$ gpdbrestore -t 20120619061835 -T <schema.table> -a
		6. Remove the symbolic links:
		$ rrm -r /data/master/gpseg-1/db_dumps/20120619
		$ gpssh -f segments 'rm -r /data1/primary/gpseg*/db_dumps/20120619'
		$ gpssh -f segments 'rm -r /data2/primary/gpseg*/db_dumps/20120619'

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 4.3.3

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.3. The minimum recommended upgrade path for this release is from Greenplum Database version 4.2.x.x. If you have an earlier major version of the database, you must first upgrade to version 4.2.x.x.

Prerequisites

Before starting the upgrade process, Pivotal recommends the following:

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

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

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

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

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

- *Upgrading from 4.3.x to 4.3.3*
- *Upgrading from 4.2.x.x to 4.3.3*
- *For Users Running Greenplum Database 4.1.x.x*
- *For Users Running Greenplum Database 4.0.x.x*
- *For Users Running Greenplum Database 3.3.x.x*
- *Migrating a Greenplum Database That Contains Append-Only Tables*

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

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

> **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 4.3.3

An upgrade from 4.3.x to 4.3.3 involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

> **Note:** If you are upgrading from Greenplum Database between 4.3.0 and 4.3.2, run the `fix_ao_upgrade.py` utility to check Greenplum Database for the upgrade issue and fix the upgrade issue (See step *9*). The utility is in this Greenplum Database 4.3.3 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:

   ```
   $ su - gpadmin
   ```

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

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

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

   ```
   $ gpstop
   ```

4. Run the installer for 4.3.3 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-4.3.3.0
   ```

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-4.3.3.0/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-4.3.3.0 /usr/local/greenplum-db
```

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

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

7. Run the `gpseginstall` utility to install the 4.3.3 binaries on all the segment hosts specified in the *hostfile*. For example:

```
$ gpseginstall -f hostfile
```

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

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

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

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

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

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

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

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

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

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

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

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

fix_ao_upgrade.py Utility

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

The upgrade process incorrectly converted append-only tables that were in the 4.2.x database to append-optimized tables during an upgrade from Greenplum Database 4.2.x to a Greenplum Database 4.3.x

release prior to 4.3.2.1. The incorrect conversion causes append-optimized table inconsistencies in the upgraded Greenplum Database system.

Syntax

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

Options

-r | --report

Report inconsistencies without making any changes.

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

Greenplum Database master hostname or IP address.

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

Greenplum Database master port.

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

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

v | --verbose

Verbose output that includes table names.

--help

Show the help message and exit.

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

Upgrading from 4.2.x.x to 4.3.3

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.3. 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.

Migrating a Greenplum Database That Contains Append-Only Tables

The migration process updates 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.

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

Upgrade Procedure

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

We have also provided you with an *Upgrade Checklist* that summarizes this procedure.

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

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

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

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

   ```
   $ su - gpadmin
   ```

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

   ```
   $ vacuumdb database_name
   ```

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

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

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

4. Run gpstate to check for failed segments.

   ```
   $ gpstate
   ```

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

   ```
   $ gprecoverseg
   ```

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

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

Install the Greenplum Database 4.3 Software Binaries

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

```
# unzip greenplum-db-4.3.3.0-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.3.0-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-4.3.3.0`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer installs the Greenplum Database software and creates a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Source the path file from your new 4.3.3 installation. For example:

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

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

```
$ gpseginstall -f hostfile
```

Upgrade Execution

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

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

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

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

```
$ gpinitstandby -r
```

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

```
$ gpstop
```

5. Source the path file from your new 4.3.3.0 installation. For example:

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

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

a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.3 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db

# ln -s /usr/local/greenplum-db-4.3.3.0 /usr/local/greenplum-db

# chown -R gpadmin /usr/local/greenplum-db
```

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

```
# gpssh -f segment_hosts_file

=> rm -rf /usr/local/greenplum-db

=> ln -s /usr/local/greenplum-db-4.3.3.0 /usr/local/greenplum-db

=> chown -R gpadmin /usr/local/greenplum-db

=> exit
```

7. (*optional but recommended*) Prior to running the migration, perform a pre-upgrade check to verify that your database is healthy by executing the 4.3.2 version of the `gpmigrator` utility with the `--check-only` option. For example:

```
# gpmigrator_mirror --check-only
   /usr/local/greenplum-db-4.2.6.3
   /usr/local/greenplum-db-4.3.3.0
```

8. As `gpadmin`, run the 4.3.2 version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ su - gpadmin
$ gpmigrator_mirror /usr/local/greenplum-db-4.2.6.3
   /usr/local/greenplum-db-4.3.3.0
```

> **Note:** If the migration does not complete successfully, contact Customer Support (see *Migrating a Greenplum Database That Contains Append-Only Tables*).

9. The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.3.2 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 4.3.3 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 4.3.3 Load Tools package. Application Packages are available at *Pivotal Network*.

3. Rebuild any custom modules against your 4.3.3 installation (for example, any shared library files for user-defined functions in `$GPHOME/lib`).

4. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, download the corresponding packages from *Pivotal Network*, and install using this utility. See the *Greenplum Database Utility Guide 4.3* for usage details.
5. If you want to utilize the Greenplum Command Center management tool, install the latest Command Center Console and update your environment variable to point to the latest Command Center binaries (source the `gpperfmon_path.sh` file from your new installation).

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

   Command Center Console packages are available from *Pivotal Network*.
6. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.3 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 4.3.3. Detailed upgrade instructions are provided in the *Known Issues in Greenplum Database 4.3.3* section.

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 4.3.3 installation.
☐ Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
☐ After the upgrade process finishes successfully, your 4.3.3 system will be up and running.
Post-Upgrade (on your 4.3 system)
* The 4.2.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 4.3.3 installation.
☐ Download and install any Greenplum Database extensions.
☐ (Optional) Install the latest Command Center Console 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 4.3.3.

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 4.3.3*.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.2.

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 4.3.3*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.2.

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 4.3.3*.

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 6: Greenplum Database Tools Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql) Greenplum MapReduce (gpmapreduce). See *Note*.	4.3	4.3
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.3	4.3

Client Package	Description of Contents	Client Version	Server Versions
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

Note: gpmapreduce is not available on Windows.

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

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

Greenplum GPText

GPText enables processing mass quantities of raw text data (such as social media feeds or e-mail databases) into mission-critical information that guides business and project decisions. GPText joins the Greenplum Database massively parallel-processing database server with Apache Solr enterprise search.

GPText requires Greenplum Database. See the GPText release notes for the required version of Greenplum Database.

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. Therefore, for the latest install and configuration information regarding any supported database package/extension, go to the *Support* site and download *Primus Article 288189* from our knowledge base (Requires a valid login to the EMC Support site).

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 7: Greenplum Database Extensions Compatibility

Greenplum Database Extension	Extension Components	
	Name	**Version**
PostGIS 2.0 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PostGIS 1.0 for Greenplum Database	PostGIS	1.4.2
	Proj	4.7.0
	Geos	3.2.2
PL/Java 1.1 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 2.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	3.1.0
PL/R 1.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.2 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 6.x 5.5.8 on RHEL 5.x, SUSE 10
PL/Perl 1.1 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
PL/Perl 1.0 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x, SUSE 10
Pgcrypto 1.1 for Greenplum Database 4.3.x.x	Pgcrypto	Based on PostgreSQL 8.3
MADlib 1.5 for Greenplum Database 4.3.x.x	MADlib	Based on MADlib version 1.8

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

Table 8: Greenplum Database 4.3 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.3
PL/Java	1.1
PL/Perl	1.2
PL/R	1.0
Pgcrypto	1.1
MADlib	1.5

Package File Naming Convention

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

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

This example is the package name for a postGIS package.

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

pkgname-ver - The package name and optional version of the software that was used to create the package extension. If the package is based on open source software, the version has format `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 specifies that the Greenplum Database package version is 2.0.

`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 9: Supported Hadoop Distributions

Hadoop Distribution	Version	gp_hadoop_ target_version
Pivotal HD	Pivotal HD 2.0 Pivotal HD 1.0 [1]	`gphd-2.0`
Greenplum HD	Greenplum HD 1.2	`gphd-1.2`
	Greenplum HD 1.1	`gphd-1.1` (default)

Hadoop Distribution	Version	gp_hadoop_ target_version
Cloudera	CDH 5.0, 5.1	`cdh4.1`
	CDH 4.1 [2] - CDH 4.7	`cdh3u2`
Hortonworks Data Platform	HDP 2.1	`hdp2`

Notes:

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

Greenplum Database 4.3.3 Documentation

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

Table 10: Greenplum Database Documentation

Title	Revision
Greenplum Database 4.3.3 Release Notes	A03
Greenplum Database 4.3 Installation Guide	A04
Greenplum Database 4.3 Administrator Guide	A03
Greenplum Database 4.3 Reference Guide	A04
Greenplum Database 4.3 Utility Guide	A04
Greenplum Database 4.3 Client Tools for UNIX	A03
Greenplum Database 4.3 Client Tools for Windows	A03
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	A03
Greenplum Database 4.3 Load Tools for Windows	A03
Greenplum Command Center 1.2.2 Administrator Guide	A01