Greenplum Database 4.3.4.1 Release Notes

Rev: A03

Updated: March, 2015

Welcome to Pivotal Greenplum Database 4.3.4.1

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

> **Note:** This document contains pertinent release information about Greenplum Database 4.3.4.1. For previous versions of the release notes for Greenplum Database, go to *Pivotal Documentation* or EMC *Support Zone*.

About Greenplum Database 4.3.4.1

Greenplum Database 4.3.4.1 is a patch release that introduces performance and stability enhancements. Please refer to the following sections for more information about this release.

- *Product Enhancements*
- *Changed Server Configuration Parameters*
- *Supported Platforms*
- *Resolved Issues in Greenplum Database 4.3.4.x*
- *Known Issues in Greenplum Database 4.3.4.x*
- *Upgrading to Greenplum Database 4.3.4.1*
- *Greenplum Database Tools Compatibility*
- *Greenplum Database Extensions Compatibility*
- *Hadoop Distribution Compatibility*
- *Greenplum Database 4.3.4.1 Documentation*

> **Warning:** The Pivotal Query Optimizer feature is a beta-level feature that is not supported and that is not permitted to be used in general availability releases. Running Greenplum Database with the Pivotal Query Optimizer enabled is not supported. For information about the Pivotal Query Optimizer in Greenplum Database, contact *gpdb@pivotal.io*.

Product Enhancements

Greenplum Database 4.3.4.1 includes enhancements in these areas:

- *Greenplum Database Server Configuration Parameter for Vacuum Operations*
- *Greenplum Database Utility gpssh Updates Target Host Environment*
- *Greenplum Command Center Performance*
- *External Table Support for Hadoop Distributions*

Greenplum Database Server Configuration Parameter for Vacuum Operations

The server configuration parameter `gp_appendonly_compaction_threshold` controls when compaction of the segment file occurs when the `VACUUM` command is run without the `FULL` option (a lazy vacuum).

For information about the parameter, see *gp_appendonly_compaction_threshhold*. For information about the `VACUUM` command, see the *Greenplum Database Reference Guide*.

Greenplum Database Utility gpssh Updates Target Host Environment

The `gpssh` utility supports the `-s` option. If specified, before executing any commands on the target host, `gpssh` sources the file `greenplum_path.sh` in the directory specified by the `$GPHOME` environment variable. This option is valid for both interactive mode and single command mode.

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

Greenplum Command Center Performance

When Greenplum Command Center monitors Greenplum Database 4.3.4.1, the default setting, `true` for the Greenplum Command Center configuration parameter `ignore_qexec_packet` reduces memory consumption spikes that were caused by the `gpmmon` process.

For information about the parameter, see *ignore_qexec_packet - Greenplum Command Center Parameter*. For information about the Greenplum Command Center, see the *Greenplum Command Center Administrator Guide*.

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 4.3.3.1, the `gphdfs` protocol has been enhanced to support Cloudera CDH 5.2 and 5.3.

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 Server Configuration Parameters

These server configuration parameters have been introduced in Greenplum Database 4.3.4.1.

- *gp_appendonly_compaction_threshhold*
- *gp_hadoop_target_version*
- *ignore_qexec_packet - Greenplum Command Center Parameter*

Greenplum Database server configuration parameter information is in "Server Configuration Parameters" of the *Greenplum Database Reference Guide*.

gp_appendonly_compaction_threshhold

Specifies the threshold ratio (as a percentage) of hidden rows to total rows that triggers compaction of the segment file when VACUUM is run without the FULL option (a lazy vacuum). If the ratio of hidden rows in a segment file on a segment is less than this threshold, the segment file is not compacted, and a log message is issued.

Value Range	Default	Set Classifications
integer(%)	10	master session reload

gp_hadoop_target_version

Specifies the installed version of Greenplum Hadoop target. For Greenplum Database 4.3.4.1, the parameter supports the value cdh5 for the Cloudera CDH 5.2 and 5.3 distributions of HDFS.

Value Range	Default	Set Classifications
gphd-1.0 gphd-1.1 gphd-1.2 gphd-2.0 gpmr-1.0 gpmr-1.2 hdp2 cdh5 cdh4.1 cdh3u2	gphd-1.1	local session reload

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

ignore_qexec_packet - Greenplum Command Center Parameter

When Greenplum Command Center monitors Greenplum Database 4.3.4.1, the parameter ignore_qexec_packet is set in the gpperfmon.conf configuration file. The default setting, true, reduces memory consumption spikes that can be caused by the gpmmon process.

ignore_qexec_packet

Specifies whether the gpsmon process ignores query execution data. The value is either true or false. The default value is true. If you specify ignore_qexec_packet = false, gpmmon memory usage could increase and an out of memory condition might occur due to a large number of UDP packets.

With the default setting, true, when a query is executed, the query plan iterators and their metrics are not added to the *iterators_** tables in the *gpperfmon* database. When you view an explain plan for a query, the query plan iterators and their metrics are displayed in the query

plan but are not stored in the *iterators_** tables. To enable storing query execution data in the *iterators_** tables, set the value of the parameter to `false`.

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 4.3.4.1 runs on the following platforms:

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

> **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 1: Data Domain Boost Compatibility

Greenplum Database	Data Domain Boost	DDOS
4.3.4.1	3.0.0.3	5.5.0.7
4.3.4.0	3.0.0.3	5.5.0.7
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.

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

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

Resolved Issues in Greenplum Database 4.3.4.x

The table below lists issues that are now resolved in Greenplum Database 4.3.4.x.

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

Table 2: Resolved Issues in 4.3.4.x

Issue Number	Category	Resolved In	Description
25202	Management Scripts: gpstart/ gpstop	4.3.4.1	The Greenplum Database utility `gpstop` did not clean up shared memory after stopping segments.
25201	Storage: Access Methods	4.3.4.1	A Greenplum Database PANIC occurred when altering a column and adding a new column in the same command on an append-optimized column-oriented table.
25193	Query Execution	4.3.4.1	In some cases, a PL/pgSQL exception handler disabled sequence access for a query. This caused Greenplum Database to return an error if the same query required a new sequence value after the PL/pgSQL exception.
25192	Storage: Transaction Management	4.3.4.1	For some queries that create sub-transactions, a fix in an earlier release degraded performance. The performance degradation issue has been resolved.
25162	Storage: Access Methods	4.3.4.1	In previous releases, the Greenplum Database server configuration parameter `gp_appendonly_compaction_threshold` was not visible. This issue has been resolved. See *Greenplum Database Server Configuration Parameter for Vacuum Operations*.
25082	Languages: R PLR	4.3.4.1	The Greenplum Database utility `gpssh` supports the `-s` option to source the file `greenplum_path.sh` before executing any commands on the target host. See *Greenplum Database Utility gpssh Updates Target Host Environment*
25066	Storage: Access Methods	4.3.4.1	In some cases, a Greenplum Database PANIC occurred when a query performed an index scan on an append-optimized table that was populated with concurrent inserts followed by updates or deletes.

Issue Number	Category	Resolved In	Description
25029	Upgrade / Downgrade	4.3.4.1	In some cases, the Greenplum Database utility `gpmigrator_mirror` failed when upgrading from 4.2.x to 4.3.x.
25017	Monitoring: gpperfmon server	4.3.4.1	In some cases, high memory consumption spikes were caused by `gpmon`. The memory consumption spikes might have caused `gpmmon` segmentation faults. The Greenplum Database Command Center parameter `ignore_qexec_packet` reduces the spikes of memory consumption. See *Greenplum Command Center Performance* .
25004	Upgrade / Downgrade	4.3.4.1	In Greenplum Database 4.3.3 and later, the `GRANT` and `REVOKE` commands support the `TRUNCATE` privilege on a table. The upgrade process to a release between 4.3.3 and 4.3.4 did not grant the `TRUNCATE` permission to database tables that were created before the upgrade. This issue has been resolved when upgrading to 4.3.4.1.
24964	Management Scripts: recoverseg	4.3.4.1	The Greenplum Database utility `gprecoverseg` did not ensure segments were stopped before attempting segment recovery.
24899	Functions and Languages	4.3.4.1	The aggregation function `sum` failed when the function arguments are empty arrays.
24898	Management Scripts: recoverseg	4.3.4.1	The Greenplum Database utility `gprecoverseg` failed when Greenplum Database persistent tables contained inconsistencies. Now, before attempting a segment recovery, the utility checks for persistent table inconsistencies and returns an error when inconsistencies are found.
24841	Storage: Access Methods	4.3.4.1	Running an `UPDATE` command after a `DROP COLUMN` and `ADD PARTITION` command on a partitioned table caused a Greenplum Database segment instance failure.
24740	Storage: Segment Mirroring	4.3.4.1	In some cases, interrupts that occurred during Greenplum Database query processing were not handled properly. This caused an increase in CPU utilization and a decrease in performance.
24678	Storage: Segment Mirroring	4.3.4.1	A performance issue was found with Greenplum Database during table scans. In some cases, unnecessary network traffic and increased disk activity occurred during read operations during the table scan.

Issue Number	Category	Resolved In	Description
24632	Catalog and Metadata	4.3.4.1	A `SELECT` command failed on a table if the table was defined with `DISTRIBUTED BY` clause that contained more than 200 columns. This issue has been resolved.
24563 20470	Query Planner	4.3.4.1	For some queries that required a correlated subquery, Greenplum Database generated a query plan that caused a segmentation fault when the plan was executed. This issue has been resolved.
24505	Management Scripts: recoverseg	4.3.4.1	The Greenplum Database utility `gprecoverseg` did not properly handle a segment that was in a transition phase, for example the primary segment is down and the mirror segment has not yet become the primary. Now the utility retries segment recovery in this situation.
23267	Functions and Languages	4.3.4.1	Memory leaks in the Greenplum Database aggregate functions `string_agg` and `array_agg` have been fixed.
21724	Query Planner	4.3.4.1	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 generated zero tuples and directly contributed to the output of the second stage plan, incorrect results might have been returned. This issue has been resolved.
17709	Catalog and Metadata	4.3.4.1	When the Greenplum Database utility `gpcheckcat` was run with the `-A` option to check all databases, the utility overwrote repair scripts for each database. This issue has been resolved.
24913	Storage Access Methods	4.3.4	In some cases, after a `VACUUM` operation was performed on an append-optimized table that contains an index, SQL queries that were run against the table returned incorrect data.
24778	Security	4.3.4	Greenplum Database software has been updated to use OpenSSL 0.9.8zc in response to the OpenSSL Security Advisory [15 Oct 2014]. For information about the advisory, see *http://www.openssl.org/news/secadv_20141015.txt*
24770	Dispatch	4.3.4	In some cases, `NULL` pointer handling was not performed correctly during query execution. This caused a panic on Greenplum Database master.

Issue Number	Category	Resolved In	Description
24694	Upgrade / Downgrade	4.3.4	The attribute name of a visimap metadata table for append-optimized tables was incorrectly named during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24693		4.3.4	For the visimap metadata table for append-optimized tables, the value of `attstorage` was set incorrectly in the *pg_attribute* table during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24692	Upgrade / Downgrade	4.3.4	A unnecessary record for the internal built-in function `pg_get_stat_walsenders` was added to the *pg_depend* table during the upgrade to 4.3.x from Greenplum Database 4.2.x. This issue has been resolved.
24684	Locking, Signals, Processes	4.3.4	In some cases, Greenplum Database did not clean up processes correctly during a commit transaction. This caused crash recovery issues after a Greenplum Database restart of both a primary segment and its mirror segment.
24683	Backup and Restore	4.3.4	There were performance issues when backing up specific tables from Greenplum Database using the Greenplum Database utility `gpcrondump` with the `--table-file`, `-t`, or `--exclude-table-file`, `-T` options. The back up performance has been improved.
24679	Locking, Signals, Processes	4.3.4	In some cases, Greenplum Database PL/Java did not exit properly and caused crash recovery issues after a Greenplum Database restart of both a primary segment and its mirror segment.
24677	Backup and Restore	4.3.4	In some cases after a successful back up operation, error messages about lock files were incorrectly displayed. This issue has been resolved.
24667	Build, Test	4.3.4	When creating a temporary table with the `ON COMMIT DELETE ROWS` clause in a heavy workload environment, the prepared transaction that created the temporary table failed in some cases.
24662	Upgrade/ Downgrade	4.3.4	The privilege of the internal system view *pg_stat_walsenders* was set to `NULL` during the upgrade to 4.3.x from Greenplum Database 4.2.x. The upgrade process now sets the privilege correctly.

Issue Number	Category	Resolved In	Description
24630	Management Scripts: gpsd	4.3.4	The Greenplum Database support utility `gpsd` has been enhanced to be more memory efficient in order to prevent failures when used on a database that contains large schemas.
24606	Storage access method	4.3.4	In some cases, queries that were run against the *gp_ toolkit* view *gp_size_of_table_and_indexes_disk* returned an error when the queries are run concurrently with DDL statements.
24515	Replication: Segment Mirroring	4.3.4	In some cases under a heavy workload, logging onto the Greenplum Database segment host as a UNIX user was not possible. This was caused by a Greenplum Database `filerep` process that was incorrectly sending signals to the user after the process failed to create a sub-process.
24383	GPHDFS	4.3.4	Greenplum Database external tables did not support using the gphdfs protocol and MapR to access HDFS data.
24264	Catalog and Metadata	4.3.4	The commands `REINDEX TABLE` *table_name* and `REINDEX INDEX` *index_name* did not re-index child partition indexes of a partitioned table.
24216	Management Scripts: gpstart/ gpstop	4.3.4	In some cases, the Greenplum Database utility `gpstop` failed to shutdown a database.
24052	Query Planner	4.3.4	In some cases, when executing a query that joined tables and the tables have extremely inaccurate statistics, Greenplum Database would return an error that reported being out of memory. This issue has been resolved.
24003	Dispatch, Interconnect	4.3.4	In some cases, executing an SQL query caused a memory leak.
23802	Query Execution	4.3.4	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.
23751	Monitoring: gpperfmon server	4.3.4	In some cases, a memory leak caused the `gpmmon` process to consume a large amount of memory and CPU resources.

Issue Number	Category	Resolved In	Description
23395	Transaction Management	4.3.4	If a Greenplum Database segment failed during two phase transaction processing, the transaction remained in a uncompleted state and was cleaned up only during a Greenplum Database restart. In many cases, this caused high disk consumption by the Greenplum Database `xlog` process.
23130	Resource Management	4.3.4	Some queries were terminated when they were run with a specific `statement_mem` value due to a rounding error.
23031	Backup and Restore	4.3.4	When restoring a specific table that was not in the default schema (public) with the `gpdbrestore` utility from a back up, the table was restored in the public schema.
22526	OS Abstraction	4.3.4	In some cases, Greenplum Database would wait indefinitely for a response from a client application. This issue has been resolved. See the server configuration parameter `gp_connection_send_timeout`.
19612	Replication: Segment Mirroring	4.3.4	In some cases, Greenplum Database did not properly handle the transition when segment mirror becomes unavailable. As a result, Greenplum Database becomes unavailable.
18509	Functions and Languages	4.3.4	In some cases, Greenplum Database did not handle data of type `date` properly and caused a segmentation fault.

Known Issues in Greenplum Database 4.3.4.x

This section lists the known issues in Greenplum Database 4.3.4.x. A workaround is provided where applicable.

For known issues discovered in previous 4.3.x releases, see the release notes at *Pivotal Network*. For known issues discovered in other previous releases, including patch releases to Greenplum Database 4.2.x, 4.1 or 4.0.x, see the corresponding release notes, available from EMC *Support Zone*:

Table 3: All Known Issues in 4.3.4.x

Issue	Category	Description
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.

Issue	Category	Description

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
		``

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`.
24944	DDL and Utility Statements	The `set_config()` function changes the sever configuration parameter only on the Greenplum Database master. The parameter value is not changed on the Greenplum Database segment instances. **Workaround:** Set the value of the server configuration parameter with the `SET` command.
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.

Issue	Category	Description
23824	Authentication	In some cases, LDAP client utility tools cannot be used after running the source command: `source $GPHOME/greenplum_path.sh` because the LDAP libraries included with Greenplum Database are not compatible with the LDAP client utility tools that are installed with operating system. **Workaround:** The LDAP tools can be used without running the source command in the environment.
23525	Query Planner	Some SQL queries that contain sub-selects fail with this error. `ERROR: Failed to locate datatype for paramid 0`
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the `gp_adjust_priority()` function. The attempt to execute this request might silently fail. The return value of the `gp_adjust_priority()` call indicates success or failure. If 1 is returned, the request was not successfully executed. If a number greater than 1 is returned, the request was successful. If the request fails, the priority of all running queries are unchanged, they remain as they were before the `gp_adjust_priority()` call.
23492	Backup and Restore,	A backup from a Greenplum Database 4.3.x system that is created with a Greenplum Database back up utility, for example gpcrondump, cannot be restored to a Greenplum Database 4.2.x system with the `psql` utility or the corresponding restore utility, for example gpdbrestore.
23521	Client Access Methods and Tools	Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database. **Workaround:** For Hadoop distributions based on Hadoop 2.2 or later that are supported by Greenplum Database, the classpath environment variable and other directory paths defined in `$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.

Issue	Category	Description
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.
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.

Issue	Category	Description
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.
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.

Issue	Category	Description
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`.
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.

Issue	Category	Description
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.
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 4.3.4.1

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.4.1. The minimum recommended upgrade path for this release is from Greenplum Database version 4.2.x.x. If you have an earlier major version of the database, you must first upgrade to version 4.2.x.x.

Prerequisites

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

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

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

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

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

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

- *Upgrading from 4.3.x to 4.3.4.1*
- *Upgrading from 4.3.x to 4.3.4.1 on Pivotal DCA Systems*
- *Upgrading from 4.2.x.x to 4.3.4.1*
- *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 4.3.4.1

An upgrade from 4.3.x to 4.3.4.1 involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

> **Important:** If you are upgrading from Greenplum Database 4.3.x on a Pivotal DCA system, see *Upgrading from 4.3.x to 4.3.4.1 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.4.1 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 *9*). 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 4.3.4.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-4.3.4.1
   ```

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

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

   to:

   ```
   source /usr/local/greenplum-db-4.3.4.1/greenplum_path.sh
   ```

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

   ```
   $ rm /usr/local/greenplum-db
   $ ln -s /usr/local/greenplum-db-4.3.4.1 /usr/local/greenplum-db
   ```

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

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

7. Run the `gpseginstall` utility to install the 4.3.4.1 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 4.3.4.1 as follows:

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

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

fix_ao_upgrade.py Utility

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

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

Syntax

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

Options

-r | --report

Report inconsistencies without making any changes.

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

Greenplum Database master hostname or IP address.

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

Greenplum Database master port.

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

User name to connect to Greenplum Database. The user must be a Greenplum Database superuser. Default is `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 4.3.4.1 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.4.1 on a Pivotal DCA system involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

> **Important:** Skip this section if you are *not* installing Greenplum Database 4.3.4.1 on DCA systems. This section is only for installing Greenplum Database 4.3 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 *6*). 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 4.3.4.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 4.3.4.1.

   ```
   # ./greenplum-db-appliance-4.3.4.1-build-1-RHEL5-x86_64.bin hostfile
   ```

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

   > **Note:** Greenplum Database extension packages are automatically migrated to the new release.

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

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

7. 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.

8. 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 4.3.4.1

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.4.1. 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 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-4.3.4.1-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.4.1-PLATFORM.bin
```

4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press ENTER to accept the default install path (for example: `/usr/local/greenplum-db-4.3.4.1`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer installs the Greenplum Database software and creates a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Source the path file from your new 4.3.4.1 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.4.1/greenplum_path.sh
```

8. Run the `gpseginstall` utility to install the 4.3.4.1 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 4.3.4.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 4.3.4.1.

```
# ./greenplum-db-appliance-4.3.4.1-build-1-RHEL5-x86_64.bin hostfile
```

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

Upgrade Execution

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

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

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

   On a DCA system, the path to the 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 4.3.4.1 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.4.1/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.4.1/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.4.1 installation.

   a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.4.1 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.4.1 /usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

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

```
# ln -s /usr/local/GP-4.3.4.1 /usr/local/greenplum-db
```

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

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.3.4.1 /usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```

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

```
=> ln -s /usr/local/GP-4.3.4.1 /usr/local/greenplum-db
```

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

```
$ gpmigrator_mirror --check-only
   /usr/local/greenplum-db-4.2.6.3
   /usr/local/greenplum-db-4.3.4.1
```

On a DCA system, the old GPHOME location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new GPHOME location would be similar to `/usr/local/GP-4.3.4.1`.

8. As `gpadmin`, run the 4.3.4.1 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-4.3.4.1
```

On a DCA system, the old `GPHOME` location might be similar to `/usr/local/GP-4.2.8.1` (depending on the old installed version) and the new `GPHOME` location would be similar to `/usr/local/GP-4.3.4.1`.

> **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 4.3.4.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 4.3.4.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 4.3.4 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 custom modules against your 4.3.4.1 installation (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 4.3 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 4.3.4.1 installation (if necessary).

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.2.x.x to 4.3.4.x. Detailed upgrade instructions are provided in the *Known Issues in Greenplum Database 4.3.4.x* 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 4.3.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 4.3.4 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 4.3.4.1.

1. Upgrade from your current release to 4.2.x.x (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at *Pivotal Documentation*).
2. Follow the upgrade instructions in these release notes for *Upgrading from 4.2.x.x to 4.3.4.1*.

For Users Running Greenplum Database 4.0.x.x

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

1. Upgrade from your current release to 4.1.x.x (follow the upgrade instructions in the latest Greenplum Database 4.1.x.x release notes available on *Support Zone*).
2. Upgrade from the current release to 4.2.x.x (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at *Pivotal Documentation*).
3. Follow the upgrade instructions in these release notes for *Upgrading from 4.2.x.x to 4.3.4.1*.

For Users Running Greenplum Database 3.3.x.x

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

1. Upgrade from your current release to the latest 4.0.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.0.x.x release notes available on *Support Zone*).
2. Upgrade the 4.0.x.x release to the latest 4.1.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.1.x.x release notes available on *Support Zone*).
3. Upgrade from the 4.1.1 release to the latest 4.2.x.x release (follow the upgrade instructions in the latest Greenplum Database 4.2.x.x release notes available at *Pivotal Documentation*).
4. Follow the upgrade instructions in these release notes for *Upgrading from 4.2.x.x to 4.3.4.1*.

Troubleshooting a Failed Upgrade

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

Be prepared to provide the following information:

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

Greenplum Database Tools Compatibility

Client Tools

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

Tool packages are available from *Pivotal Network*.

Table 4: Greenplum Database Tools Compatibility

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

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

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

- Windows Server 2008 R2 (64-bit)
- Windows XP (32-bit and 64-bit)

Greenplum Database Extensions Compatibility

Greenplum Database delivers an agile, extensible platform for in-database analytics, leveraging the system's massively parallel architecture. Greenplum Database enables turn-key in-database analytics with Greenplum extensions.

You can download Greenplum extensions packages from *Pivotal Network* and install them using the Greenplum Packager Manager (`gppkg`). See the *Greenplum Database Utility Guide* for details.

Note that Greenplum Package Manager installation files for extension packages may release outside of standard Database release cycles.

The following table provides information about the compatibility of the Greenplum Database Extensions and their components with this Greenplum Database release.

> **Note:** The PL/Python database extension is already included with the standard Greenplum Database distribution.

Table 5: Greenplum Database Extensions Compatibility

Greenplum Database Extension	Extension Components	
	Name	**Version**
PostGIS 2.0.1 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PostGIS 1.0 for Greenplum Database	PostGIS	1.4.2
	Proj	4.7.0
	Geos	3.2.2
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

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

Table 6: Greenplum Database 4.3 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1
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, pv2.0.1 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 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)
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 4.3.4.1 Documentation

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

Table 8: Greenplum Database Documentation

Title	Revision
Greenplum Database 4.3.4.1 Release Notes	A03
Greenplum Database 4.3 Installation Guide	A06
Greenplum Database 4.3 Administrator Guide	A06
Greenplum Database 4.3 Reference Guide	A07
Greenplum Database 4.3 Utility Guide	A07
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	A05
Greenplum Database 4.3 Load Tools for Windows	A04
Greenplum Command Center 1.3 Administrator Guide	A01