Greenplum Database 4.3.11.1 Release Notes

Rev: A01

Updated: January, 2017

Welcome to Pivotal Greenplum Database 4.3.11.1

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

Warning: Greenplum Database 4.3.11.0 was removed from General Availability due to an upgrade issue that is resolved in 4.3.11.1. Do not upgrade to Greenplum Database 4.3.11.0 and do not attempt to change an existing system with data from 4.3.11.0 to another version.

Note: For Greenplum Database that is installed on Red Hat Enterprise Linux 7.x or CentOS 7.x prior to 7.3, an operating system issue might cause Greenplum Database that is running large workloads to hang in the workload.. The Greenplum Database issue is caused by Linux kernel bugs.

RHEL 7.3 and CentOS 7.3 resolves the issue.

Note: This document contains pertinent release information about Greenplum Database 4.3.11.1. For previous versions of the release notes for Greenplum Database, go to *Pivotal Documentation* or EMC *Support Zone*. For information about Greenplum Database end of life, see *Greenplum Database end of life policy*.

Important: Pivotal Global Support Services (GSS) does **not** provide support for open source versions of Greenplum Database. Only Pivotal Greenplum Database is supported by Pivotal GSS.

About Greenplum Database 4.3.11.1

Greenplum Database 4.3.11.1 is a patch release that includes enhancements and changes and resolves some known issues. Please refer to the following sections for more information about this release.

Product Enhancements

Greenplum Database includes these enhancements.

Improved PQO Query Execution

Greenplum Database introduces two Greenplum Database server configuration parameters that can improve the performance of queries that are executed by Pivotal Query Optimizer.

- The `optimizer_parallel_union` server configuration parameter controls the amount of parallelization that occurs for queries that contain a `UNION` or `UNION ALL` clause. When the value is `on`, PQO can generate a query plan where the child operations of a `UNION` or `UNION ALL` node operation execute in parallel on segment instances.
- The `optimizer_sort_factor` server configuration parameter controls the cost factor to apply to sorting operations during query optimization. The cost factor can be adjusted for queries when data skew is present.

For information about the parameters, see *New Parameters*.

Improved Greenplum Database Memory Management

Greenplum Database introduces index based memory accounting that optimizes overall query performance and memory overhead. For example, index based memory accounting is more efficient for long running transactions. This makes memory management more efficient and more scalable for persistent sessions from third party applications. The new memory accounting is also faster for sessions that run a lot of queries, with or without transactions.

For information about Greenplum Database memory management, see the *Greenplum Database Administrator Guide*.

Enhanced Query Cancellation

Greenplum Database includes the new server configuration parameter *`vmem_process_interrupt`*. The parameter enables Greenplum Database to respond more quickly to query cancellation requests. Moreover, executing queries that have a pending cancellation request are prevented from allocating more memory from the system. This feature enhances query lifespan and memory management.

When the parameter is enabled, Greenplum Database checks the current session for pending query cancellations or other pending interrupts and processes the interrupts before reserving additional vmem memory for a query during query execution. This ensures more responsive interrupt processing, including query cancellation requests.

For information about the parameter, see *New Parameters*.

Improved Query Execution of Hash Aggregates

Greenplum Database includes optimizations that improve hash aggregate memory cleanup. These optimizations improve query execution for queries that generate large hash tables.

gptransfer Transfers Data from Partitioned to Non-partition Tables

With Greenplum Database , you can specify the `gptransfer` utility option `--partition-transfer-non-partition-target` to copy data from leaf child partition tables of partitioned tables in a source database to non-partitioned table in a destination database. In previous releases, the `gptransfer` utility only supported copying data from leaf child partition tables of partitioned tables to only partitioned tables with the `--partition-transfer` option.

Specify the `--partition-transfer-non-partition-target` option with the `-f` option to copy data from partitioned tables non-partitioned tables. The text file specified by the `-f` option contains a list of fully qualified leaf child partition table names in the source database and non partitioned tables names in the destination database. Each line lists the fully qualified source and destination table names. Both source and destination table names are required in the file, and the destination tables must exist.

For the partitioned table in the source database and the table in the destination database, the number of table columns and the order of the column data types (the source and destination table column names can be different). The same destination table can be specified in the file for multiple source leaf child partition tables from a single partitioned table. Transferring data from source leaf child partition tables that are from different partitioned tables to a single non-partitioned table is not supported.

This option is not valid with these options: `-d`, `--dest-database`, `--drop`, `-F`, `--full`, `--schema-only`, `-T`, `-t`, `--truncate`, `--validate`.

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

Enhanced PL/Java Environment for Development

For Greenplum Database , the new server configuration parameter *`pljava_classpath_insecure`* controls the ability of normal database user to set the server configuration parameter `pljava_classpath`. Greenplum Database uses the list of jar files or directories containing jar files specified by `pljava_classpath` when running PL/Java functions. When *`pljava_classpath_insecure`* is enabled, Greenplum Database developers who are working on PL/Java functions do not have to be database superusers to change `pljava_classpath`. In previous releases, only database superusers could change `pljava_classpath`.

> **Warning:** Enabling *`pljava_classpath_insecure`* exposes a security risk by giving non-administrator database users the ability to run unauthorized Java methods.

For information about the *`pljava_classpath_insecure`* parameter, see *New Parameters*. For information about the PL/Java procedural language, see the *Greenplum Database Reference Guide*.

Changed Features

Greenplum Database includes these feature changes.

- The external table `gphdfs` protocol supports MapR 5.x and Cloudera 5.7.x and 5.8.x. See *Hadoop Distribution Compatibility*.
- The maximum length of the external table custom formatter specification has been increased. The specification is now stored as a text data type (approximately 50K bytes). In previous releases, the maximum length was 8K characters.

  For an external table definition that specifies the `CUSTOM` format type (`FORMAT 'CUSTOM' (Formatter=`*`formatter_specifications`*), the length of the formatter specification, the string including `Formatter=`, can be up to approximately 50K bytes.

For information about the external table `gphdfs` protocol, see the *Greenplum Database Administrator Guide*. For information about the `CREATE EXTERNAL TABLE` command, see the *Greenplum Database Reference Guide*

New and Changed Parameters

New Parameters

Greenplum Database includes new server configuration parameters.

- *optimizer_parallel_union*
- *optimizer_sort_factor*
- *pljava_classpath_insecure*
- *vmem_process_interrupt*

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

optimizer_parallel_union

When Pivotal Query Optimizer (PQO) is enabled, `optimizer_parallel_union` controls the amount of parallelization that occurs for queries that contain a `UNION` or `UNION ALL` clause.

When the value is `off`, the default, PQO generates a query plan where each child of an APPEND(UNION) operator is in the same slice as the APPEND operator. During query execution, the children are executed in a sequential manner.

When the value is `on`, PQO generates a query plan where a redistribution motion node is under an APPEND(UNION) operator. During query execution, the children and the parent APPEND operator are on different slices, allowing the children of the APPEND(UNION) operator to execute in parallel on segment instances.

The parameter can be set for a database system, an individual database, or a session or query.

Value Range	Default	Set Classifications
boolean	off	master session reload

optimizer_sort_factor

When Pivotal Query Optimizer (PQO) is enabled, `optimizer_sort_factor` controls the cost factor to apply to sorting operations during query optimization. The default value `1` specifies the default sort cost factor. The value is a ratio of increase or decrease from the default factor. For example, a value of `2.0` sets the cost factor at twice the default, and a value of `0.5` sets the factor at half the default.

The parameter can be set for a database system, an individual database, or a session or query.

Value Range	Default	Set Classifications
Decimal > 0	1	master session reload

pljava_classpath_insecure

Controls whether the server configuration parameter *pljava_classpath* can be set by a user without Greenplum Database superuser privileges. When true, pljava_classpath can be set by a regular user. Otherwise, *pljava_classpath* can be set only by a superuser.

The default is false.

> **Warning:** Enabling this parameter exposes a security risk by giving non-administrator database users the ability to run unauthorized Java methods.

Value Range	Default	Set Classifications
Boolean	false	master session reload superuser

vmem_process_interrupt

Enables checking for interrupts before reserving vmem memory for a query during Greenplum Database query execution. Before reserving additional vmem memory for a running query, Greenplum Database checks the current session running the query for a pending query cancellation or other pending interrupts. This ensures more responsive interrupt processing, including query cancellation requests. The default is off.

The parameter can be set for a database system, an individual database, or a session or query.

Value Range	Default	Set Classifications
Boolean	off	master session reload

Changed Parameters

The description of the pljava_classpath server configuration parameter includes information about the *pljava_classpath_insecure* server configuration parameter.

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

pljava_classpath

A colon (:) separated list of jar files or directories containing jar files needed for PL/Java functions. The full path to the jar file or directory must be specified, except the path can be omitted for jar files in the $GPHOME/lib/postgresql/java directory. The jar files must be installed in the same locations on all Greenplum hosts and readable by the gpadmin user.

The pljava_classpath parameter is used to assemble the PL/Java classpath at the beginning of each user session. Jar files added after a session has started are not available to that session.

If the full path to a jar file is specified in pljava_classpath it is added to the PL/Java classpath. When a directory is specified, any jar files the directory contains are added to the PL/Java classpath. The search does not descend into subdirectories of the specified directories. If the name of a jar file is included

in `pljava_classpath` with no path, the jar file must be in the `$GPHOME/lib/postgresql/java` directory.

> **Note:** Performance can be affected if there are many directories to search or a large number of jar files.

If *`pljava_classpath_insecure`* is `false`, setting the `pljava_classpath` parameter requires superuser privilege. Setting the classpath in SQL code will fail when the code is executed by a user without superuser privilege. The `pljava_classpath` parameter must have been set previously by a superuser or in the `postgresql.conf` file. Changing the classpath in the `postgresql.conf` file requires a reload (`gpstop -u`).

Value Range	Default	Set Classifications
string		master session reload superuser

Downloading Greenplum Database

These are the locations of the Greenplum Database software and documentation:

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

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

Supported Platforms

Greenplum Database runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 7.x (See *Supported Platform Notes*. See the Warning in *Welcome*)
- Red Hat Enterprise Linux 64-bit 6.x
- Red Hat Enterprise Linux 64-bit 5.x
- SuSE Linux Enterprise Server 64-bit 11 SP1, 11 SP2, 11 SP4
- Oracle Unbreakable Linux 64-bit 5.5
- CentOS 64-bit 7.x
- CentOS 64-bit 6.x
- CentOS 64-bit 5.x

> **Important:** Support for SuSE Linux Enterprise Server 64-bit 10 SP4 has been dropped for Greenplum Database 4.3.9.0 and later releases.

Greenplum Database 4.3.x supports these Java versions:

- 8.*xxx*
- 7.*xxx*
- 6.*xxx*

Greenplum Database 4.3.x supports Data Domain Boost on Red Hat Enterprise Linux.

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

Table 1: Data Domain Boost Compatibility

Greenplum Database	Data Domain Boost	DDOS
4.3.11.1	3.0.0.3	5.7 (all versions) 5.6 (all versions) 5.5 (all versions) 5.4 (all versions) 5.3 (all versions)
4.3.10.0	3.0.0.3	5.7 (all versions) 5.6 (all versions) 5.5 (all versions) 5.4 (all versions) 5.3 (all versions)
4.3.9.1 4.3.9.0	3.0.0.3	5.7 (all versions) 5.6 (all versions) 5.5 (all versions) 5.4 (all versions) 5.3 (all versions)
4.3.8.1 4.3.8.0	3.0.0.3	5.6 (all versions) 5.5 (all versions) 5.4 (all versions) 5.3 (all versions)
4.3.7.3 4.3.7.2 4.3.7.1 4.3.7.0	3.0.0.3	5.6 (all versions) 5.5 (all versions) 5.4 (all versions) 5.3 (all versions)
4.3.6.2 4.3.6.1 4.3.6.0	3.0.0.3	5.6 (all versions) 5.5.0.*x* 5.4 (all versions) 5.3 (all versions)
4.3.5.3 4.3.5.2 4.3.5.1 4.3.5.0	3.0.0.3	5.5.0.*x* 5.4 (all versions) 5.3 (all versions)

Greenplum Database	Data Domain Boost	DDOS
4.3.4.2 4.3.4.1 4.3.4.0	3.0.0.3	5.5.0.*x* 5.4 (all versions) 5.3 (all versions)
4.3.3.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.2.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.1.0	2.6.2.0	5.2, 5.3, and 5.4
4.3.0.0	2.4.2.2	5.0.1.0, 5.1, and 5.2

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

Greenplum Database support on DCA:

- Greenplum Database 4.3.x, all versions, is supported on DCA V3.
- 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: Greenplum Database does not support the ODBC driver for Cognos Analytics V11.

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 you migrate to a version of IBM Cognos software that supports connectivity to Greenplum Database with a JDBC driver.

Supported Platform Notes

The following notes describe platform support for Greenplum Database. Please send any questions or comments to Pivotal Support at *https://support.pivotal.io*.

- The only file system supported for running Greenplum Database is the XFS file system. All other file systems are explicitly *not* supported by Pivotal.
- Greenplum Database is supported on all 1U and 2U commodity servers with local storage. Special purpose hardware that is not commodity *may* be supported at the full discretion of Pivotal Product Management based on the general similarity of the hardware to commodity servers.
- Greenplum Database is supported on network or shared storage if the shared storage is presented as a block device to the servers running Greenplum Database and the XFS file system is mounted on the block device. Network file systems are *not* supported. When using network or shared storage, Greenplum Database mirroring must be used in the same way as with local storage, and no modifications may be made to the mirroring scheme or the recovery scheme of the segments. Other features of the shared storage such as de-duplication and/or replication are not directly supported by Pivotal Greenplum Database, but may be used with support of the storage vendor as long as they do not interfere with the expected operation of Greenplum Database at the discretion of Pivotal.
- Greenplum Database is supported when running on virtualized systems, as long as the storage is presented as block devices and the XFS file system is mounted for the storage of the segment directories.
- A minimum of 10-gigabit network is required for a system configuration to be supported by Pivotal.

- Greenplum Database is supported on Amazon Web Services (AWS) servers using either Amazon instance store (Amazon uses the volume names `ephemeral[0-20]`) or Amazon Elastic Block Store (Amazon EBS) storage. If using Amazon EBS storage the storage should be RAID of Amazon EBS volumes and mounted with the XFS file system for it to be a supported configuration.
- For Red Hat Enterprise Linux 7.2 or CentOS 7.2, the default `systemd` setting `RemoveIPC=yes` removes IPC connections when non-system users logout. This causes the Greenplum Database utility `gpinitsystem` to fail with semaphore errors. To avoid this issue, see "Setting the Greenplum Recommended OS Parameters" in the *Greenplum Database Installation Guide*.

Resolved Issues in Greenplum Database 4.3.11.1

The table below lists issues that are now resolved in Pivotal Greenplum Database 4.3.11.1

For issues resolved in prior 4.3 releases, refer to the corresponding release notes. Release notes are available from *Pivotal Network* or on the Pivotal Greenplum Database documentation site at *Release Notes*. A consolidated list of resolved issues for all 4.3 releases is also available on the documentation site.

Table 2: Resolved Issues in 4.3.11.1

Issue Number	Category	Resolved In	Description
26735	Upgrade/ Downgrade	4.3.11.1	Upgrading from Greenplum Database 4.3.x.x to Greenplum Database 4.3.11.0 failed due to a persistent table issue. The issue occurred if a database was dropped from the Greenplum Database system prior to the upgrade. The issue has been resolved in Greenplum Database 4.3. 11.1. See the Warning in the *Welcome* section.
26688	Query Execution	4.3.11.1	The following types of queries failed and returned a message stating that no such file or directory exists: queries with a cursor statement or some queries from a JDBC or ODBC driver. The failures occurred when there were utility mode connections to Greenplum Database segment instances. The failures occurred because the Greenplum Database did not properly track the utility mode connections. Now, Greenplum Database correctly tracks utility mode connections to segment instances.
26669	Query Optimizer	4.3.11.1	Pivotal Query Optimizer (PQO) did not correctly rewrite some queries that contain an outer reference in a `GROUP BY` clause and did not use the outer reference in another aggregate function. The incorrect rewire caused a Greenplum Database PANIC. Now, PQO performs additional validations to ensure the specified type of query is correctly rewritten.

Issue Number	Category	Resolved In	Description
26665	Transaction Management	4.3.11.1	In some cases, Greenplum Database did not properly handle SQL transactions that consist of multiple SQL statements with a `COMMIT` or `END` statement that occurs before a statement that requires a global subtransaction. This issue might force the database into recovery mode, or might cause a Greenplum Database PANIC. The processing of global subtransactions for the specified type of SQL transactions has been improved.
26659	Scripts: Backup and Restore	4.3.11.1	In some cases, the Greenplum Database `gpcrondump` utility failed when performing an incremental backup operation to a Data Domain system with the `--ddboost` option. The backup failed when the `report` and `cdatabase` files were generated on different days (for example, if the backup is started just before midnight). Now, the `gpcrondump` utility uses the backup timestamp instead of the current timestamp when generating files to avoid this issue.
26654	Dispatch	4.3.11.1	In some cases during Greenplum Database query execution, communication between the query dispatcher (QD) and a query executor (QE) was not handled properly. This caused the QE to not recognize a cancel request and caused the query execution to hang. Message validation between QD and QE has been improved.
26631	Query Optimizer	4.3.11.1	Pivotal Query Optimizer (PQO) generated an incorrect cardinality estimate for queries that contain an `OR` predicate with these properties. • Both conditions in the `OR` expression are on the same column. • One condition of the `OR` expression uses <> (not equal) or `NOT IN`, the other condition is an `IS NULL` condition. Incorrect estimates can prevent PQO from generating an optimal plan. PQO cardinality estimation has been enhanced.
26613	Query Optimizer	4.3.11.1	For some Pivotal Query Optimizer (PQO) errors, PQO incorrectly returned a WARNING message followed by an ERROR message. Now, PQO returns only the ERROR message.

Issue Number	Category	Resolved In	Description
26612 26600 26576	Catalog and Metadata	4.3.11.1	In some cases, a PANIC on a Greenplum Database segment was caused by inconsistencies in a system catalog table. The inconsistencies were caused by incorrect data being inserted into the table. Additional checks have been added to Greenplum Database that prevent incorrect data from being inserted into system catalog tables.
26605	Locking, Signals, Processes	4.3.11.1	In some cases while processing append-optimized table tables, Greenplum Database returned an `invalid page header` message that did not specify the correct relation ID. Now, the message includes the correct relation ID.
26603	Query Planner	4.3.11.1	In Greenplum Database, for queries that contain a correlated `EXIST` subquery and a `WITH` clause, the legacy planner generated a correlated plan that incorrectly attempted to access parameters across slices. This caused the query executor (QE) to panic. Now, the legacy query planner generates a plan that resolves the issue.
26587	Query Execution, Query Optimizer	4.3.11.1	Pivotal Query Optimizer (PQO) caused a Greenplum Database PANIC for queries that contain a scalar subquery with a single constant table child and the scalar subquery returns an outer reference. For those queries, PQO incorrectly processed the subquery. This issue has been resolved.
26524	Scripts: Backup and Restore	4.3.11.1	In some cases, when restoring database data, the Greenplum Database `gpdbrestore` utility incorrectly reported errors when restoring indexes, constraints, and other metadata when restoring the data. The restore operation completed successfully. Now, the incorrect errors are no longer reported.
26504	Scripts: gptransfer	4.3.11.1	In some cases, a transfer operation failed when the `gptransfer` utility did not correctly compare source and destination partitioned tables. The utility returned a message stating that the column layout of the two tables was different. The failure occurred if a column was dropped and then added back to the source partitioned table and then the DDL for the table was used to create the destination table. Now, the `gptransfer` utility correctly compares the column layout of the partitioned tables.

Issue Number	Category	Resolved In	Description
26469	Storage: Vacuum/ Reindex/ Truncates	4.3.11.1	Executing `gpstop -i` after reindexing the `pg_class` catalog table in the `template1` database could leave the database in an inconsistent state. This issue has been resolved.
26187	Query Execution	4.3.11.1	In some cases, long running sessions were consuming memory on Greenplum Database and the sessions were idle on the master segment and did not have connections to segment instances. Memory management has been improved to better handle long running transactions. See *Improved Greenplum Database Memory Management*.
134801867	Transaction Management	4.3.11.1	In some cases, Greenplum Database did not handle the processing of two phase commits properly when the first phase of the two phase commit failed. This issue caused the `pg_xlogs` to grow to a large size. The processing of two phase commits has been enhanced to handle the failure properly.
133827909	Query Optimizer	4.3.11.1	Pivotal Query Optimizer (PQO) returned incorrect results for queries that contained a predicate on a partitioned table partition key column and a subquery that contained a `LIMIT` clause. PQO incorrectly pushed the predicate into a subquery. This issue has been resolved.
132718373	Query Optimizer	4.3.11.1	When Pivotal Query Optimizer (PQO) and the server configuration parameter `optimizer_parallel_union` are both enabled, some queries that contained a `UNION` operator would incorrectly fall back to the legacy query optimizer. Now, PQO executes the queries.
132714585	Management Scripts	4.3.11.1	In some cases, after performing an incremental segment recovery (running the Greenplum Database utility `gprecoverseg` without the `-F` option), did not recover the Greenplum Database segment correctly and caused a SIGSEGV. Now, `gprecoverseg` correctly performs incremental segment recovery.
131931989	S3 External Table	4.3.11.1	When an Greenplum Database readable external table is defined with the `s3` protocol and the table definition uses the `HEADER` option, Greenplum Database did not correctly process table header information when a Greenplum Database segment read data from multiple files. This issue has been resolved.

Issue Number	Category	Resolved In	Description
131931971	S3 External Table	4.3.11.1	When an Greenplum Database readable external table is defined with the `s3` protocol, Greenplum Database did not correctly read multiple files when the last line of a file did not contain a trailing return character. The last line of the first file (without a trailing return character) was joined to first line of the second file and was read as a single line. Now, files without a trailing return character are processed correctly.
131309733	Query Execution	4.3.11.1	The `EXPLAIN ANALYZE` command did not report the correct number of segments that spill workfiles for a query, if the query contains an `ORDER BY` clause and the query's sort operator spilled workfiles, and the server configuration parameter `gp_enable_mk_sort` is `off`. Now, `EXPLAIN ANALYZE` reports the correct number of segments that execute sort operations and spill workfiles.
130407893	Transaction Management	4.3.11.1	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. This issue has been resolved.
130121327	Query Optimizer	4.3.11.1	When Pivotal Query Optimizer (PQO) and the server configuration parameter `optimizer_parallel_union` are both enabled, PQO evaluated query plans that use only `parallel union all` and did not evaluate plans that use `serial union all`. Now, both types of plans are now evaluated.
129871531	Query Optimizer	4.3.11.1	When Pivotal Query Optimizer (PQO) is enabled, PQO did not infer additional predicates based on the join and select predicates for some queries that contain a `LIMIT` clause over subqueries and a join. This resulted in the generated query plan being sub-optimal. This issue has been resolved.

Known Issues in Greenplum Database 4.3.11.1

This section lists the known issues in Greenplum Database 4.3.11.1. 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 4.3.11.1

Issue	Category	Description
115746399	Operating System	For Greenplum Database that is installed on Red Hat Enterprise Linux 7.x or CentOS 7.x prior to 7.3, an operating system issue might cause Greenplum Database that is running large workloads to hang in the workload. The Greenplum Database issue is caused by Linux kernel bugs. **Workaround:** RHEL 7.3 and CentOS 7.3 resolves the issue.
26626	GPHDFS	For Greenplum Database external tables, the `gphdfs` protocol supports Avro files that contain a single top-level schema. Avro files that contain multiple top-level schemas are not supported.
25584	Query Execution	In some situations, a running Greenplum Database query cannot be terminated with the functions `pg_cancel_backend` or `pg_terminate_backend`. The functions could not terminate the query due to a blocking `fopen` of a FIFO file for write.
26249	GPHDFS	When reading data from an Avro file, the `gphdfs` protocol does not support the double quote character (") within string data. The `gphdfs` protocol uses the double quote as the column delimiter. **Workaround:** Before reading data from an Avro file, either remove double quotes that are in string data or replace the character with a different character.
26292	Loaders: gpload	The Greenplum Database `gpload` utility fails on MacOS X El Capitan. The utility script is included with the Greenplum Database Load Tools installer package for Apple OS X `greenplum-loaders-`*`version`*`-OSX-i386.bin`. **Workaround:** Run the python script `gpload.py` directly. For example, python command displays the `gpload` help information on the command line. `python gpload.py -h`

Issue	Category	Description
26128	Loaders: gpload	When the YAML control file for the Greenplum Database `gpload` utility specifies the key `LOG_ERRORS: true` without the key `REUSE TABLES: true`, the `gpload` operation returns only summary information about formatting errors. The formatting errors are deleted from Greenplum Database error logs. When `REUSE TABLES: true` is not specified, the temporary tables that are used by `gpload` are dropped after the `gpload` operation, and the formatting errors are also deleted from the Greenplum Database error logs. **Workaround:** Specify the YAML control file key `REUSE TABLES: true` to retain the temporary tables that are used to load the data. The log information is also retained. You can delete the formatting errors in the Greenplum Database logs with the Greenplum Database function `gp_truncate_error_log()`. For information about the `gpload` utility, see the *Greenplum Database Utility Guide*.

25934
25936
	Query Optimizer
	
	Query Planner
		For queries that compare data from columns of different character types, for example a join comparing a columns of data types `CHAR(n)` and `VARCHAR(m)`, the returned results might not be as expected depending the padding added to the data (space characters added after the last non-space character).
		
		For example, this comparison returns false.
		
		`select 'A '::char(2) ='A '::text ;`
		
		This comparison returns true.
		
		`select 'A'::char(2) ='A '::varchar(5) ;`
		
		Workaround: Pivotal recommends specifying character column types to be of data type `VARCHAR` or `TEXT` so that comparisons include padding added to the data.
		
		For information about how the character data types `CHAR`, `VARCHAR`, and `TEXT` handle padding added to the data see the `CREATE TABLE` command in the *Greenplum Database Reference Guide*.

25737	Catalog and Metadata	Greenplum Database does not support the `FILTER` clause within aggregate expressions.
25754	Management Scripts: expansion	The Greenplum Database `gpexpand` utility fails to create an input file for system expansion if the Greenplum Database system define different TCP/IP port numbers on different hosts for Greenplum Database internal communication. **Workaround:** Create the input file manually.
25833	Management Scripts: gpexpand	The Greenplum Database utility `gpexpand` fails when expanding a Greenplum Database system and in the system a database table column name contains a tab character. The utility does not support database names, table names, or column names that contain a tab character.

Issue	Category	Description
15835	DDL and Utility Statements	For multi-level partitioned tables that have these characteristics: • The top level partition is partitioned by range. • The lowest level partition (the leaf child partitions) are partitioned by list. Splitting a subpartition with the `ALTER TABLE SPLIT PARTITION` command returns an error and rolls back the transaction.
12019	Management Scripts: checkperf	When the Greenplum Database `gpcheckperf` utility is run with the option `-f` *`host_file`* and the host that is running `gpcheckperf` is listed in *host_file*, processes that were started `gpcheckperf` might not be cleaned up after the utility completes. **Workaround**: Manually stop the processes that were started by `gpcheckperf`.
24870	Query Optimizer	The Pivotal Query Optimizer might terminate all sessions if a query attempts to cast to a timestamp a date with year greater than 200,000.
23571	Query Optimizer	For queries that contain inequality conditions such as `!=` , `<` and , `>`, the Pivotal Query Optimizer does not consider table indexes when generating a query plan. For those queries, indexes are not used and the query might run slower than expected.
21508	Query Optimizer	The Pivotal Query Optimizer does not support GiST indexes.
20030	Query Optimizer	The Pivotal Query Optimizer does not support partition elimination when the query contains functions that are applied to the partition key.
20360	Query Execution	The Pivotal Query Optimizer does not enforce different access rights in different parts of a partition table. Pivotal recommends that you set the same access privileges for the partitioned table and all its parts (child tables).
20241	Query Optimizer	The Pivotal Query Optimizer does not consider indices when querying parts/child tables of partitioned tables directly.
25326	Interconnect	Setting the Greenplum Database server configuration parameter `log_hostname` to `on` Greenplum Database segment hosts causes an Interconnect Error that states that the listeneraddress name or service not known. The parameter should be set to `on` only on the Greenplum Database master.

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@example.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@example.com'"`

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

Issue	Category	Description
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the `gp_adjust_priority()` function. The attempt to execute this request might silently fail. The return value of the `gp_adjust_priority()` call indicates success or failure. If 1 is returned, the request was not successfully executed. If a number greater than 1 is returned, the request was successful. If the request fails, the priority of all running queries are unchanged, they remain as they were before the `gp_adjust_priority()` call.
23492	Backup and Restore,	A backup from a Greenplum Database 4.3.x system that is created with a Greenplum Database back up utility, for example gpcrondump, cannot be restored to a Greenplum Database 4.2.x system with the `psql` utility or the corresponding restore utility, for example gpdbrestore.
23521	Client Access Methods and Tools	Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database. **Workaround:** For Hadoop distributions based on Hadoop 2.2 or later that are supported by Greenplum Database, the classpath environment variable and other directory paths defined in `$GPHOME/lib/hadoop/hadoop_env.sh` must be to be modified so that the paths point to the appropriate JAR files.

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

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

Issue	Category	Description
18247	DDL and Utility Statements	`TRUNCATE` command does not remove rows from a sub-table of a partitioned table. If you specify a sub-table of a partitioned table with the `TRUNCATE` command, the command does not remove rows from the sub-table and its child tables. **Workaround:** Use the `ALTER TABLE` command with the `TRUNCATE PARTITION` clause to remove rows from the sub-table and its child tables.
19705	Loaders: gpload	`gpload` fails on Windows XP with Python 2.6. **Workaround:** Install Python 2.5 on the system where gpload is installed.
19493 19464 19426	Backup and Restore	The `gpcrondump` and `gpdbrestore` utilities do not handle errors returned by DD Boost or Data Domain correctly. These are two examples: • If invalid Data Domain credentials are specified when setting the Data Domain Boost credentials with the `gpcrondump` utility, the error message does not indicate that invalid credentials were specified. • Restoring a Greenplum database from a Data Domain system with `gpdbrestore` and the `--ddboost` option indicates success even though segment failures occured during the restore. **Workaround:** The errors are logged in the master and segment server backup or restore status and report files. Scan the status and report files to check for error messages.
15692 17192	Backup and Restore	Greenplum Database's implementation of RSA lock box for Data Domain Boost changes backup and restore requirements for customers running SuSE. The current implementation of the RSA lock box for Data Domain Boost login credential encryption only supports customers running on Red Hat Enterprise Linux. Workaround: If you run Greenplum Database on SuSE, use NFS as your backup solution. See the *Greenplum Database Administrator Guide* for information on setting up a NFS backup.
18850	Backup and Restore	Data Domain Boost credentials cannot be set up in some environments due to the absence of certain libraries (for example, `libstdc++`) expected to reside on the platform. **Workaround:** Install the missing libraries manually on the system.
18851	Backup and Restore	When performing a data-only restore of a particular table, it is possible to introduce data into Greenplum Database that contradicts the distribution policy of that table. In such cases, subsequent queries may return unexpected and incorrect results. To avoid this scenario, we suggest you carefully consider the table schema when performing a restore.

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

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

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

Upgrading to Greenplum Database 4.3.11.1

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.11.1. 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.

- Verify the health of the Greenplum Database host hardware, and that you verify that the hosts meet the requirements for running Greenplum Database. The Greenplum Database `gpcheckperf` utility can assist you in confirming the host requirements.
- If upgrading from Greenplum Database 4.2.x.x, Pivotal recommends running the `gpcheckcat` utility to check for Greenplum Database catalog inconsistencies.

  Note: If you need to run the `gpcheckcat` utility, Pivotal recommends running it a few weeks before the upgrade and that you run `gpcheckcat` during a maintenance period. If necessary, you can resolve any issues found by the utility before the scheduled upgrade.

  The utility is in `$GPHOME/bin`. 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*.
- Ensure that the Linux `sed` utility is installed on the Greenplum Database hosts. In Greenplum Database releases prior to 4.3.10.0, the Linux `ed` utility is required on Greenplum Database hosts. The `gpinitsystem` utility requires the Linux utility.

- During the migration process from Greenplum Database 4.2.x.x, a backup is made of some files and directories in `$MASTER_DATA_DIRECTORY`. Pivotal recommends that files and directories that are not used by Greenplum Database be backed up, if necessary, and removed from the `$MASTER_DATA_DIRECTORY` before migration. For information about the Greenplum Database migration utilities, see the *Greenplum Database Utility Guide*.

  Important: If you intend to use an extension package with Greenplum Database 4.3.11.1, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later.

  If you use the Greenplum Database MADlib extension, Pivotal recommends that you upgrade to MADlib 1.9.1 on Greenplum Database 4.3.11.1. If you do not upgrade to MADlib 1.9.1, the MADlib `madpack` utility will not function. The MADlib analytics functionality will continue to work. See *Modifying the MADlib madpack Utility*

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

- *Upgrading from 4.3.x to 4.3.11.1*
- *Upgrading from 4.3.x to 4.3.11.1 on Pivotal DCA Systems*
- *Upgrading from 4.2.x.x to 4.3.11.1*
- *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 4.3.11.1

An upgrade from 4.3.x to 4.3.11.1 involves stopping Greenplum Database, updating the Greenplum Database software binaries, upgrading and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 or later extension packages. If you are using custom modules with the extensions, you must also use modules that were built for use with Greenplum Database 4.3.5.0 or later.

Important: If you are upgrading from Greenplum Database 4.3.x on a Pivotal DCA system, see *Upgrading from 4.3.x to 4.3.11.1 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.11.1 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 *11*). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`

For information about the utility, see *fix_ao_upgrade.py Utility*.

Note: If your database contains append-optimized tables that were converted from Greenplum Database 4.2.x append-only tables, and you are upgrading from a 4.3.x release earlier than 4.3.6.0, run the `fix_visimap_owner.sql` script to fix a Greenplum Database append-optimized

table issue (See step *12*). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`

For information about the script, see *fix_visimap_owner.sql Script*.

Note: If the Greenplum Command Center database `gpperfmon` is installed in your Greenplum Database system, the migration process changes the distribution key of the Greenplum Database *log_alert_** tables to the `logtime` column. The redistribution of the table data might take some time the first time you start Greenplum Database after migration. The change occurs only the first time you start Greenplum Database after a migration.

1. Log in to your Greenplum Database master host as the Greenplum administrative user:

```
$ su - gpadmin
```

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

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

3. Perform a smart shutdown of your current Greenplum Database 4.3.x system (there can be no active connections to the database). This example uses the `-a` option to disable confirmation prompts:

```
$ gpstop -a
```

4. Run the installer for 4.3.11.1 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-4.3.11.1
```

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-4.3.11.1/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-4.3.11.1 /usr/local/greenplum-db
```

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

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

7. Run the `gpseginstall` utility to install the 4.3.11.1 binaries on all the segment hosts specified in the *hostfile*. For example:

```
$ gpseginstall -f hostfile
```

8. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.

9. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, download the corresponding packages from *Pivotal Network*, and install using this utility. See the *Greenplum Database 4.3 Utility Guide* for `gppkg` usage details.
10. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

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

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

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

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

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

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

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

    c. (*optional*) Run the `fix_ao_upgrade.py` utility with the option `--report` again. No inconsistencies should be reported.
12. For databases that contain append-optimized tables that were created from Greenplum Database 4.2.x append-only tables, run the `fix_visimap_owner.sql` script. The script resolves an issue associated with relations associated with append-optimized tables. For example, this command runs the script on the database `testdb`.

```
$ psql -d testdb1 -f $GPHOME/share/postgresql/upgrade/
fix_visimap_owner.sql
```

    The script displays this prompt that allows you to display changes to the affected relations without performing the operation.

```
Dry run, without making any modifications (y/n)?
```

    - Enter `y` to list ownership changes that would have been made. The owner of the relation is not changed.
    - Enter `n` make the ownership changes and display the changes to relation ownership.

    Note: Pivotal recommends that you run the script during low activity period. Heavy workloads do not affect database functionality but might affect performance.
13. If you are utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.3.x to 4.3.11.1 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.

Modifying the MADlib madpack Utility

Installing or upgrading the MADlib extension to MADlib 1.9.1 on Greenplum Database 4.3.11.1 requires modifying the `madpack` installation utility to make the utility compatible with Greenplum Database 4.3.11.1.

After you upgrade to Greenplum Database 4.3.11.1, complete these steps:

1. Download and unpack the MADlib 1.9.1 binary distribution from *Pivotal Network*.
2. Install or upgrade the MADlib extension with the Greenplum Database utility `gppkg`. The `-i` option to installs an extension, the `-u` option to upgrades a extension. This example upgrades the MADlib extension.

   ```
   $ gppkg -u path-to/madlib-ossv1.9.1_pv1.9.6_gpdb4.3orca-rhel5-x86_64.gppkg
   ```

3. Execute the `fix_madpack.sh` script. You can provide the path to the MADlib installation with the `--prefix` option.

   ```
   $ fix_madpack.sh --prefix madlib-installation-path
   ```

   If you do not include the `--prefix` option, the script uses the location `$GPHOME/madlib`.
4. Install or upgrade MADlib using the `madpack` utility. The `install` command installs MADlib, the `upgrade`command upgrades MADlib. For example, this `madpack` command upgrades MADlib.

   ```
   $ madpack -p greenplum upgrade
   ```

For information about the Greenplum Database MADlib extension including installing and upgrading the MADlib, See "Greenplum MADlib Extension for Analytics" in the *Greenplum Database Reference Guide*.

fix_visimap_owner.sql Script

The SQL script `fix_visimap_owner.sql` resolves ownership issues related to visimap relations that are associated with append-optimized tables.

When upgrading from Greenplum Database 4.2.x to 4.3.x, the 4.2.x append-only tables are converted to 4.3 append-optimized tables. When upgrading from 4.2.x to Greenplum Database 4.3.x earlier than 4.3.6.0, the upgrade process incorrectly assigned the owner of visimap relations to gpadmin, not the owner of the associated append-optimized table.

If you are migrating to this release Greenplum Database from a 4.3.x release earlier than 4.3.6.0, run this SQL script as the `gpadmin` superuser to fix the incorrect assignment issue for a database.

```
$GPHOME/share/postgresql/upgrade/fix_visimap_owner.sql
```

When you run the script, it temporarily creates two functions that update the visimap relations ownership and displays this message that lets you perform a test run without changing ownership.

```
Dry run, without making any modifications (y/n)?
```

If you enter `y`, the script displays the changes that would have been made. The owner of the relation is not changed.

If you enter `n`, the script changes the owner of the relations and displays the changes that are made.

Before exiting, the script deletes the functions it created.

> **Note:** If you are migrating from Greenplum Database 4.2.x directly to Greenplum Database 4.3.11.1 you do not need to run the `fix_visimap_owner.sql` script. Also, you can run this script on Greenplum Database 4.3.x earlier than 4.3.6.0 to fix the incorrect ownership assignment of visimap relations.

fix_ao_upgrade.py Utility

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

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

Syntax

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

Options

-r | --report

Report inconsistencies without making any changes.

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

Greenplum Database master hostname or IP address.

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

Greenplum Database master port.

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

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

v | --verbose

Verbose output that includes table names.

--help

Show the help message and exit.

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

Dropping Orphan Tables on Greenplum Database Segments

If you upgraded to Greenplum Database 4.3.6.0 and a user dropped a table, in some cases, the table would be dropped only on the Greenplum Database master, not on the Greenplum Database segments. This created orphan tables on Greenplum Database segments. This issue occurs only with Greenplum Database 4.3.6.0. However, the orphan tables remain in Greenplum Database after upgrading to 4.3.11.1.

For Greenplum Database 4.3.6.2 and later, the installation contains this Python script to check for and drop orphan tables on segments.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py
```

You can run this script on Greenplum Database 4.3.11.1 to check for and drop orphan tables.

The script performs these operations:

- Checks for orphan tables on segments and generates file that contains a list of the orphan tables.
- Deletes orphan tables specified in a text file.

You run the script as a Greenplum Database administrator. The script attempts to log into Greenplum Database as user who runs the script.

To check all databases in the Greenplum Database instance, run this command on the Greenplum Database master. Specify the *port* to connect to Greenplum Database.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py -p port
```

To check a single database, specify the option `-d database`.

The command generates a list of orphan tables in the text file `orphan_tables_file_timestamp`. You can review the list and, if needed, modify it.

To delete orphan tables on the Greenplum Database segments, run this command on the Greenplum Database master. Specify the *port* to connect to Greenplum Database and the file containing the orphan tables to delete.

```
$GPHOME/share/postgresql/upgrade/fix_orphan_segment_tables.py -p port -f
 orphan_tables_file_timestamp
```

The script connects only to the databases required to drop orphan tables.

> **Note:** Pivotal recommends that you run the script during a period of low activity to prevent any issues that might occur due to concurrent drop operations.

Upgrading from 4.3.x to 4.3.11.1 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.11.1 on a Pivotal DCA system involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database. If you are using Greenplum Extension packages, you must install and use Greenplum Database 4.3.5.0 or later extension packages. If you are using custom modules with the extensions, you must also use modules that were built for use with Greenplum Database 4.3.5.0 or later.

> **Important:** Skip this section if you are *not* installing Greenplum Database 4.3.11.1 on DCA systems. This section is only for installing Greenplum Database 4.3.11.1 on DCA systems.
>
> **Note:** If you are upgrading from Greenplum Database between 4.3.0 and 4.3.2, run the `fix_ao_upgrade.py` utility to check Greenplum Database for the upgrade issue and fix the upgrade issue (See step *8*). The utility is in this Greenplum Database directory: `$GPHOME/share/postgresql/upgrade`
>
> For information about the utility, see *fix_ao_upgrade.py Utility*.

1. Log in to your Greenplum Database master host as the Greenplum administrative user (`gpadmin`):

   ```
   # su - gpadmin
   ```

2. Download or copy the installer file to the Greenplum Database master host.
3. Uninstall the Greenplum Database gNet extension package if it is installed. For information about uninstalling a Greenplum Database extension package, see `gppkg` in the *Greenplum Database Utility Guide*.

   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 4.3.11.1 on the Greenplum Database master host and specify the file `hostfile` that lists all hosts in the cluster. If necessary, copy `hostfile` to the directory containing the installer before running the installer.

This example command runs the installer for Greenplum Database 4.3.11.1 for Redhat Enterprise Linux 5.x.

```
# ./greenplum-db-appliance-4.3.11.1-build-1-RHEL5-x86_64.bin hostfile
```

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

6. Install Greenplum Database extension packages. For information about installing a Greenplum Database extension package, see `gppkg` in the *Greenplum Database Utility Guide*.

   Important: Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.

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

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

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

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

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

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

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

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

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

9. If you are utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.3.x to 4.3.11.1 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 4.3.11.1

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.11.1. 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.
>
> **Important:** If you intend to use an extension packages with Greenplum Database 4.3.5.0 or later, you must install and use a Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later.

Migrating a Greenplum Database That Contains Append-Only Tables

The migration process converts append-only tables that are in a Greenplum Database to append-optimized tables. For a database that contains a large number of append-only tables, the conversion to append-optimized tables might take a considerable amount of time. Pivotal supplies a user-defined function that can help estimate the time required to migrate from Greenplum Database 4.2.x to 4.3.x. For information about the user-defined function, *estimate_42_to_43_migrate_time.pdf*.

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

Upgrade Procedure

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

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

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

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

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

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

   ```
   # su - gpadmin
   ```

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

   ```
   $ vacuumdb database_name
   ```

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

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

   Running `VACUUM` and cleaning out old logs files is not required, but it will reduce the size of Greenplum Database files to be backed up and migrated.
4. Run `gpstate` to check for failed segments.

```
$ gpstate
```

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

```
$ gprecoverseg
```

   Note: It might be necessary to restart the database if the preferred role does not match the current role; for example, if a primary segment is acting as a mirror segment or a mirror segment is acting as a primary segment.
6. Copy or preserve any additional folders or files (such as backup folders) that you have added in the Greenplum data directories or `$GPHOME` directory. Only files or folders strictly related to Greenplum Database operations are preserved by the migration utility.

Install the Greenplum Database 4.3 Software Binaries (non-DCA)

> **Important:** If you are installing Greenplum Database 4.3 on a Pivotal DCA system, see *Install the Greenplum Database 4.3 Software Binaries on DCA Systems*. This section is for installing Greenplum Database 4.3 on non-DCA systems.

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

```
# unzip greenplum-db-4.3.11.1-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.11.1-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-4.3.11.1`), 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 4.3.11.1 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.11.1/greenplum_path.sh
```

8. Run the `gpseginstall` utility to install the 4.3.11.1 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 4.3.11.1 on the Greenplum Database master host and specify the file `hostfile` that lists all hosts in the cluster. If necessary, copy `hostfile` to the directory containing the installer before running the installer.

   This example command runs the installer for Greenplum Database 4.3.11.1.

   ```
   # ./greenplum-db-appliance-4.3.11.1-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-4.2.8.1/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 4.3.11.1 installation. For example:

   ```
   $ source /usr/local/greenplum-db-4.3.11.1/greenplum_path.sh
   ```

   On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.11.1/greenplum_path.sh`.
6. Update the Greenplum Database environment so it is referencing your new 4.3.11.1 installation.

   a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.11.1 installation directory. For example (as root):

   ```
   # rm -rf /usr/local/greenplum-db
   # ln -s /usr/local/greenplum-db-4.3.11.1 /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-4.3.11.1 /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-4.3.11.1 /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-4.3.11.1 /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-4.3.11.1
```

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-4.3.11.1`.

8. As `gpadmin`, run the 4.3.11.1 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-4.3.11.1
```

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-4.3.11.1`.

> **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 4.3.11.1 system will be running and accepting connections.

> **Note:** After the migration utility has completed, the resynchronization of the mirror segments with the primary segments continues. Even though the system is running, the mirrors are not active until the resynchronization is complete.

Post-Upgrade (on your 4.3.11.1 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.x Load Tools package. Application Packages are available at *Pivotal Network*. For information about `gpfdist`, see the *Greenplum Database 4.3 Administrator Guide*.
3. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later. (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.
4. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, download the corresponding packages from *Pivotal Network*, and install using this utility. See the *Greenplum Database Utility Guide* for `gppkg` usage details.
5. If you want to utilize the Greenplum Command Center management tool, install the latest Command Center Console and update your environment variable to point to the latest Command Center binaries (source the `gpperfmon_path.sh` file from your new installation). See the Greenplum Command Center documentation for information about installing and configuring Greenplum Command Center.

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

   Command Center Console packages are available from *Pivotal Network*.
6. (*optional*) Check the status of Greenplum Database. For example, you can run the Greenplum Database utility `gpstate` to display status information of a running Greenplum Database.

   ```
   $ gpstate
   ```
7. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.11.1 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 4.3.11.1. Detailed upgrade instructions are provided in *Upgrading from 4.2.x.x to 4.3.11.1*.

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

☐	Inform all database users of the upgrade and lockout time frame.
Upgrade Execution * The system will be locked down to all user activity during the upgrade process	
☐	Backup your current databases.
☐	Remove the standby master (`gpinitstandby -r`).
☐	Do a clean shutdown of your current system (`gpstop`).
☐	Update your environment to source the new Greenplum Database 4.3.x installation.
☐	Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
☐	After the upgrade process finishes successfully, your 4.3.x system will be up and running.
Post-Upgrade (on your 4.3 system) * The 4.3.x.x system is up	
☐	Reinitialize your standby master host (`gpinitstandby`).
☐	Upgrade `gpfdist` on all of your ETL hosts.
☐	Rebuild any custom modules against your 4.3.x installation.
☐	Download and install any Greenplum Database extensions.
☐	(Optional) Install the latest Greenplum Command Center and update your environment to point to the latest Command Center binaries.
☐	Inform all database users of the completed upgrade.

For Users Running Greenplum Database 4.1.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.11.1.

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 4.3.11.1*.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.11.1.

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 4.3.11.1*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.11.1.

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 4.3.11.1*.

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

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC[1]) PostgreSQL Client C API (libpq)	4.3	4.3
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.3	4.3
Greenplum Command Center	Greenplum Database management tool.	1.3.0.2	4.3

Note: [1]The JDBC drivers that are shipped with the Greenplum Connectivity Tools are official PostgreSQL JDBC drivers built by the PostgreSQL JDBC Driver team (*https://jdbc.postgresql.org*).

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

- AIX 5.3L (32-bit)
- AIX 5.3L and AIX 6.1 (64-bit)
- Apple OS X 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 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)

Important: Support for SuSE Linux Enterprise Server 64-bit 10 SP4 has been dropped for Greenplum Database 4.3.11.1.

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.
>
> Pivotal supplies separate PL/Perl extension packages for Red Hat Enterprise Linux 7.x, 6.x and 5.x. Ensure you install the correct package for your operating system.

Table 5: Greenplum Database Extensions Compatibility

Greenplum Database Extension	Extension Components	
	Name	**Version**
PostGIS 2.0.1 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PL/Java 1.3 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 2.2 for Greenplum Database 4.3.x.x	PL/R	8.3.0.16
	R	3.1.1
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
PL/Perl 1.2 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.16.3 on RHEL 7.x 5.12.4 on RHEL 6.x 5.5.8 on RHEL 5.x
PL/Perl 1.1 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x
PL/Perl 1.0 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x
Pgcrypto 1.2 for Greenplum Database 4.3.x.x	Pgcrypto	Based on PostgreSQL 8.3

Greenplum Database Extension	Extension Components	
	Name	Version
MADlib 1.9.x for Greenplum Database 4.3.x.x	MADlib	Based on MADlib version 1.9.x

Note: Greenplum Database 4.3.11.1 does not support the PostGIS 1.0 extension package.

Only MADlib 1.9.1, or 1.9 or can be installed on Greenplum Database 4.3.11.1 and later releases. Pivotal recommends that you upgrade to MADlib 1.9.1 on Greenplum Database 4.3.11.1. If you do not upgrade MADlib, the MADlib `madpack` utility will not function on Greenplum Database 4.3.11.1. The MADlib analytics functionality will continue to work. See *Modifying the MADlib madpack Utility*

Greenplum Database 4.3.11.1 supports these minimum Greenplum Database extensions package versions.

Table 6: Greenplum Database 4.3.11.1 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1 and release `gpdb4.3orca`
PL/Java	1.3 and release `gpdb4.3orca`
PL/Perl	1.2 and release `gpdb4.3orca`
PL/R	2.1 and release `gpdb4.3orca`
Pgcrypto	1.2 and release `gpdb4.3orca`
MADlib	1.9.5 and release `gpdb4.3orca`

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

To use extension packages with Greenplum Database 4.3.11.1, you must install and use Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 or later. For custom modules that were used with Greenplum Database 4.3.4.x and earlier, you must rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.11.1.

Package File Naming Convention

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

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

This example is the package name for a postGIS package.

```
postgis-ossv2.0.3_pv2.0.1_gpdb4.3-rhel5-x86_64.gppkg
```

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

The version is the version of the open source software that the package is based on. For the postGIS package, ossv2.0.3 specifies that the package is based on postGIS version 2.0.3.

`pv`*pkg-version* - The package version. The version of the Greenplum Database package. For the postGIS package, pv2.0.1 specifies that the Greenplum Database package version is 2.0.1.

`gpdb`*rel-OS-version-arch* - The compatible Greenplum Database release. For the postGIS package, gpdb4.3-rhel5-x86_64 specifies that package is compatible with Greenplum Database 4.3 on Red Hat Enterprise Linux version 5.x, x86 64-bit architecture.

Hadoop Distribution Compatibility

This table lists the supported Hadoop distributions:

Table 7: Supported Hadoop Distributions

Hadoop Distribution	Version	gp_hadoop_ target_version
Pivotal HD	Pivotal HD 3.0, 3.0.1	`gphd-3.0`
	Pivotal HD 2.0, 2.1 Pivotal HD 1.0 [1]	`gphd-2.0`
Greenplum HD	Greenplum HD 1.2	`gphd-1.2`
	Greenplum HD 1.1	`gphd-1.1` (default)
Cloudera	CDH 5.2, 5.3, 5.4.x - 5.8.x	`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, 2.3	`hdp2`
MapR [3]	MapR 4.x, MapR 5.x	`gpmr-1.2`
	MapR 1.x, 2.x, 3.x	`gpmr-1.0`
Apache Hadoop	2.x	`hadoop2`

Notes:

1. Pivotal HD 1.0 is a distribution of Hadoop 2.0
2. For CDH 4.1, only CDH4 with MRv1 is supported
3. MapR requires the MapR client. For MapR 5.x, only `TEXT` and `CSV` are supported in the `FORMAT` clause of the `CREATE EXTERNAL TABLE` command.

Greenplum Database 4.3.11.1 Documentation

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

Table 8: Greenplum Database Documentation

Title	Revision
Greenplum Database 4.3.11.1 Release Notes	A01
Greenplum Database 4.3 Installation Guide	A16
Greenplum Database 4.3 Administrator Guide	A21
Greenplum Database 4.3 Reference Guide	A22
Greenplum Database 4.3 Utility Guide	A23
Greenplum Database 4.3 Client Tools for UNIX	A08
Greenplum Database 4.3 Client Tools for Windows	A06
Greenplum Database 4.3 Connectivity Tools for UNIX	A07
Greenplum Database 4.3 Connectivity Tools for Windows	A06
Greenplum Database 4.3 Load Tools for UNIX	A10
Greenplum Database 4.3 Load Tools for Windows	A10
Greenplum Command Center Administrator Guide *	----
Greenplum Workload Manager User Guide *	----

Note: * HTML format only. Documentation is at *gpcc.docs.pivotal.io*.

Copyright