

Greenplum Database 4.3.33.7 Release Notes

Rev: A01

Updated: May, 2020

Welcome to Pivotal Greenplum Database 4.3.33.7

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.

Important: For Greenplum Database 4.3.16.0 and later, the pgcrypto extension has been updated to package version pv1.3.

- Previous releases of the pgcrypto extension are not compatible with Greenplum Database 4.3.16.0 and later.
- The pgcrypto extension package version pv1.3 is not compatible with previous Greenplum Database releases.

For information about the pgcrypto extension package, see *Greenplum Database Extensions*.

Important: The Greenplum `gpbackup` and `gprestore` utilities are now distributed separately from Pivotal Greenplum Database, and are updated independently of the core Greenplum Database server. These utilities will not be updated in future Greenplum Database 4.3.x releases. You can upgrade to the latest `gpbackup` and `gprestore` versions by downloading and installing the latest Pivotal Greenplum Backup and Restore release from *Pivotal Network*.

Note: This document contains pertinent release information about Greenplum Database 4.3.33.7. For previous versions of the release notes for Greenplum Database, go to *Pivotal Greenplum Database Documentation*. For information about Greenplum Database end of life, see the *Pivotal Support Lifecycle Policy*.

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

About Greenplum Database 4.3.33.7

Greenplum Database 4.3.33.7 is a maintenance release that resolves some issues. Please refer to the following sections for more information about this release.

- *Resolved Issues in Greenplum Database 4.3.33.x*
- *Supported Platforms*

- [Downloading Greenplum Database](#)
- [Beta Features](#)
- [Known Issues in Greenplum Database 4.3.33.7](#)
- [Upgrading to Greenplum Database 4.3.33.7](#)
- [Greenplum Database Tools Compatibility](#)
- [Greenplum Database Extensions](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.3.33.7 Documentation](#)

Resolved Issues in Greenplum Database 4.3.33.x

The table below lists issues that are now resolved in Pivotal Greenplum Database 4.3.33.x

For issues resolved in prior 4.3 releases, refer to the corresponding release notes. Release notes are available from the Pivotal Greenplum page on [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 1: Resolved Issues in 4.3.33.x

Issue Number	Category	Resolved In	Description
9127	Storage	4.3.33.7	Data file blocks could be corrupted during concurrent writes to multiple tables, because of a bug in correlating a table to its blocks in the storage module. This problem has been resolved.
30585	Locking	4.3.33.7	Resolved a problem that could corrupt resource queue locks, and potentially other types of locks, in shared memory. This problem could cause errors such as <code>lock lock_name on object object_identifier is already held</code> .
30506	Server: Execution	4.3.33.7	Resolved an invalid memory context that could occur when executing subplans in certain complex queries. This problem could lead to a segmentation fault associated with the internal <code>ExecScanSubplan</code> code.
29709	gpfdist	4.3.33.6	Some messages issued by Greenplum Database as part of a <code>gpfdist</code> session, such as messages about the status of a retry operation, were labeled as warning messages. In some cases, these warning messages were treated as errors by a client ETL application and would cause the application to fail. The warning label has been removed from the messages.
30311	gpfdist	4.3.33.6	In some cases, <code>gpfdist</code> incorrectly logged an HTTP error code 500 for a <code>gpfdist</code> session after the session was closed. Now the <code>gpfdist</code> does not log the error for closed sessions.
30391	Planner	4.3.33.6	Some queries returned incorrect results when the queries contain subqueries that perform a join and also contain one or more equality predicates. Incorrect results were returned when either a merge join or a nested loop join did not correctly process the predicates. This issue is resolved.

Issue Number	Category	Resolved In	Description
30382	VACUUM, TRUNCATE	4.3.33.6	In some cases, performing <code>VACUUM FULL</code> operation on the <code>pg_class</code> catalog table and concurrently performing a <code>TRUNCATE</code> operation on a user created heap table returned the error <code>updated tuple is already HEAP_MOVED_OFF</code> and caused the database to become unavailable. The <code>TRUNCATE</code> command did not properly manage the heap table entry in <code>pg_class</code> during the <code>TRUNCATE</code> operation. This issue is resolved.
30452	Dispatch	4.3.33.6	If the server configuration parameter <code>check_function_bodies</code> was set in a session on the master, the parameter setting did not persist when a related segment instance session was reset. This caused some functions to fail. Now the parameter setting persists when a segment instance session is reset.
30287	Server: Execution	4.3.33.6	When GPORCA was enabled, queries against an append-only, column-oriented table could cause a PANIC due to shared memory corruption. The code was modified to guard against out-of-bound writes that caused the memory corruption.
30274	Query Optimizer	4.3.33.5	GPORCA would fall back to using the Postgres Planner if a statement inserted data into a randomly-distributed table and the target column for the insert had a <code>NOT NULL</code> constraint. GPORCA was fixed so that it generates a plan for these statements.
30334	Storage: DDL	4.3.33.5	A system panic could occur if <code>ALTER TABLE</code> was used to add a new partition, and <code>WITH (OIDS=FALSE)</code> was specified as the only storage parameter for the new partition. The problem was caused by code that failed to handle the possibility of a null <code>reloption</code> value, which is generated when the single default storage parameter is specified. The partitioning code was modified to correctly handle the possibility of such null values.
29984	Server	4.3.33.4	During start up, idle query executor (QE) processes can commit up to 16MB of memory each, but they are not tracked by the Linux virtual memory tracker. In a worst-case scenario, these idle processes could trigger OOM errors that were difficult to diagnose. To prevent these situations, Greenplum now hard-codes a startup memory cost to account for untracked QE processes.
30071	Catalog and Metadata	4.3.33.4	In some cases, an <code>Exclusive</code> lock was acquired on the system catalog table <code>pg_class</code> preventing concurrent updates to the table. This issue has been resolved. Now a <code>Row Exclusive</code> lock is used to update <code>pg_class</code> .
30098	COPY	4.3.33.4	Greenplum Database generated a PANIC when a <code>COPY</code> command attempted to write to a catalog table. This issue has been resolved, now Greenplum Database returns an error.

Issue Number	Category	Resolved In	Description
30106	Server	4.3.33.4	In some cases, DDL and DML commands ran slower than usual due to mishandling of <code>fsync</code> -related messages by the backend. This issue is now resolved.
30151	Catalog and Metadata	4.3.33.4	Queries failed with an unexpected internal error if the server configuration parameter <code>search_path</code> contained a large number of schemas because of the <code>search_path</code> length limit. This issue is resolved. The <code>search_path</code> length limit has been increased significantly.
30272	Access Methods	4.3.33.4	Inserting a large amount of data into a partitioned, append-optimized column-oriented table was slow due to Greenplum Database memory management issues. This issue has been resolved. Greenplum Database memory management has been improved for the specified situation.
30027	Planner	4.3.33.3	When creating a view with multiple Set Operations together with window functions, the planner would incorrectly push down the qualification clause outside the window functions. As a result, the query execution would fail with <code>optimizer=off</code> and only succeed with <code>optimizer=on</code> . This issue has now been resolved.
30070	VACUUM	4.3.33.3	A mismatch in segment file state on the master and a segment during a <code>VACUUM</code> operation could cause a subsequent operation on the relation to return an error. The error occurred when the query executor (QE) did not update the segment file state appropriately when it could not acquire an exclusive lock on an AO/AOCO relation during the <code>VACUUM</code> . This issue has been resolved. The QE now waits for the lock, which ensures that the segment file state is consistent between master and segment.
30058	Query Execution	4.3.33.3	Running <code>EXPLAIN ANALYZE</code> on some queries generates a Greenplum Database PANIC. The error occurs when an analyze operation was dispatched to a query executor (QE). This issue has been fixed, analyze operations can be dispatched to a QE.
30020	Planner	4.3.33.3	A query caused a panic when an upper-level predicate was pushed down into a subquery that was not pushdown-safe because the predicate referenced a set-returning function in a subquery output column. The planner failed to detect the subquery was not pushdown-safe because of nested unions or intersections. This has been fixed.
167886881	COPY	4.3.33.3	When a segment has more than 2,147,483,647 rows, the <code>COPY</code> command would load the data correctly but the validation count overflowed the <code>int32</code> variable. This issue was resolved by using an <code>int64</code> data type.

Issue Number	Category	Resolved In	Description
30021	Dispatch	4.3.33.2	In some cases, Greenplum Database generated a SIGSEGV when <code>pg_cancel_backend()</code> or <code>pg_terminate_backend()</code> was used to terminate user connections to Greenplum Database. The error occurred when processes associated with the user session were not properly terminated during session start up. This issue has been resolved. Greenplum Database has improved how processes are terminated during process start up.
29997	Process Management	4.3.33.2	In some cases, the database entered recovery mode when the process management module incorrectly destroyed resources when the <code>InitPlan</code> for a query completed. This issue has been resolved. The process management module now correctly sets the process state and properly reuses backend processes.
29991	Server	4.3.33.2	<code>ALTER TABLE ... DROP COLUMN</code> followed by an <code>ALTER TABLE ... SET WITH (REORGANIZE=true)</code> in some cases may lose the column encoding settings of the dropped column. This problem manifested itself later as a block version mismatch error when the table was further altered. This issue is resolved. The Greenplum Database server now preserves column encoding settings for dropped columns when redistributing the data.
29887	Server	4.3.33.2	The query executor passed an incorrect address for the in-progress transaction array. This problem may have affected extended queries such as <code>DECLARE ... CURSOR</code> statements and <code>RETURN</code> and <code>EXECUTE PL/pgSQL</code> statements. This issue has been fixed.
29881	Server COPY	4.3.33.2	An internal function used to execute the <code>COPY</code> statement did not properly check its connection state. This could lead to a situation where a segment failure caused a <code>COPY</code> statement to fail, and the statement could not be canceled using the <code>pg_terminate_backend()</code> function. This issue has been resolved.
29752	Server	4.3.33.2	An <code>INSERT</code> operation on an append-optimized partitioned table returned an error when a concurrent <code>VACUUM</code> operation was performed on the table. The error occurred because <code>INSERT</code> did not lock the child partition tables. This issue has been resolved. Now <code>INSERT</code> locks the child partition tables.
29424	Dispatch	4.3.33.2	Greenplum Database performance was poor for some DDL commands such as creating a partitioned table with a large number of partitions. When executing the DDL command, Greenplum Database transfers a large amount of data to segment instances. The amount of data being transferred caused performance issues. This issue has been resolved. Greenplum Database has reduced the amount of data being transferred in the specified situation.

Issue Number	Category	Resolved In	Description
29752	Server	4.3.33.1	An <code>INSERT</code> operation on an append-optimized partitioned table returned an error when a concurrent <code>VACUUM</code> operation was performed on the table. The error occurred because <code>INSERT</code> did not lock the child partition tables. This issue has been resolved. Now <code>INSERT</code> locks the child partition tables.
26225	gpcheckcat	4.3.33.1	The <code>gpcheckcat</code> utility failed to generate a summary report if there was an orphan <code>TOAST</code> table entry in one of the segments. This is fixed. The string "N/A" is reported when there is no relation oid to report.
29963	Planner	4.3.33.1	Due to a bug in the planner, users may have encountered a query failure with the error message "Could not devise a query plan for the given query" when the query included a table join, an <code>IN</code> clause was present in the <code>WHERE</code> condition, and the <code>WHERE</code> condition was in conflict with one of table constraints. Greenplum 4.3.33.1 resolves this issue by properly handling path de-duplication logic in the planner.
29563	gpperfmon	4.3.33.1	Fixed a problem where <code>gpperfmon_install</code> would write the clear text password to its output. The code was modified to replace the password text with asterisks (*****).
167267884	gpcrondump	4.3.33.1	Increase the maximum value of the server configuration parameter <code>extra_float_digits</code> to 3 and enable <code>gpcrondump</code> to use that value. This fixes a problem where <code>float4</code> columns used as distribution keys were not backed up with sufficient precision to match the original distribution hash when the backup was restored.
29887	Server	4.3.33.1	The query executor passed an incorrect address for the in-progress transaction array. This problem may have affected extended queries such as <code>DECLARE ... CURSOR</code> statements and <code>RETURN</code> and <code>EXECUTE PL/pgSQL</code> statements. This issue has been fixed.
29881	Server COPY	4.3.33.1	An internal function used to execute the <code>COPY</code> statement did not properly check its connection state. This could lead to a situation where a segment failure caused a <code>COPY</code> statement to fail, and the statement could not be canceled using the <code>pg_terminate_backend()</code> function. This issue has been resolved.
165950889	Query Planner	4.3.33.0	Earlier Greenplum Database versions do not support Window expressions that use a subquery in the <code>PRECEDING</code> or <code>FOLLOWING</code> clause to define the window range. In versions prior to 4.3.30, such a query would generate the error <code>no parameter found for initplan subquery</code> . In versions 4.3.30 to 4.3.32 no error was displayed, but this type of query would generate an incorrect query plan. Greenplum 4.3.33 resolves this problem by adding support for a subquery in the <code>PRECEDING</code> or <code>FOLLOWING</code> clause.

Issue Number	Category	Resolved In	Description
29597, 165780346	Storage: Transaction Management	4.3.33.0	<p>Fixed a problem that could cause a panic to occur after a master segment restart if a superuser role had a non-null value for <code>pg_authid.rolvaliduntil</code>. (A null value in this column indicates that the role was originally created without an expiration date.)</p> <p>When selecting a superuser role to perform distributed transaction recovery, Greenplum skipped any account that had a non-null <code>rolvaliduntil</code> value. Later, attempting to free resources for the unselected (null) superuser role would fail with a panic.</p> <p>The distributed transaction management code was modified to correctly validate each superuser role, and to only reject those roles that have a <code>rolvaliduntil</code> date that occurred in the past. In the event that <i>all</i> available superuser roles have expired, Greenplum displays an error and no panic condition occurs.</p>
29728	Query Optimizer	4.3.33.0	<p>Fixed a problem in the Pivotal Query Optimizer algebraizer where the optimizer incorrectly normalized subqueries in a query with a complex <code>GROUP BY</code> target list. This included an aggregate function referenced in a <code>HAVING</code> clause or a <code>GROUP BY</code> target list. In some cases, such a query fell back to the Postgres Planner. The Pivotal Query Optimizer now correctly supports subqueries with <code>GROUP BY</code> target lists that contain outer references, even those in other (nested) subqueries.</p>
29767	Storage: Catalog and Metadata	4.3.33.0	<p>When a table has both a <code>PRIMARY KEY</code> and a <code>DISTRIBUTED BY</code> clause, the <code>DISTRIBUTED BY</code> columns must be equal to or a left-subset of the <code>PRIMARY KEY</code>. However, this rule could be broken by altering the <code>DISTRIBUTED BY</code> columns after the table is created. It is also possible to <code>DROP</code> or <code>ADD</code> a primary key constraint to break this rule. This issue is resolved. The rule is now enforced when altering the distribution key or creating a unique index for the primary key.</p>
29838	Server: Execution	4.3.33.0	<p>Fixed a problem where <code>pg_cancel()</code> would fail to cancel a query due to writer processes waiting for acknowledgement from reader processes that had already been cancelled. The code was changed to break this wait loop during a cancel operation.</p>

Supported Platforms

Greenplum Database 4.3.33.7 runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 7.x (See the following *Note*)
- 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 (deprecated)
- Oracle Unbreakable Linux 64-bit 5.5

- CentOS 64-bit 7.x
- CentOS 64-bit 6.x
- CentOS 64-bit 5.x

Note: For the supported Linux operating systems, Pivotal Greenplum Database is supported on system hosts using either AMD or Intel CPUs based on the x86-64 architecture. Pivotal recommends using a homogeneous set of hardware (system hosts) in a Greenplum Database system.

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: 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 4.3.33.7 software that runs on Linux systems uses OpenSSL 1.0.2l (with FIPS 2.0.16), cURL 7.54, OpenLDAP 2.4.44, and Python 2.6.9.

Greenplum Database client software that runs on Windows and AIX systems uses OpenSSL 0.9.8zg.

The Greenplum Database s3 external table protocol supports these data sources:

- Amazon Simple Storage Service (Amazon S3)
- *Dell EMC Elastic Cloud Storage* (ECS), an Amazon S3 compatible service

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 2: Data Domain Boost Compatibility

Greenplum Database	Data Domain Boost ¹	DDOS
4.3.33.7	3.3	6.1 (all versions)
4.3.33.6	3.0.0.3 ²	6.0 (all versions)
4.3.33.5		5.7 (all versions)
4.3.33.4		5.6 (all versions)
4.3.33.3		5.5 (all versions) ²
4.3.33.2		
4.3.33.1		
4.3.33.0		
4.3.32.1		
4.3.32.0		
4.3.31.0		

Greenplum Database	Data Domain Boost ¹	DDOS
4.3.30.4 4.3.30.3 4.3.30.0		
4.3.29.0 4.3.28.0 4.3.27.0 4.3.26.0		
4.3.25.1 4.3.25.0		
4.3.24.0 4.3.23.0 4.3.22.0 4.3.21.0 4.3.20.0 4.3.19.0 4.3.18.0		
4.3.17.1 4.3.17.0	3.3 3.0.0.3 ²	6.0 (all versions) 5.7 (all versions) 5.6 (all versions) 5.5 (all versions) ²
4.3.16.1 4.3.16.0		
4.3.15.0		
4.3.14.1 4.3.14.0		
4.3.13.0		
4.3.12.0	3.0.0.3 ²	5.7 (all versions) 5.6 (all versions) 5.5 (all versions) ²
4.3.11.3 4.3.11.2 4.3.11.1		
4.3.10.0		
4.3.9.1 4.3.9.0		

Note: In addition to the DDOS versions listed in the previous table, Greenplum Database 4.3.9.0 and later supports all minor patch releases (fourth digit releases) later than the certified version.

¹The Greenplum Database utilities `gpbackup` and `gprestore` support Data Domain DD Boost File System Plugin (BoostFS) v1.1 with DDOS 6.0 or greater.

The `gpbackup` and `gprestore` utilities support using Dell EMC Data Domain Boost software with the DD Boost Storage Plugin.

²Support for Data Domain Boost 3.0.0.3 and DDOS 5.5 is deprecated. The DELL EMC end of Primary Support date is December 31, 2017.

Greenplum Database 4.3.33.5 supports Veritas NetBackup:

- NetBackup Master Server software.
 - NetBackup Master Server Version 7.7 and NetBackup Media Server Version 7.7
 - NetBackup Master Server Version 7.6 and NetBackup Media Server Version 7.6
 - NetBackup Master Server Version 7.5 and NetBackup Media Server Version 7.5
- NetBackup Client version: 7.1, 7.5, or 7.6.

Note: For NetBackup version 7.5 or 7.6, the client version that is installed and configured on the Greenplum Database hosts must match the NetBackup Server version that stores the Greenplum Database backup.

For NetBackup Client version 7.1, Greenplum Database supports only NetBackup Server Version 7.5.

Greenplum Database uses the NetBackup API (XBSA) to communicate with the NetBackup. Greenplum Database uses SDK version XBSA 1.1.0.

Greenplum Database support for NetBackup Client version 7.1 is deprecated. The NetBackup SDK library files for NetBackup version 7.1 will be removed from the Greenplum Database installation in a future release.

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.

Informatica PowerCenter 10.2 and 9.x are certified with Greenplum Database 4.3.x, all versions.

Note: Greenplum Database 4.3.33.7 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

Important: When data loss is not acceptable for a Pivotal Greenplum Database cluster, master and segment mirroring must be enabled in order for the cluster to be supported by Pivotal. Without mirroring, system and data availability is not guaranteed, Pivotal will make best efforts to restore a cluster in this case. For information about master and segment mirroring, see *About Redundancy and Failover* in the *Greenplum Database Administrator Guide*.

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), Microsoft Azure, and Google Cloud Compute (GCP).

- **AWS** - For production workloads, r4.8xlarge and r4.16xlarge instance types with four 12TB ST1 EBS volumes for each segment host are supported.

Greenplum Database is supported on 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 instance store storage, the storage should be RAID of Amazon volumes.

EBS storage is more reliable and provides more features than ephemeral storage but if ephemeral storage is desired, d2.8xlarge is supported for production workloads. With d2.8xlarge and ephemeral storage, use four RAID 0 volumes. Amazon has no provisions to replace a bad drive. If a disk failure occurs, the node with the bad disk must be replaced.

Pivotal recommends using an Auto Scaling Group (ASG) to provision nodes in AWS. An ASG automatically replaces bad nodes and further automation can be added to automatically recover the Greenplum processes on the new nodes.

Deployment should be in a Placement Group within a single Availability Zone, and since Amazon recommends using the same instance type in a Placement Group, use a single instance type for all nodes, including the masters.

- **Azure** - For production workloads, Pivotal recommends configuring Standard_H8 instance type with 4 2TB disks and 2 segments per host, and recommend using 8 2TB disks and 4 segments per host with Standard_H16 instance type. Standard_H16 uses 8 2TB disks and 4 segments per host. This means software RAID 0 is required so that the number of volumes do not exceed the number of segments.

For Azure deployments, you must also configure the Greenplum Database system to not use port 65330. Add the following line to the `sysctl.conf` file on all Greenplum Database hosts.

```
net.ipv4.ip_local_reserved_ports=65330
```

- **GCP** - For all workloads, n1-standard-8 and n1-highmem-8 are supported which are relatively small instance types. This is because of the disk performance in GCP forces the configuration to have just 2 segments per host but with many hosts to scale. Use pd-standard disks and the size of the disk is recommended to be 6 TB. For performance perspective, use a factor of 8 when determining how many nodes to deploy in GCP, so a 16 segment host cluster in AWS would require 128 nodes in GCP.
- 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 `gpinitssystem` to fail with semaphore errors. To avoid this issue, see "Setting the Greenplum Recommended OS Parameters" in the *Greenplum Database Installation Guide*.

Downloading Greenplum Database

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

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

Beta Features

Because Pivotal Greenplum Database is based on the open source *Greenplum Database project* code, it includes Beta features to allow interested developers to experiment with their use on development systems. Feedback will help drive development of these features, and they may become supported in future versions of the product.

Warning: Beta features are not supported for production deployments.

Greenplum Database 4.3.33.7 includes this Beta feature:

- Storage plugin framework API. Partners, customers, and OSS developers can develop plugins to use in conjunction with `gpbackup` and `gprestore`.

For information about the storage plugin API, see *Backup/Restore Storage Plugin API* in the *Pivotal Greenplum Database Documentation*.

Known Issues in Greenplum Database 4.3.33.7

This section lists the known issues in Greenplum Database 4.3.33.7. A workaround is provided where applicable.

For known issues discovered in previous 4.3.x releases, see the release notes available from the Pivotal Greenplum page on *Pivotal Network* or on the Pivotal Greenplum Database documentation site at *Release Notes*. 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 Dell EMC *Support Zone*

Table 3: All Known Issues in 4.3.33.7

Issue	Category	Description
30537	Postgres Planner	<p>The Postgres Planner generates a very large query plan that causes out of memory issues for the following type of CTE (common table expression) query: the <code>WITH</code> clause of the CTE contains a partitioned table with a large number partitions, and the <code>WITH</code> reference is used in a subquery that joins another partitioned table.</p> <p>Workaround: If possible, use the GPORCA query optimizer. With the server configuration parameter <code>optimizer=on</code>, Greenplum Database attempts to use GPORCA for query planning and optimization when possible and falls back to the Postgres Planner when GPORCA cannot be used. Also, the specified type of query might require a long time to complete.</p>

Issue	Category	Description
29631	gpbackup	On Greenplum Database 4.3 systems, the <code>gpbackup</code> utility sets the <code>enable_mergejoins</code> server configuration utility to <code>off</code> for the <code>gpbackup</code> session. This can cause catalog queries to fail during the backup session. This issue is resolved in Greenplum Backup and Restore v1.15 or later. See the Greenplum Backup and Restore Release Notes .
30204	gpcheck	The <code>gpcheck</code> utility reports errors for the <code>sysctl.kernel.sem</code> and <code>net.ipv4.ip_local_port_range</code> parameters when the parameters are set to the values recommended in the documentation. The documentation is more readily updated than the <code>gpcheck</code> software, so you can assume that the latest version of the <i>Greenplum Database Installation Guide</i> contains the most up-to-date recommendations for operation system parameters. If your settings match the most recent documented recommendations, you should ignore errors reported by <code>gpcheck</code> .
30245	Resource Management	Resource queue-related statistics may be inaccurate in certain cases. Pivotal recommends that you migrate to Greenplum Database 5.x and use the resource group resource management scheme that is available in that release.
168159520	Query Planner	A query specifying a <code>GROUP BY GROUPING SETS (...)</code> clause that includes duplicate groups may produce an incorrect result when the Greenplum Database server configuration parameter <code>gp_eager_one_phase_agg</code> is set to <code>true</code> . This issue is resolved in Greenplum Database 6.x.
169052763	gprestore	You can create a full backup of a database with <code>gpbackup</code> using the <code>--with-stats</code> option to back up table statistics. However, when you try to restore only some of the tables and the statistics for the tables using <code>gprestore</code> with a table filter option and the <code>--with-stats</code> option, <code>gprestore</code> attempts to restore all the table statistics from the backup, not just the statistics for the tables being restored. When restoring all the table statistics, if a table is not in the target database, <code>gprestore</code> returns an error. If a table exists in the database, <code>gprestore</code> replaces the existing table statistics.
30029	metrics_collector	A PL/pgSQL function with a <code>RAISE EXCEPTION</code> message longer than 1024 bytes can cause an overflow in the <code>metrics_collector</code> extension, leading to a segmentation fault on the Greenplum master. Workarounds: <ul style="list-style-type: none"> • Shorten <code>RAISE EXCEPTION</code> messages to under 1024 bytes. • Set the <code>gpcc.enable_send_query_info</code> server configuration parameter to <code>off</code> before running queries that have long <code>RAISE EXCEPTION</code> messages. <pre>SET gpcc.enable_send_query_info="off";</pre> <p>This parameter can be set in the session, and will prevent the <code>metrics_collector</code> extension from calling the code that produces the error.</p> <p>This issue is fixed in Greenplum Database version 5.21.1.</p>

Issue	Category	Description
29861	gpstart	<p>If a segment's data directory becomes inaccessible or the contents of the data directory are deleted, the segment goes offline and is marked as down in <code>gp_segment_configuration</code>. However, if you have temporary or transaction files in a separate filesystem or if you stop and start the database, <code>gpstart</code> can fail with the error:</p> <pre data-bbox="657 401 1380 598">20190404:17:26:44:025089 gpstart:mdw:gpadmin- [ERROR]: -Multiple OIDs found in flat files 20190404:17:26:44:025089 gpstart:mdw:gpadmin-[ERROR]:-Filespaces are inconsistent. Abort Greenplum Database start.</pre> <p>The error indicates that two files are missing from a segment directory, <code>gp_temporary_files_filespace</code> and <code>gp_transaction_files_filespace</code>. <code>gpstart</code> reports the error even if the segments have already been marked down.</p> <p>Workaround: To resolve this problem you must manually create the missing files in the segment directory using the segment's corresponding dbid, transaction file segment directory, and filesystem oid. See the detailed instructions for creating these files in Article Number: 6784 gpstart fails with error "Multiple OIDs found in flat files" on the Pivotal Support site.</p>
29124	DDL	<p>The <code>CREATE TABLE AS</code> command can create a table that is incorrectly defined with a distribution policy that contains duplicate columns as distribution keys. This issue has been resolved in Greenplum Database 5.10.0.</p> <p>This issue causes the <code>gprestore</code> utility to fail when the utility attempts to restore a table from a backup when the table is incorrectly defined with duplicate columns as distribution keys. See known issue 29395.</p>
29139	DML	<p>In some cases for an append-optimized partitioned table, Greenplum Database acquires a <code>ROW EXCLUSIVE</code> lock on all leaf partitions of the table when inserting data directly into one of the leaf partitions of the table. The locks are acquired the first time Greenplum Database performs validation on the leaf partitions. When inserting data into one leaf partition, the locks are not acquired on the other leaf partitions as long as the validation information remains in memory.</p> <p>The issue does not occur for heap-storage partitioned tables.</p>

Issue	Category	Description
29523	gp toolkit	<p>An upgrade between minor releases does not update the <code>template0</code> database, and in some cases, using these views in the <code>gp_toolkit</code> schema might cause issues if you create a database using <code>template0</code> as the template database after you upgrade to Greenplum Database 5.11.0 or later.</p> <ul style="list-style-type: none"> The <code>gp_toolkit.gp_bloat_expected_pages</code> view might incorrectly report that a root partition table is bloated even though root partition tables do not contain data if you upgrade from Greenplum Database 4.3.29.0 or earlier. The <code>gp_toolkit.gp_bloat_diag</code> view might return an integer out of range error in some cases if you upgrade from Greenplum Database 4.3.19.0 or earlier. <p>For example, the issues might occur if you upgrade a Greenplum Database system from 4.3.19.0 or an earlier 5.x release and then run a <code>gprestore</code> operation with the <code>--redirect-db</code> option to create a new database. The utility creates a new database using <code>template0</code> as the template database.</p> <p>Workaround: You can update the views in the <code>gp_toolkit</code> schema in the new database. For information about checking and updating <code>gp_toolkit</code>, see Update for gp_toolkit.gp_bloat_expected_pages Issue and Update for gp_toolkit.gp_bloat_diag Issue.</p>
29485	Catalog and Metadata	<p>When a session creates temporary objects in a database, Greenplum Database might not drop temporary objects when the session ends if the session terminates abnormally or is terminated from an administrator command.</p>
29496	gpconfig	<p>For a small number of server configuration parameters such as <code>log_min_messages</code>, the command <code>gpconfig -s <config_param></code> does not display the correct value of the parameter for the segment hosts when the value of the parameter on master is different than the value on the segments.</p> <p>For parameters with the set classification <code>master</code>, the utility displays the value set on the master for both master and segments (for information about set classifications, see Setting Parameters in the <i>Pivotal Greenplum Database Documentation</i>). For those parameters, the value on the master is passed as part of queries to segment instances. The SQL query that <code>gpconfig</code> runs to display the master and segment parameter values returns the master host value that is passed to the segment as part of the query.</p> <p>For a few parameters such as <code>log_min_messages</code>, segment instances use the segment host value specified in the <code>postgresql.conf</code> file at start up. The segment value can be overridden for the scope of a query.</p> <p>Workaround: To display the parameter value specified in the <code>postgresql.conf</code> file on the master host and segment hosts, you can specify the <code>gpconfig</code> option <code>--file</code>.</p>

Issue	Category	Description
29395	DDL	The <code>gpdbrestore</code> or <code>gprestore</code> utility fails when the utility attempts to restore a table from a backup and the table is incorrectly defined with duplicate columns as distribution keys. The issue is caused when the <code>gpccrondump</code> or <code>gpbackup</code> utility backed up a table that is incorrectly defined. The <code>CREATE TABLE AS</code> command could create a table that is incorrectly defined with a distribution policy that contains duplicate columns as distribution keys.
29351	gptransfer	The <code>gptransfer</code> utility can copy a data row with a maximum length of 256 MB.
30180	Locking, Signals, Processes	The <code>pg_cancel_backend()</code> and <code>pg_terminate_backend()</code> functions might leave some orphan processes when they are used to cancel a running <code>VACUUM</code> command. Workaround: You can stop the orphan processes by restarting the Greenplum Database system.

Issue	Category	Description
164671144	gpssh-exkeys, gpsegininstall	<p>The <code>gpssh-exkeys</code> utility uses the Paramiko SSH library for Python, which has a dependency on the Python Cryptography Toolkit (PyCrypto) library. The following security vulnerabilities have been identified in some versions of PyCrypto.</p> <ul style="list-style-type: none"> • https://nvd.nist.gov/vuln/detail/CVE-2018-6594 - <code>lib/Crypto/PublicKey/EIGamal.py</code> in PyCrypto through 2.6.1 generates weak EIGamal key parameters, which allows attackers to obtain sensitive information by reading ciphertext data (i.e., it does not have semantic security in the face of a ciphertext-only attack). The Decisional Diffie-Hellman (DDH) assumption does not hold for PyCrypto's EIGamal implementation. <p>Paramiko does not import this algorithm, so Greenplum Database is unaffected by this vulnerability.</p> <ul style="list-style-type: none"> • https://nvd.nist.gov/vuln/detail/CVE-2013-7459 - Heap-based buffer overflow in the <code>ALGnew</code> function in <code>block_template.c</code> in Python Cryptography Toolkit (aka <code>pycrypto</code>) allows remote attackers to execute arbitrary code as demonstrated by a crafted <code>iv</code> parameter to <code>cryptmsg.py</code>. <p>This bug was introduced in PyCrypto 2.6. Greenplum Database has PyCrypto 2.0.1, and is unaffected by the vulnerability.</p> <ul style="list-style-type: none"> • https://nvd.nist.gov/vuln/detail/CVE-2013-1445 - The <code>Crypto.Random.atfork</code> function in PyCrypto before 2.6.1 does not properly reseed the pseudo-random number generator (PRNG) before allowing a child process to access it, which makes it easier for context-dependent attackers to obtain sensitive information by leveraging a race condition in which a child process is created and accesses the PRNG within the same rate-limit period as another process. <p>Paramiko version 1.7.6-9, used in Greenplum Database 4.x and 5.x, introduced a workaround to this bug.</p> <ul style="list-style-type: none"> • https://nvd.nist.gov/vuln/detail/CVE-2012-2417 - PyCrypto before 2.6 does not produce appropriate prime numbers when using an EIGamal scheme to generate a key, which reduces the signature space or public key space and makes it easier for attackers to conduct brute force attacks to obtain the private key. <p>Paramiko does not import this algorithm, so Greenplum Database is unaffected by the vulnerability.</p> <p>Through testing and investigation, Pivotal has determined that these vulnerabilities do not affect Greenplum Database, and no actions are required for existing Greenplum Database 4.3 or 5.x releases. However, there may be additional unidentified vulnerabilities in the PyCrypto library, and users who install a later version of PyCrypto could be exposed to other vulnerabilities.</p> <p>The PyCrypto library is removed from Greenplum Database 6.0. The <code>gpsegininstall</code> utility is deprecated and has been removed in Greenplum Database 6.0. The dependency on Paramiko and PyCrypto is removed from <code>gpssh-exkeys</code> in Greenplum Database 6.0.</p> <p>Workaround: Administrators can set up passwordless SSH between hosts in the Greenplum Database cluster without using the <code>gpssh-</code></p>

Issue	Category	Description
		<p>exkeys utility. This must be done before initializing the Greenplum Database system.</p> <ol style="list-style-type: none"> 1. Create an SSH key for the <code>gpadmin</code> user on each host. For example, log in to each host as <code>gpadmin</code> and use the <code>ssh-keygen</code> command to generate an SSH key. Do not enter a passphrase. 2. Add the public key for each host in the cluster to every other host in the cluster. For example, use the <code>ssh-copy-id</code> command from each host to copy the public key to every other host in the cluster. <p>When adding new hosts to the Greenplum Database system, create a new SSH key for each new host and exchange keys between the existing hosts and new hosts.</p>
165434975	search_path	<p>An identified PostgreSQL security vulnerability (https://nvd.nist.gov/vuln/detail/CVE-2018-1058) also exists in Greenplum Database. The problem centers around the default <code>public</code> schema and how Greenplum Database uses the <code>search_path</code> setting. The ability to create objects with the same names in different schemas, combined with how Greenplum Database searches for objects within schemas, presents an opportunity for a user to modify the behavior of a query for other users. For example, a malicious user could insert a trojan-horse function that, when executed by a superuser, grants escalated privileges to the malicious user.</p> <p>There are methods to protect from this vulnerability. See <i>A Guide to CVE-2018-1058: Protect Your Search Path</i> on the PostgreSQL wiki for a full explanation of the vulnerability and the steps you can take to protect your data.</p>
151135629	COPY	<p>When the <code>ON SEGMENT</code> clause is specified, the <code>COPY</code> command does not support specifying a <code>SELECT</code> statement in the <code>COPY TO</code> command. However, this command completes successfully, but the files are not created on the segment hosts.</p> <pre data-bbox="657 1291 1218 1354">COPY (SELECT * FROM testtbl) TO '/tmp/mytst<SEGID>' ON SEGMENT</pre>
150625402	Session Management	<p>When the server configuration parameter <code>gp_strict_xml_parse</code> is set for a session and the session is idle for longer than <code>gp_vmem_idle_resource_timeout</code>, the value of <code>gp_strict_xml_parse</code> changes back to the value set for the system (or the database if the parameter is set for the database).</p>
29064	Storage: DDL	<p>The <code>money</code> datatype accepts out-of-range values as negative values, and no error message is displayed.</p> <p>Workaround: Use only in-range values for the <code>money</code> datatype (32-bit for Greenplum Database 4.x, or 64-bit for Greenplum Database 5.x). Or, use an alternative datatype such as <code>numeric</code> or <code>decimal</code>.</p>

Issue	Category	Description
28947	Access Methods	<p>A deadlock might occur on an append-optimized columnar table when a <code>VACUUM</code> operation and an <code>INSERT</code> operation are performed concurrently on the table.</p> <p>Workaround: If a deadlock condition occurs, terminate the <code>INSERT</code> operation to break the deadlock. To eliminate the possibility of encountering this issue, avoid concurrent <code>VACUUM</code> and <code>INSERT</code> operations.</p>
26675	gpcrondump	<p>During the transition from Daylight Saving Time to Standard Time, this sequence of events which might cause a <code>gpcrondump</code> backup operation to fail.</p> <p>If an initial backup is taken between 1:00AM and 2:00AM Daylight Saving Time, and a second backup is taken between 1:00AM and 2:00AM Standard Time, the second backup might fail if the first backup has a timestamp newer than the second.</p> <p>Pivotal recommends performing only a single backup between the hours of 1:00AM and 2:00AM on the days when the time changes:</p> <ul style="list-style-type: none"> • November 1, 2020 • November 7, 2021 • November 6, 2022 <p>If the failure scenario is encountered, it can be remedied by restarting the backup operation after 2:00AM Standard Time.</p>
146542311	gpload	<p>When running the Greenplum Database utility <code>gpload</code> on AIX systems, the utility returns an error if the YAML control file for utility contains a line that specifies the <code>\</code> (backslash) as the escape character, <code>ESCAPE: '\'</code>. The error states that the <code>\</code> at the end of a string could not be decoded.</p> <p>Workaround: To avoid the error, remove the line from the file, or specify the line without a character, <code>ESCAPE: .</code> The <code>\</code> character is the default escape character. The line is not required in the file.</p>
142743943	S3 External Tables	<p>The <code>s3</code> protocol might not handle the header row in data files properly in this situation:</p> <ul style="list-style-type: none"> • A readable external table is defined with the <code>s3</code> protocol and the <code>HEADER</code> option. • The external table has been exchanged to be a leaf child table of a partitioned table. <p>Queries against the partitioned table might return an error.</p>
26591	Query Execution	<p>For the Greenplum Database function <code>get_ao_compression_ratio()</code>, specifying a <code>null</code> value or the name of table that contains no rows causes a Greenplum Database PANIC.</p> <p>Workaround: Specify a non-null value or a table that contains rows.</p>

Issue	Category	Description
115746399	Operating System	<p>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.</p> <p>Workaround: RHEL 7.3 and CentOS 7.3 resolves the issue.</p>
26626	GPHDFS	<p>For Greenplum Database external tables, the <code>gphdfs</code> protocol supports Avro files that contain a single top-level schema. Avro files that contain multiple top-level schemas are not supported.</p>
25584	Query Execution	<p>In some situations, a running Greenplum Database query cannot be terminated with the functions <code>pg_cancel_backend</code> or <code>pg_terminate_backend</code>.</p> <p>The functions could not terminate the query due to a blocking <code>fopen</code> of a FIFO file for write.</p>
26249	GPHDFS	<p>When reading data from an Avro file, the <code>gphdfs</code> protocol does not support the double quote character (") within string data. The <code>gphdfs</code> protocol uses the double quote as the column delimiter.</p> <p>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.</p>
26292	Loaders: gpload	<p>The Greenplum Database <code>gpload</code> utility fails on MacOS X El Capitan. The utility script is included with the Greenplum Database Load Tools installer package for Apple OS X <code>greenplum-loaders-version-OSX-i386.bin</code>.</p> <p>Workaround: Run the python script <code>gpload.py</code> directly. For example, <code>python</code> command displays the <code>gpload</code> help information on the command line.</p> <pre>python gpload.py -h</pre>
26128	Loaders: gpload	<p>When the YAML control file for the Greenplum Database <code>gpload</code> utility specifies the key <code>LOG_ERRORS: true</code> without the key <code>REUSE TABLES: true</code>, the <code>gpload</code> operation returns only summary information about formatting errors. The formatting errors are deleted from Greenplum Database error logs. When <code>REUSE TABLES: true</code> is not specified, the temporary tables that are used by <code>gpload</code> are dropped after the <code>gpload</code> operation, and the formatting errors are also deleted from the Greenplum Database error logs.</p> <p>Workaround: Specify the YAML control file key <code>REUSE TABLES: true</code> 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 <code>gp_truncate_error_log()</code>.</p> <p>For information about the <code>gpload</code> utility, see the <i>Greenplum Database Utility Guide</i>.</p>

Issue	Category	Description
25934 25936	Query Optimizer Query Planner	<p>For queries that compare data from columns of different character types, for example a join comparing a columns of data types <code>CHAR(n)</code> and <code>VARCHAR(m)</code>, the returned results might not be as expected depending the padding added to the data (space characters added after the last non-space character).</p> <p>For example, this comparison returns false.</p> <pre>select 'A '::char(2) = 'A '::text ;</pre> <p>This comparison returns true.</p> <pre>select 'A '::char(2) = 'A '::varchar(5) ;</pre> <p>Workaround: Pivotal recommends specifying character column types to be of data type <code>VARCHAR</code> or <code>TEXT</code> so that comparisons include padding added to the data.</p> <p>For information about how the character data types <code>CHAR</code>, <code>VARCHAR</code>, and <code>TEXT</code> handle padding added to the data see the <code>CREATE TABLE</code> command in the <i>Greenplum Database Reference Guide</i>.</p>
25737	Catalog and Metadata	Greenplum Database does not support the <code>FILTER</code> clause within aggregate expressions.
25754	Management Scripts: expansion	<p>The Greenplum Database <code>gpexpand</code> 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.</p> <p>Workaround: Create the input file manually.</p>
25833	Management Scripts: gpexpand	The Greenplum Database utility <code>gpexpand</code> 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.
15835	DDL and Utility Statements	<p>For multi-level partitioned tables that have these characteristics:</p> <ul style="list-style-type: none"> • The top level partition is partitioned by range. • The lowest level partition (the leaf child partitions) are partitioned by list. <p>Splitting a subpartition with the <code>ALTER TABLE SPLIT PARTITION</code> command returns an error and rolls back the transaction.</p>
12019	Management Scripts: checkperf	<p>When the Greenplum Database <code>gpcheckperf</code> utility is run with the option <code>-f host_file</code> and the host that is running <code>gpcheckperf</code> is listed in <code>host_file</code>, processes that were started <code>gpcheckperf</code> might not be cleaned up after the utility completes.</p> <p>Workaround: Manually stop the processes that were started by <code>gpcheckperf</code>.</p>
24870	Query Optimizer	GPORCA might terminate all sessions if a query attempts to cast to a timestamp a date with year greater than 200,000.

Issue	Category	Description
23571	Query Optimizer	For queries that contain inequality conditions such as != , < and , >, GPORCA 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	GPORCA does not support GiST indexes.
20030	Query Optimizer	GPORCA does not support partition elimination when the query contains functions that are applied to the partition key.
20360	Query Execution	GPORCA 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 GPORCA does not consider indices when querying parts/child tables of partitioned tables directly.
25326	Interconnect	Setting the Greenplum Database server configuration parameter <code>log_hostname</code> to <code>on</code> on Greenplum Database segment hosts causes an Interconnect Error that states that the listeneraddress name or service not known. The parameter should be set to <code>on</code> only on the Greenplum Database master.
25280	Management Scripts: <code>gpstart/gpstop</code>	The Greenplum Database utility <code>gpstop</code> , the utility returns an error if it is run and the system environment variable <code>LANG</code> is set, for example, <code>export LANG=ja_JP.UTF-8</code> . Workaround: Unset the environment variable <code>LANG</code> before running the <code>gpstop</code> utility. For example: <pre>\$ unset LANG</pre>
25246	Management Scripts: <code>gpconfig</code>	When you set the server configuration parameters <code>gp_email_to</code> and <code>gp_email_from</code> with the Greenplum Database utility <code>gpconfig</code> , the utility removes the single quotes from the values. <pre>\$ gpconfig -c gp_email_to -v 'test@example.com'</pre> The improperly set parameter causes Greenplum Database to fail when it is restarted. Workaround: Enclose the value for <code>gp_email_to</code> or <code>gp_email_from</code> with double quotes. <pre>\$ gpconfig -c gp_email_to -v "'test@example.com'"</pre>
25168	Locking, Signals, Processes	When the server configuration parameter <code>client_min_messages</code> is set to either <code>PANIC</code> or <code>FATAL</code> and a <code>PANIC</code> or <code>FATAL</code> level message is encountered, Greenplum Database hangs. The <code>client_min_messages</code> parameter should not be set a value higher than <code>ERROR</code> .

Issue	Category	Description
24588	Management Scripts: gpconfig	The Greenplum Database <code>gpconfig</code> utility does not display the correct information for the server configuration parameter <code>gp_enable_gpperfmon</code> . The parameter displays the state of the Greenplum Command Center data collection agents (<code>gpperfmon</code>). Workaround: The SQL command <code>SHOW</code> displays the correct <code>gp_enable_gpperfmon</code> value.
24031	gphdfs	If a readable external table is created with <code>FORMAT 'CSV'</code> and uses the gphdfs protocol, reading a record fails if the record spans multiple lines and the record is stored in multiple HDFS blocks. 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: <pre>source \$GPHOME/greenplum_path.sh</pre> 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.
23366	Resource Management	In Greenplum Database 4.2.7.0 and later, the priority of some running queries, cannot be dynamically adjusted with the <code>gp_adjust_priority()</code> function. The attempt to execute this request might silently fail. The return value of the <code>gp_adjust_priority()</code> call indicates success or failure. If 1 is returned, the request was not successfully executed. If a number greater than 1 is returned, the request was successful. If the request fails, the priority of all running queries are unchanged, they remain as they were before the <code>gp_adjust_priority()</code> call.
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 <code>gpcrondump</code> , cannot be restored to a Greenplum Database 4.2.x system with the <code>psql</code> utility or the corresponding restore utility, for example <code>gpdbrestore</code> .
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 <code>\$GPHOME/lib/hadoop/hadoop_env.sh</code> must be to be modified so that the paths point to the appropriate JAR files.

Issue	Category	Description
20453	Query Planner	<p>For SQL queries of either of the following forms:</p> <pre>SELECT columns FROM table WHERE table.column NOT IN subquery; SELECT columns FROM table WHERE table.column = ALL subquery;</pre> <p>tuples that satisfy both of the following conditions are not included in the result set:</p> <ul style="list-style-type: none"> <code>table.column</code> is NULL. <code>subquery</code> returns the empty result.
21838	Backup and Restore	<p>When restoring sets of tables with the Greenplum Database utility <code>gpdrestore</code>, the table schemas must be defined in the database. If a table's schema is not defined in the database, the table is not restored. When performing a full restore, the database schemas are created when the tables are restored.</p> <p>Workaround: Before restoring a set of tables, create the schemas for the tables in the database.</p>
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 <code>!</code> , <code>\$</code> , <code>#</code> , and <code>@</code> cannot be used in the password for the Data Domain Boost user when specifying the Data Domain Boost credentials with the <code>gpcrondump</code> options <code>--ddbboost-host</code> and <code>--ddbboost-user</code> .
18247	DDL and Utility Statements	<p><code>TRUNCATE</code> command does not remove rows from a sub-table of a partitioned table. If you specify a sub-table of a partitioned table with the <code>TRUNCATE</code> command, the command does not remove rows from the sub-table and its child tables.</p> <p>Workaround: Use the <code>ALTER TABLE</code> command with the <code>TRUNCATE PARTITION</code> clause to remove rows from the sub-table and its child tables.</p>
19705	Loaders: gpload	<p><code>gpload</code> fails on Windows XP with Python 2.6.</p> <p>Workaround: Install Python 2.5 on the system where <code>gpload</code> is installed.</p>

Issue	Category	Description
19493 19464 19426	Backup and Restore	<p>The <code>gpccrondump</code> and <code>gpdbrestore</code> utilities do not handle errors returned by DD Boost or Data Domain correctly.</p> <p>These are two examples:</p> <ul style="list-style-type: none"> If invalid Data Domain credentials are specified when setting the Data Domain Boost credentials with the <code>gpccrondump</code> utility, the error message does not indicate that invalid credentials were specified. Restoring a Greenplum database from a Data Domain system with <code>gpdbrestore</code> and the <code>--ddboost</code> option indicates success even though segment failures occurred during the restore. <p>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.</p>
15692 17192	Backup and Restore	<p>Greenplum Database's implementation of RSA lock box for Data Domain Boost changes backup and restore requirements for customers running SuSE.</p> <p>The current implementation of the RSA lock box for Data Domain Boost login credential encryption only supports customers running on Red Hat Enterprise Linux.</p> <p>Workaround: If you run Greenplum Database on SuSE, use NFS as your backup solution. See the <i>Greenplum Database Administrator Guide</i> for information on setting up a NFS backup.</p>
18850	Backup and Restore	<p>Data Domain Boost credentials cannot be set up in some environments due to the absence of certain libraries (for example, <code>libstdc++</code>) expected to reside on the platform.</p> <p>Workaround: Install the missing libraries manually on the system.</p>
18851	Backup and Restore	<p>When restoring table data to an existing table with the Greenplum Database utility <code>gpdbrestore</code>, the utility assumes that the database table definition is the same as the table that was backed up. The utility does not check the table definition.</p> <p>For example, the distribution key for a table is changed after it is backed up. You back up the table, change the table distribution key, truncate the table, and then restore the table data from the backup. Subsequent queries against the table might return unexpected and incorrect results.</p> <p>Workaround: For the previous example, run the <code>ALTER TABLE</code> command with the <code>REORGANIZE=true</code> clause to redistribute the table data among the Greenplum Database segments. See <code>ALTER TABLE</code> in the <i>Greenplum Database Reference Guide</i>.</p>
18713	Catalog and Metadata	<p>Drop language <code>plpgsql</code> cascade results in a loss of <code>gp_toolkit</code> functionality.</p> <p>Workaround: Reinstall <code>gp_toolkit</code>.</p>

Issue	Category	Description
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.
17513	Management Scripts Suite	Running more than one gpfilespace command concurrently with itself to move either temporary files (--movetempfilespace) or transaction files (--movetransfilespace) to a new filespace can in some circumstances cause OID inconsistencies. Workaround: Do not run more than one gpfilespace command concurrently with itself. If an OID inconsistency is introduced gpfilespace --movetempfilespace or gpfilespace --movetransfilespace can be used to revert to the default filespace.

Issue	Category	Description
17780	DDL/DML: Partitioning	<p><code>ALTER TABLE ADD PARTITION</code> inheritance issue</p> <p>When performing an <code>ALTER TABLE ADD PARTITION</code> operation, the resulting parts may not correctly inherit the storage properties of the parent table in cases such as adding a default partition or more complex subpartitioning. This issue can be avoided by explicitly dictating the storage properties during the <code>ADD PARTITION</code> invocation. For leaf partitions that are already afflicted, the issue can be rectified through use of <code>EXCHANGE PARTITION</code>.</p>
17795	Management Scripts Suite	<p>Under some circumstances, <code>gppkg</code> on SuSE is unable to correctly interpret error messages returned by <code>rpm</code>.</p> <p>On SuSE, <code>gppkg</code> is unable to operate correctly under circumstances that require a non-trivial interpretation of underlying <code>rpm</code> commands. This includes scenarios that result from overlapping packages, partial installs, and partial uninstalls.</p>
17604	Security	<p>A Red Hat Enterprise Linux (RHEL) 6.x security configuration file limits the number of processes that can run on <code>gpadmin</code>.</p> <p>RHEL 6.x contains a security file (<code>/etc/security/limits.d/90-nproc.conf</code>) that limits available processes running on <code>gpadmin</code> to 1064.</p> <p>Workaround: Remove this file or increase the processes to 131072.</p>
17334	Management Scripts Suite	<p>You may see warning messages that interfere with the operation of management scripts when logging in.</p> <p>Greenplum recommends that you edit the <code>/etc/motd</code> file and add the warning message to it. This will send the messages to be redirected to <code>stdout</code> and not <code>stderr</code>. You must encode these warning messages in UTF-8 format.</p>
17221	Resource Management	Resource queue deadlocks may be encountered if a cursor is associated with a query invoking a function within another function.
17113	Management Scripts Suite	<p>Filespaces are inconsistent when the Greenplum database is down.</p> <p>Filespaces become inconsistent in case of a network failure. Greenplum recommends that processes such as moving a filespace be done in an environment with an uninterrupted power supply.</p>
17189	Loaders: <code>gpfdist</code>	<p><code>gpfdist</code> shows the error "Address already in use" after successfully binding to socket IPv6.</p> <p>Greenplum supports IPv4 and IPv6. However, <code>gpfdist</code> fails to bind to socket IPv4, and shows the message "Address already in use", but binds successfully to socket IPv6.</p>
16064	Backup and Restore	<p>Restoring a compressed dump with the <code>--ddboost</code> option displays incorrect dump parameter information.</p> <p>When using <code>gpdbrestore --ddboost</code> to restore a compressed dump, the restore parameters incorrectly show "Restore compressed dump = Off". This error occurs even if <code>gpdbrestore</code> passes the <code>--gp-c</code> option to use <code>gunzip</code> for in-line de-compression.</p>

Issue	Category	Description
15899	Backup and Restore	When running <code>gpdbrestore</code> with the list (<code>-L</code>) option, external tables do not appear; this has no functional impact on the restore job.
168142530	Backup and Restore	<p>Backups created on Greenplum Database versions before 4.3.33.0 or 5.1.19.0 may fail to restore to Greenplum Database versions 4.3.33.0 or 5.1.19.0 or later.</p> <p>In Greenplum Database 4.3.33.0 and 5.1.19.0, a check was introduced to ensure that the distribution key for a table is equal to the primary key or is a left-subset of the primary key. If you add a primary key to a table that contains no data, Greenplum Database automatically updates the distribution key to match the primary key.</p> <p>The index key for any unique index on a table must also match or be a left-subset of the distribution key.</p> <p>Earlier Greenplum Database versions did not enforce these policies.</p> <p>Restoring a table from an older backup that has a different distribution key causes errors because the backup data file on each segment contains data that was distributed using the original distribution key.</p> <p>Restoring a unique index with a key that does not match the distribution key will fail with an error when attempting to create the index.</p> <p>This issue affects the <code>gprestore</code>, <code>gpdbrestore</code>, and <code>pg_restore</code> utilities.</p>

Upgrading to Greenplum Database 4.3.33.7

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.33.7. 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 `gpinitssystem` 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 4.3.33.7, 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 the most recent version of MADlib. For MADlib support and upgrade information, refer to the *MADlib FAQ*. For information on installing the MADlib extension in Greenplum Database, see *Greenplum MADlib Extension for Analytics* in the *Greenplum Database Reference Guide*.

If the `pgcrypto` extension package version `pv1.2` or earlier is installed in your system, you must uninstall the `pgcrypto` extension and install `pgcrypto` package version `pv1.3`.

For information about supported versions of Greenplum Database extensions, see *Greenplum Database Extensions*.

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

- *Upgrading from 4.3.x to 4.3.33.7*
- *Upgrading from 4.3.x to 4.3.33.7 on Pivotal DCA Systems*
- *Upgrading from 4.2.x.x to 4.3.33.7*
- *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 4.3.33.7

An upgrade from 4.3.x to 4.3.33.7 involves stopping Greenplum Database, updating the Greenplum Database software binaries, upgrading and restarting Greenplum Database. If you are using Greenplum Database extension packages there are additional requirements. See *Prerequisites* in the previous section.

Important: If you are upgrading from Greenplum Database 4.3.x on a Pivotal DCA system, see *Upgrading from 4.3.x to 4.3.33.7 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.33.7 on non-DCA systems.

Note: If you have databases that were created with Greenplum Database 4.3.29.0 or an earlier 4.3.x release, upgrade the `gp_bloat_expected_pages` view in the `gp_toolkit` schema. For information about the issue and how check a database for the issue, see *Update for gp_toolkit.gp_bloat_expected_pages Issue*.

Note: If you are upgrading from Greenplum Database 4.3.27.0 or an earlier 4.3.x release and have configured PgBouncer in your Greenplum Database installation, you must migrate to the new PgBouncer when you upgrade Greenplum Database. Refer to *Migrating PgBouncer* for specific migration instructions.

Note: If you have databases that were created with Greenplum Database 4.3.19.0 or an earlier 4.3.x release, upgrade the `gp_bloat_diagfunction` and view in the `gp_toolkit` schema. For information about the issue and how check a database for the issue, see *Update for gp_toolkit.gp_bloat_diag Issue*.

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 4.3.33.7 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-4.3.33.7
```

5. If needed, update the `greenplum_path.sh` file for use with your specific installation. These are some examples.

- If Greenplum Database uses LDAP authentication, edit the `greenplum_path.sh` file to add the line:

```
export LDAPCONF=/etc/openldap/ldap.conf
```

- If Greenplum Database uses PL/Java, you might need to set or update the environment variables `JAVA_HOME` and `LD_LIBRARY_PATH` in `greenplum_path.sh`.

Note: When comparing the previous and new `greenplum_path.sh` files, be aware that installing some Greenplum Database extensions also updates the `greenplum_path.sh` file. The `greenplum_path.sh` from the previous release might contain updates that were the result of those extensions. See step 10 for installing Greenplum Database extensions.

6. 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-4.3.33.7/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-4.3.33.7 /usr/local/greenplum-db
```

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

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

8. Run the `gpsegininstall` utility to install the 4.3.33.7 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

9. 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.
10. 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.

Also copy any additional files that are used by the extensions (such as JAR files, shared object files, and libraries) from the previous version installation directory to the new version installation directory on the master and segment host systems.

11. If you are upgrading from Greenplum Database 4.3.27.0 or an earlier 4.3.x release and have configured PgBouncer in your Greenplum Database installation, you must migrate to the new PgBouncer when you upgrade Greenplum Database. Refer to *Migrating PgBouncer* for specific migration instructions.
12. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

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

13. 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.

14. 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.

15. 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 4.3.33.7 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.

After upgrading Greenplum Database, ensure features work as expected. For example, you should test that backup and restore perform as expected, and Greenplum Database features such as user-defined functions, and extensions such as MADlib and PostGIS perform as expected.

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 4.3.33.7 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=-h master_host | --host=master_host

Greenplum Database master hostname or IP address.

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

Greenplum Database master port.

-u user | --user=-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 4.3.33.7.

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 4.3.33.7 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 4.3.33.7 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.33.7 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 4.3.33.7 on DCA systems. This section is only for installing Greenplum Database 4.3.33.7 on DCA systems.

Note: If you have databases that were created with Greenplum Database 4.3.29.0 or an earlier 4.3.x release, upgrade the `gp_bloat_expected_pages` view in the `gp_toolkit`

schema. For information about the issue and how check a database for the issue, see [Update for gp_toolkit.gp_bloat_expected_pages Issue](#).

Note: If you are upgrading from Greenplum Database 4.3.27.0 or an earlier 4.3.x release and have configured PgBouncer in your Greenplum Database installation, you must migrate to the new PgBouncer when you upgrade Greenplum Database. Refer to [Migrating PgBouncer](#) for specific migration instructions.

Note: If you have databases that were created with Greenplum Database 4.3.19.0 or an earlier 4.3.x release, upgrade the `gp_bloat_diagfunction` and view in the `gp_toolkit` schema. For information about the issue and how check a database for the issue, see [Update for gp_toolkit.gp_bloat_diag Issue](#).

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 4.3.33.7 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 4.3.33.7 for Redhat Enterprise Linux 5.x.

```
# ./greenplum-db-appliance-4.3.33.7-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. If needed, update the `greenplum_path.sh` file on the system hosts for use with your specific installation. These are some examples.
 - If Greenplum Database uses LDAP authentication, edit the `greenplum_path.sh` file to add the line:

```
export LDAPCONF=/etc/openldap/ldap.conf
```

- If Greenplum Database uses PL/Java, you might need to set or update the environment variables `JAVA_HOME` and `LD_LIBRARY_PATH` in `greenplum_path.sh`.

Note: When comparing the previous and new `greenplum_path.sh` files, be aware that installing some Greenplum Database extensions also updates the `greenplum_path.sh` file. The `greenplum_path.sh` from the previous release might contain updates that were the result of those extensions. See step 7 for installing Greenplum Database extensions.

7. 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.

Also copy any additional files that are used by the extensions (such as JAR files, shared object files, and libraries) from the previous version installation directory to the new version installation directory on the master and segment host systems.

8. If you are upgrading from Greenplum Database 4.3.27.0 or an earlier 4.3.x release and have configured PgBouncer in your Greenplum Database installation, you must migrate to the new PgBouncer when you upgrade Greenplum Database. Refer to *Migrating PgBouncer* for specific migration instructions.
9. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

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

10. 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.

11. 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 4.3.33.7 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.

After upgrading Greenplum Database, ensure features work as expected. For example, you should test that backup and restore perform as expected, and Greenplum Database features such as user-defined functions, and extensions such as MADlib and PostGIS perform as expected.

Upgrading from 4.2.x.x to 4.3.33.7

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.33.7. For users running versions prior to 4.2.x.x of Greenplum Database, see the following:

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

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

Planning Your Upgrade

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

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

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

For example:

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

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

Important: If you intend to use an extension packages with Greenplum Database 4.3.5.0 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-4.3.33.7-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.33.7-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-4.3.33.7`), 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 4.3.33.7 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.33.7/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.33.7 binaries on all the segment hosts specified in the `hostfile`. For example:

```
$ gpsegininstall -f hostfile
```

Install the Greenplum Database 4.3 Software Binaries on DCA Systems

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

1. Download or copy the installer file to the Greenplum Database master host.
2. As root, run the Pivotal DCA installer for 4.3.33.7 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 4.3.33.7.

```
# ./greenplum-db-appliance-4.3.33.7-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 4.3.33.7 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.33.7/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.33.7/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.33.7 installation.

- a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.33.7 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.33.7 /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-4.3.33.7 /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-4.3.33.7 /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-4.3.33.7 /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-4.3.33.7
```

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-4.3.33.7`.

8. As `gpadmin`, run the 4.3.33.7 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-4.3.33.7
```

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-4.3.33.7`.

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 4.3.33.7 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 4.3.33.7 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 from the Pivotal Greenplum page on [Pivotal Network](#). For information about `gpfdist`, see the *Greenplum Database 4.3 Administrator Guide*.
3. If needed, update the `greenplum_path.sh` file on the system hosts for use with your specific installation. For example, if Greenplum Database uses PL/Java, you might need to set or update the environment variables `JAVA_HOME` and `LD_LIBRARY_PATH` in `greenplum_path.sh`.

Note: When comparing the previous and new `greenplum_path.sh` files, be aware that installing some Greenplum Database extensions also updates the `greenplum_path.sh` file. The `greenplum_path.sh` from the previous release might contain updates that were the result of those extensions.

4. 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.
5. 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.

Also copy any additional files that are used by the extensions (such as JAR files, shared object files, and libraries) from the previous version installation directory to the new version installation directory on the master and segment host systems.

6. 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 the Pivotal Greenplum page on [Pivotal Network](#).

7. (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
```

8. Ensure features work as expected. For example, you should test that backup and restore perform as expected, and Greenplum Database features such as user-defined functions, and extensions such as `MADlib` and `PostGIS` perform as expected.
9. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.33.7 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 4.3.33.7. Detailed upgrade instructions are provided in [Upgrading from 4.2.x.x to 4.3.33.7](#).

Pre-Upgrade Preparation (on your current system)	
* 4.2.x.x system is up and available	
<input type="checkbox"/>	Log in to your master host as the <code>gpadmin</code> user (your Greenplum superuser).
<input type="checkbox"/>	(Optional) Run <code>VACUUM</code> on all databases.

<input type="checkbox"/>	(Optional) Remove old server log files from <code>pg_log</code> in your master and segment data directories.
<input type="checkbox"/>	Check for and recover any failed segments (<code>gpstate</code> , <code>gprecoverseg</code>).
<input type="checkbox"/>	Copy or preserve any additional folders or files (such as backup folders).
<input type="checkbox"/>	Install the Greenplum Database 4.3 binaries on all Greenplum hosts.
<input type="checkbox"/>	Inform all database users of the upgrade and lockout time frame.
Upgrade Execution	
* The system will be locked down to all user activity during the upgrade process	
<input type="checkbox"/>	Backup your current databases.
<input type="checkbox"/>	Remove the standby master (<code>gpinitstandby -r</code>).
<input type="checkbox"/>	Do a clean shutdown of your current system (<code>gpstop</code>).
<input type="checkbox"/>	Update your environment to source the new Greenplum Database 4.3.x installation.
<input type="checkbox"/>	Run the upgrade utility (<code>gpmigrator_mirror</code> if you have mirrors, <code>gpmigrator</code> if you do not).
<input type="checkbox"/>	After the upgrade process finishes successfully, your 4.3.x system will be up and running.
Post-Upgrade (on your 4.3 system)	
* The 4.3.x.x system is up	
<input type="checkbox"/>	Reinitialize your standby master host (<code>gpinitstandby</code>).
<input type="checkbox"/>	Upgrade <code>gpfdist</code> on all of your ETL hosts.
<input type="checkbox"/>	Rebuild any custom modules against your 4.3.x installation.
<input type="checkbox"/>	Download and install any Greenplum Database extensions.
<input type="checkbox"/>	(Optional) Install the latest Greenplum Command Center and update your environment to point to the latest Command Center binaries.
<input type="checkbox"/>	Inform all database users of the completed upgrade.

For Users Running Greenplum Database 4.1.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.33.7.

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 4.3.33.7*.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.33.7.

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 *Dell EMC 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 4.3.33.7*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.33.7.

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 *Dell EMC 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 *Dell EMC 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 4.3.33.7*.

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_mirror` and `gpcheckcat` (located in `~/gpAdminLogs`)

Update for `gp_toolkit.gp_bloat_expected_pages` Issue

In Greenplum Database 4.3.29.0 and earlier 4.3.x releases, the Greenplum Database view `gp_toolkit.gp_bloat_expected_pages` view might incorrectly report that a root partition table is bloated even though root partition tables do not contain data. This information could cause a user to run a `VACUUM FULL` operation on the partitioned table when the operation was not required. The issue was resolved in Greenplum Database 4.3.30.0 (resolved issue 29523).

When updating Greenplum Database, the `gp_toolkit.gp_bloat_expected_pages` function must be updated in databases created with a Greenplum Database 4.3.29.0 or earlier 4.3.x release. This issue has been fixed in databases created with Greenplum Database 4.3.30.0 and later.

To check whether the `gp_toolkit.gp_bloat_expected_pages` view in a database requires an update, run the `psql` command `\d+` to display the view definition.

```
\d+ gp_toolkit.gp_bloat_expected_pages
```

The updated view definition contains this predicate.

```
AND NOT EXISTS
( SELECT parrelid
  FROM pg_partition
  WHERE parrelid = pgc.oid )
```

Perform the following steps as the gpadmin user to update the view on each database that was created with Greenplum Database 5.11.0 or an earlier 5.x release.

1. Copy the script into a text file on the Greenplum Database master.
2. Run the script on each database that requires the update.

This example updates `gp_toolkit.gp_bloat_expected_pages` view in the database `mytest` and assumes that the script is in the `gp_bloat_expected_pages` in the `gpadmin` home directory.

```
psql -f /home/gpadmin/gp_bloat_expected_pages.sql -d mytest
```

Run the script during a low activity period. Running the script during a high activity period does not affect database functionality but might affect performance.

Script to Update `gp_toolkit.gp_bloat_expected_pages` View

```
BEGIN;
CREATE OR REPLACE VIEW gp_toolkit.gp_bloat_expected_pages
AS
  SELECT
    btdrelid,
    btdrelpages,
    CASE WHEN btdexppages < numsegments
      THEN numsegments
      ELSE btdexppages
    END as btdexppages
  FROM
    ( SELECT
      oid as btdrelid,
      pgc.relpages as btdrelpages,
      CEIL((pgc.reltuples * (25 + width))::numeric /
current_setting('block_size')::numeric) AS btdexppages,
      (SELECT numsegments FROM gp_toolkit.__gp_number_of_segments) AS
numsegments
    FROM
      ( SELECT pgc.oid, pgc.reltuples, pgc.relpages
        FROM pg_class pgc
        WHERE NOT EXISTS
          ( SELECT iaoid
            FROM gp_toolkit.__gp_is_append_only
            WHERE iaoid = pgc.oid AND iaotype = 't' )
        AND NOT EXISTS
          ( SELECT parrelid
            FROM pg_partition
            WHERE parrelid = pgc.oid )) AS pgc
      LEFT OUTER JOIN
        ( SELECT starelid, SUM(stawidth * (1.0 - stanullfrac)) AS width
          FROM pg_statistic pgs
          GROUP BY 1) AS btwcols
      ON pgc.oid = btwcols.starelid
      WHERE starelid IS NOT NULL) AS subq;

GRANT SELECT ON TABLE gp_toolkit.gp_bloat_expected_pages TO public;
COMMIT;
```



```

        THEN 'significant amount of bloat suspected'::text
    WHEN bloatidx = -1
        THEN 'diagnosis inconclusive or no bloat suspected'::text
    END AS bloatdiag
FROM
(
    SELECT
        CASE
            WHEN $3 = 't' THEN 0
            WHEN $1 < 10 AND $2 = 0 THEN -1
            WHEN $2 = 0 THEN 2
            WHEN $1 < $2 THEN 0
            WHEN ($1/$2)::numeric > 10 THEN 2
            WHEN ($1/$2)::numeric > 3 THEN 1
            ELSE -1
        END AS bloatidx
    ) AS bloatmapping

$$
LANGUAGE SQL READS SQL DATA;

GRANT EXECUTE ON FUNCTION gp_toolkit.gp_bloat_diag(int, numeric, bool, OUT
int, OUT text) TO public;

CREATE OR REPLACE VIEW gp_toolkit.gp_bloat_diag
AS
    SELECT
        btdrelid AS bdirelid,
        fnspname AS bdinspname,
        fnrelname AS bdirelname,
        btdrelpages AS bdirelpages,
        btdexppages AS bdiexppages,
        bltdiag(bd) AS bdidiag
    FROM
    (
        SELECT
            fn.*, beg.*,
            gp_toolkit.gp_bloat_diag(btdrelpages::int, btdexppages::numeric,
iao.iaotype::bool) AS bd
        FROM
            gp_toolkit.gp_bloat_expected_pages beg,
            pg_catalog.pg_class pgc,
            gp_toolkit.__gp_fullname fn,
            gp_toolkit.__gp_is_append_only iao

        WHERE beg.btdrelid = pgc.oid
            AND pgc.oid = fn.fnoid
            AND iao.iaoid = pgc.oid
        ) as bloatsummary
    WHERE bltidx(bd) > 0;

GRANT SELECT ON TABLE gp_toolkit.gp_bloat_diag TO public;
COMMIT;

```

Greenplum Database Tools Compatibility

- *Client Tools*
- *Greenplum Command Center*

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 the Pivotal Greenplum page on *Pivotal Network*.

Table 4: Greenplum Database Client Tools Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql)	4.3	4.3
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC ¹) PostgreSQL Client C API (libpq)	4.3	4.3
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.3 ²	4.3

Note:

¹The 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>).

²Greenplum Database Loaders 4.3.33.7 are compatible with Greenplum Database servers 4.3.5 and later.

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

- AIX 7.1 and AIX 7.2 (Client and Load Tools only)
- AIX 5.3L and AIX 6.1 (64-bit)
- AIX 5.3L (32-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 10 (32-bit and 64-bit) (Client and Load Tools only)
- Windows 8 (32-bit and 64-bit) (Client and Load Tools only)
- Windows 7 (32-bit and 64-bit)
- Windows Server 2012 R2 (32-bit and 64-bit) (Client and Load Tools only)
- Windows Server 2012 (32-bit and 64-bit) (Client and Load Tools only)

- 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 4.3.33.2.

Greenplum Command Center

Greenplum Command Center monitors system performance metrics, analyzes system health, and allows administrators to perform some management tasks in a Greenplum environment.

Greenplum Command Center is available from the Pivotal Greenplum page on *Pivotal Network*.

Table 5: Greenplum Command Center Compatibility

Greenplum Database Version	Greenplum Command Center Version
4.3.13.0 and later	3.2.1 and later
4.3.12.0 and earlier	All versions except 3.2.1

Greenplum Database Extensions

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 6: Greenplum Database Extension Components

Greenplum Database Extension	Extension Components	
	Name	Version
PostGIS 2.0.2 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PL/Java 1.3.2 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.8.0 Update 172
PL/Java 1.3, 1.3.1 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31

Greenplum Database Extension	Extension Components	
	Name	Version
PL/R 2.2, 2.3 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 1.3 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 7.x, 6.x, 5.x
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
MADlib 1.x for Greenplum Database 4.3.x.x ¹	MADlib	Based on MADlib version 1.x (1.17 - 1.10, 1.9.1, 1.9)

Note: Greenplum Database 4.3.33.7 does not support the PostGIS 1.0 extension package.

¹Pivotal recommends that you upgrade to the most recent version of MADlib. For MADlib support and upgrade information, refer to the [MADlib FAQ](#). For information on installing the MADlib extension in Greenplum Database, see [Greenplum MADlib Extension for Analytics](#) in the [Greenplum Database Reference Guide](#).

Greenplum Database 4.3.33.7 supports these minimum Greenplum Database extensions package versions.

Table 7: Greenplum Database 4.3.33.7 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.2 and release gpdb4.3orca
PL/Java	1.3.2 and release gp4
PL/Perl	1.2 and release gpdb4.3orca
PL/R	2.3 and release gp4
Pgcrypto (see Note)	1.3 and release gpdb4.3orca
MADlib	1.17 and release gpdb4.3orca

Greenplum Database Extension	Minimum Package Version
Python Data Science Modules	1.0.0 and release gp4
R Data Science Libraries	1.0.0 and release gp4

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 GPORCA. 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 4.3.33.7, 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 4.3.33.7.

For the pgcrypto extension, these restrictions apply.

- The pgcrypto extension package version pv1.2 and earlier are not compatible with Greenplum Database 4.3.33.7.

When you upgrade to Greenplum Database 4.3.33.7 and the pgcrypto package version pv1.2 or earlier is installed in your current system, you must uninstall the old pgcrypto extension and install the new pgcrypto extension.

- The pgcrypto extension package version pv1.3 is not compatible with Greenplum Database 4.3.15.0 and earlier. Do not install this release of the pgcrypto extension in systems running Greenplum Database 4.3.15.0 and earlier.

Package File Naming Convention

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

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

This example is the package name for a postGIS package.

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

pkgname-ver - The package name and optional version of the software that was used to create the package extension. If the package is based on open source software, the version has format *ossvversion*. The version is the version of the open source software that the package is based on. For the postGIS package, *ossv2.0.3* specifies that the package is based on postGIS version 2.0.3.

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

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

Hadoop Distribution Compatibility

This table lists the supported Hadoop distributions:

Table 8: Supported Hadoop Distributions

Hadoop Distribution	Version	gp_hadoop_target_version
Pivotal HD ³	Pivotal HD 3.0, 3.0.1	gphd-3.0
	Pivotal HD 2.0, 2.1	gphd-2.0
	Pivotal HD 1.0 ¹	
Greenplum HD ³	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
Hortonworks Data Platform	HDP 2.x	hdp2
MapR ²	MapR 4.x, MapR 5.x	gpmr-1.2
Apache Hadoop	2.x	hadoop2

Notes:

1. Pivotal HD 1.0 is a distribution of Hadoop 2.0.
2. MapR requires the MapR client.
3. Support for these Hadoop distributions has been deprecated and will be removed in a future release: Pivotal HD and Greenplum HD.

Greenplum Database 4.3.33.7 Documentation

For the latest Greenplum Database documentation go to [Pivotal Greenplum Database Documentation](#). Greenplum Database documentation is provided in HTML and PDF formats.

Table 9: Greenplum Database Documentation

Title	Revision
<i>Greenplum Database 4.3.33.7 Release Notes</i>	A01
<i>Greenplum Database 4.3 Installation Guide</i>	A32
<i>Greenplum Database 4.3 Administrator Guide</i>	A48
<i>Greenplum Database 4.3 Reference Guide</i>	A51
<i>Greenplum Database 4.3 Utility Guide</i>	A47
<i>Greenplum Database 4.3 Client Tools for UNIX</i>	A09
<i>Greenplum Database 4.3 Client Tools for Windows</i>	A07
<i>Greenplum Database 4.3 Connectivity Tools for UNIX</i>	A09
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A07
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A15
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A15

Title	Revision
<i>Greenplum Command Center Administrator Guide</i> *	----
<i>Greenplum Workload Manager User Guide</i> *	----

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

Copyright

[Privacy Policy](#) | [Terms of Use](#)

Copyright © 2020 VMware, Inc. or its affiliates. All rights reserved.