

Greenplum Database 4.3.21.0 Release Notes

Rev: A02

Updated: February, 2018

Welcome to Pivotal Greenplum Database 4.3.21.0

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*.

Note: This document contains pertinent release information about Greenplum Database 4.3.21.0. 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 *Pivotal Greenplum Database end of life 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 4.3.21.0

Greenplum Database 4.3.21.0 is a minor release that includes a product enhancement, changes, and resolves some known issues. Please refer to the following sections for more information about this release.

- *Product Enhancement*
- *Experimental Feature*
- *Supported Platforms*
- *Resolved Issues in Greenplum Database 4.3.21.0*
- *Known Issues in Greenplum Database 4.3.21.0*
- *Upgrading to Greenplum Database 4.3.21.0*
- *Greenplum Database Tools Compatibility*
- *Greenplum Database Extensions*
- *Hadoop Distribution Compatibility*
- *Greenplum Database 4.3.21.0 Documentation*

Product Enhancement

Greenplum Database 4.3.21.0 includes this new feature.

Specify the Temporary Loading Table for the gpload Utility

In Greenplum Database 4.3.21.0, to improve performance, you can specify the external table that the `gpload` utility uses to load data.

In the `gpload` control file YAML file, you can specify the table name with the `STAGING_TABLE` parameter in the `PRELOAD` section. The utility creates the table if it does not exist. If the table exists, the utility uses the existing table. If the table exists but the table schema does not match the table that is being loaded, an error is returned.

In previous releases, the utility constructed a unique, random table name for the external table.

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

Experimental Feature

In Pivotal Greenplum Database, experimental features 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: Experimental features are not recommended or supported for production deployments. These features may change in or be removed from future versions of the product based on further testing and feedback. Moreover, any features that may be visible in the open source code but that are not described in the product documentation should be considered experimental and unsupported for production use.

Pivotal Greenplum Database includes this experimental feature.

- The `gpbackup` and `gprestore` are experimental utilities that are designed to improve the performance, functionality, and reliability of backups as compared to `gpcrondump` and `gpdbrstore`. The utilities were introduced in 4.3.18.0. `gpbackup` utilizes `ACCESS SHARE` locks at the individual table level, instead of `EXCLUSIVE` locks on the `pg_class` catalog table. This enables you to execute DDL statements during the backup, such as `CREATE`, `ALTER`, `DROP`, and `TRUNCATE` operations, as long as those operations do not target the current backup set.

In Greenplum Database 4.3.21.0, `gpbackup` and `gprestore` include these new features.

- `gpbackup` lets you specify the gzip compression level for data files with the `--compression-level` option. A valid level is an integer between 1 and 9. `gpbackup` uses the default compression level of 1.
- `gpbackup` can optionally create a single backup file per segment instance when you specify the `-single-data-file` option. By default, `gpbackup` creates one `.csv` file per table, per segment, during a backup operation.
- `gprestore` now supports restoring specific schemas and tables in a backup set using the `-include-schema` and `-include-table-file` options.

In Greenplum Database 4.3.21.0 the `gpbackup` and `gprestore` utilities been updated to v1.0.0-alpha.3 and use a single file for global and database-specific metadata. This version is not compatible with backup file sets created with versions v1.0.0-alpha.1 or v1.0.0-alpha.2.

Backup files created with `gpbackup` are designed to provide future capabilities for restoring individual database objects along with their dependencies, such as functions and required user-defined datatypes. See *Parallel Backup with gpbackup and gprestore* in the *Greenplum Database Administrator Guide* for more information.

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.

Supported Platforms

Greenplum Database 4.3.21.0 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 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 4.3.21.0 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 1: Data Domain Boost Compatibility

Greenplum Database	Data Domain Boost	DDOS
4.3.21.0	3.3	6.1 (all versions)
4.3.20.0	3.0.0.3 ¹	6.0 (all versions)
4.3.19.0		5.7 (all versions)
4.3.18.0		5.6 (all versions) 5.5 (all versions) ¹
4.3.17.1	3.3	6.0 (all versions)
4.3.17.0	3.0.0.3 ¹	5.7 (all versions)
4.3.16.1		5.6 (all versions)
4.3.16.0		5.5 (all versions) ¹
4.3.15.0		
4.3.14.1		
4.3.14.0		
4.3.13.0		
4.3.12.0	3.0.0.3 ¹	5.7 (all versions)
4.3.11.3		5.6 (all versions)
4.3.11.2		5.5 (all versions) ¹
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.

¹Support 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 4.3.21.0 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.
- Greenplum Database 4.3.x, all versions, is supported on DCA V1, and requires DCA software version 1.2.2.2 or greater due to known DCA software issues in older DCA software versions.

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

Note: Greenplum Database 4.3.21.0 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) servers using either Amazon instance store (Amazon uses the volume names `ephemeral[0-20]`) or Amazon Elastic Block Store

(Amazon EBS) storage. If using Amazon EBS storage the storage should be RAID of Amazon EBS volumes and mounted with the XFS file system for it to be a supported configuration.

- For Red Hat Enterprise Linux 7.2 or CentOS 7.2, the default `systemd` setting `RemoveIPC=yes` removes IPC connections when non-system users logout. This causes the Greenplum Database utility `gpinitssystem` to fail with semaphore errors. To avoid this issue, see "Setting the Greenplum Recommended OS Parameters" in the *Greenplum Database Installation Guide*.

Resolved Issues in Greenplum Database 4.3.21.0

The table below lists issues that are now resolved in Pivotal Greenplum Database 4.3.21.0

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 2: Resolved Issues in 4.3.21.0

Issue Number	Category	Resolved In	Description
29186	Query Execution	4.3.21.0	<p>Greenplum Database generated an ERROR on a Greenplum Database segment when Greenplum Database aborted a transaction when a query executor process received invalid numeric input. The error was caused by improper memory handling during query workfile cleanup after the transaction was aborted.</p> <p>This issue has been resolved. Memory handling for the specified situation has been improved.</p>
29182	Transaction Management	4.3.21.0	<p>In some cases, a <code>VACUUM</code> operation did not complete a transaction when vacuuming a database object and caused an orphan prepared transaction. The orphan prepared transaction caused issues when attempting another <code>VACUUM</code> operation on the same object, and after activating a primary segment after a failover. The issue was caused when the <code>VACUUM</code> operation did not handle transaction failures properly.</p> <p>This issue has been resolved. The handling of failed transactions by <code>VACUUM</code> has been improved.</p>
29167	Query Planner	4.3.21.0	<p>For an <code>UPDATE</code> or <code>DELETE</code> command on a partitioned table that contains a subquery and a filter on a relation that appears in the subquery, the legacy query planner generated a plan with an incorrect filter for the relation inside the subquery. This caused a Greenplum Database PANIC.</p> <p>This issue has been resolved. Now the legacy planner generates a valid plan.</p>

Issue Number	Category	Resolved In	Description
29095	Query Optimizer	4.3.21.0	GPORCA fell back to the legacy planner when translating a query plan that contains a <code>CAST</code> on a subplan. This issue has been resolved. Now GPORCA handles the specified type of query plan.
26781	PL/Java	4.3.21.0	If the Greenplum Database server configuration parameter <code>pljava_classpath_insecure</code> was set to <code>true</code> , a database user without superuser privileges could not change the server configuration parameter <code>pljava_classpath</code> . This issue has been resolved. Now the parameter setting <code>pljava_classpath_insecure=true</code> allows non-superusers to change <code>pljava_classpath</code> .
153757596	Query Optimizer	4.3.21.0	GPORCA did not estimate the cardinality correctly when there is a comparison between a scalar <code>ident</code> and an array that consists of constants and non-constants. This issue has been resolved. Cardinality estimation has been improved for the specified situation.
136132985	Query Optimizer	4.3.21.0	For some queries that contain correlated subqueries (CSQ), GPORCA returned incorrect results. GPORCA did not properly handle a subquery that contains both a filter such as <code>exists</code> or <code>not in</code> and an outer reference. This issue has been resolved. The handling of the specified type of CSQ has been improved.

Known Issues in Greenplum Database 4.3.21.0

This section lists the known issues in Greenplum Database 4.3.21.0. 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 4.3.21.0

Issue	Category	Description
26589	Storage: Catalog and Metadata	<p>Greenplum Database does not acquire a lock on a schema when creating a table inside the schema. A concurrent <code>DROP</code> of the schema and <code>CREATE TABLE</code> operation will result in a leaked object (an orphan table or orphan relation columns) in the system catalog and possibly on disk. For example, a relation created with this command</p> <pre data-bbox="656 478 1243 510">CREATE TABLE mytest.foobar (coll int)</pre> <p>during a concurrent <code>DROP</code> of the schema <code>mytest</code> might leak either the table itself or the <code>coll</code> attribute object.</p> <p>These leaked objects do not affect any future queries. The catalog inconsistencies can be detected with the <code>gpcheckcat</code> utility.</p> <p>Workaround: To block a concurrent <code>DROP</code> from occurring, acquire and hold a <code>ROW SHARE</code> lock on the row of the <code>pg_namespace</code> catalog table that corresponds to the schema. For example, this transaction acquires a <code>ROW SHARE</code> lock on the <code>pg_namespace</code> catalog table row for the schema <code>my_schema</code>.</p> <pre data-bbox="656 894 1425 1058">begin; SELECT nspname FROM pg_namespace WHERE nspname = 'my_schema' FOR SHARE; ... end;</pre> <p>During the transaction, the lock prevents the schema from being dropped.</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="656 1352 1219 1404">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 5, 2017 • November 4, 2018 • November 3, 2019 <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, python command displays the <code>gpload</code> help information on the command line.</p> <pre data-bbox="651 1310 964 1352">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>

Issue	Category	Description
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.
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	<p>Setting the Greenplum Database server configuration parameter <code>log_hostname</code> to <code>on</code> Greenplum Database segment hosts causes an Interconnect Error that states that the listeneraddress name or service not known.</p> <p>The parameter should be set to <code>on</code> only on the Greenplum Database master.</p>
25280	Management Scripts: gpstart/gpstop	<p>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>.</p> <p>Workaround: Unset the environment variable <code>LANG</code> before running the <code>gpstop</code> utility. For example:</p> <pre>\$ unset LANG</pre>
25246	Management Scripts: gpconfig	<p>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.</p> <pre>\$ gpconfig -c gp_email_to -v 'test@example.com'</pre> <p>The improperly set parameter causes Greenplum Database to fail when it is restarted.</p> <p>Workaround: Enclose the value for <code>gp_email_to</code> or <code>gp_email_from</code> with double quotes.</p> <pre>\$ gpconfig -c gp_email_to -v "'test@example.com'"</pre>

Issue	Category	Description
25168	Locking, Signals, Processes	<p>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.</p> <p>The <code>client_min_messages</code> parameter should not be set a value higher than <code>ERROR</code>.</p>
24588	Management Scripts: <code>gpconfig</code>	<p>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>).</p> <p>Workaround: The SQL command <code>SHOW</code> displays the correct <code>gp_enable_gpperfmon</code> value.</p>
24031	<code>gphdfs</code>	<p>If a readable external table is created with <code>FORMAT 'CSV'</code> and uses the <code>gphdfs</code> protocol, reading a record fails if the record spans multiple lines and the record is stored in multiple HDFS blocks.</p> <p>Workaround: Remove line separators from within the record so that the record does not span multiple lines.</p>
23824	Authentication	<p>In some cases, LDAP client utility tools cannot be used after running the source command:</p> <pre>source \$GPHOME/greenplum_path.sh</pre> <p>because the LDAP libraries included with Greenplum Database are not compatible with the LDAP client utility tools that are installed with operating system.</p> <p>Workaround: The LDAP tools can be used without running the source command in the environment.</p>
23366	Resource Management	<p>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.</p>
23492	Backup and Restore,	<p>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>.</p>

Issue	Category	Description
23521	Client Access Methods and Tools	<p>Hadoop YARN based on Hadoop 2.2 or later does not work with Greenplum Database.</p> <p>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.</p>
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>gpdbrestore</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	<p>SSL is only supported on the master host. It is not supported on segment hosts.</p>
20822	Backup and Restore	<p>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>--ddboost-host</code> and <code>--ddboost-user</code>.</p>
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>gpcrondump</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>gpcrondump</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>--ddbboost</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.

Issue	Category	Description
17513	Management Scripts Suite	<p>Running more than one <code>gpfilespace</code> command concurrently with itself to move either temporary files (<code>--movetempfilespace</code>) or transaction files (<code>--movetransfilespace</code>) to a new filespace can in some circumstances cause OID inconsistencies.</p> <p>Workaround: Do not run more than one <code>gpfilespace</code> command concurrently with itself. If an OID inconsistency is introduced <code>gpfilespace --movetempfilespace</code> or <code>gpfilespace --movetransfilespace</code> can be used to revert to the default filespace.</p>
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	<p>Resource queue deadlocks may be encountered if a cursor is associated with a query invoking a function within another function.</p>
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>

Issue	Category	Description
17189	Loaders: gpfdist	<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>gpdbrstore --ddboost</code> to restore a compressed dump, the restore parameters incorrectly show “Restore compressed dump = Off”. This error occurs even if <code>gpdbrstore</code> passes the <code>--gp-c</code> option to use <code>gunzip</code> for in-line de-compression.</p>
15899	Backup and Restore	When running <code>gpdbrstore</code> with the list (<code>-L</code>) option, external tables do not appear; this has no functional impact on the restore job.

Upgrading to Greenplum Database 4.3.21.0

The upgrade path supported for this release is Greenplum Database 4.2.x.x to Greenplum Database 4.3.21.0. 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 4.3.21.0, 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 4.3.21.0*
- *Upgrading from 4.3.x to 4.3.21.0 on Pivotal DCA Systems*
- *Upgrading from 4.2.x.x to 4.3.21.0*
- *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:

```
gpccrondump --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 4.3.21.0

An upgrade from 4.3.x to 4.3.21.0 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 4.3.21.0 on Pivotal DCA Systems*. This section is for upgrading to Greenplum Database 4.3.21.0 on non-DCA systems.

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_diag` function 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 4.3.21.0 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-4.3.21.0
```

5. If your Greenplum Database deployment uses LDAP authentication, manually edit the `/usr/local/greenplum-db/greenplum_path.sh` file to add the line:

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

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-4.3.21.0/greenplum_path.sh
```

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

```
$ rm /usr/local/greenplum-db
$ ln -s /usr/local/greenplum-db-4.3.21.0 /usr/local/greenplum-db
```

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

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

8. Run the `gpsegininstall` utility to install the 4.3.21.0 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.
11. After all segment hosts have been upgraded, you can log in as the `gpadmin` user and restart your Greenplum Database system:

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

12. 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.

13. 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.

14. 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 4.3.21.0 as follows:

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

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

fix_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 4.3.21.0 you do not need to run the `fix_visimap_owner.sql` script. Also, you can run this script on Greenplum Database 4.3.x earlier than 4.3.6.0 to fix the incorrect ownership assignment of visimap relations.

fix_ao_upgrade.py Utility

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

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

Syntax

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

Options**-r | --report**

Report inconsistencies without making any changes.

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

Greenplum Database master hostname or IP address.

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

Greenplum Database master port.

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

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

v | --verbose

Verbose output that includes table names.

--help

Show the help message and exit.

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

Dropping Orphan Tables on Greenplum Database Segments

If you upgraded to Greenplum Database 4.3.6.0 and a user dropped a table, in some cases, the table would be dropped only on the Greenplum Database master, not on the Greenplum Database segments. This created orphan tables on Greenplum Database segments. This issue occurs only with Greenplum Database 4.3.6.0. However, the orphan tables remain in Greenplum Database after upgrading to 4.3.21.0.

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 4.3.21.0 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 4.3.21.0 on Pivotal DCA Systems

Upgrading Greenplum Database from 4.3.x to 4.3.21.0 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 4.3.21.0 on DCA systems. This section is only for installing Greenplum Database 4.3.21.0 on DCA systems.

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_diag` function 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 4.3.21.0 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 4.3.21.0 for Redhat Enterprise Linux 5.x.

```
# ./greenplum-db-appliance-4.3.21.0-build-1-RHEL5-x86_64.bin hostfile
```


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

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

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

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

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

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

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

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

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

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

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

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

9. If you are utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.3.x to 4.3.21.0 as follows:

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

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

Upgrading from 4.2.x.x to 4.3.21.0

This section describes how you can upgrade from Greenplum Database 4.2.x.x or later to Greenplum Database 4.3.21.0. 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 filesystem) 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-4.3.21.0-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.3.21.0-PLATFORM.bin
```

4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press `ENTER` to accept the default install path (for example: `/usr/local/greenplum-db-4.3.21.0`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer installs the Greenplum Database software and creates a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Source the path file from your new 4.3.21.0 installation. This example changes to the `gpadmin` user before sourcing the file:

```
# su - gpadmin
$ source /usr/local/greenplum-db-4.3.21.0/greenplum_path.sh
```

8. Run the `gpsegininstall` utility to install the 4.3.21.0 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 4.3.21.0 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 4.3.21.0.

```
# ./greenplum-db-appliance-4.3.21.0-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 file 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 4.3.21.0 installation. For example:

```
$ source /usr/local/greenplum-db-4.3.21.0/greenplum_path.sh
```

On a DCA system, the path to the file would be similar to `/usr/local/GP-4.3.21.0/greenplum_path.sh`.

6. Update the Greenplum Database environment so it is referencing your new 4.3.21.0 installation.
 - a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.3.21.0 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.3.21.0 /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-4.3.21.0 /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-4.3.21.0 /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-4.3.21.0 /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-4.3.21.0
```

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-4.3.21.0`.

8. As `gpadmin`, run the 4.3.21.0 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-4.3.21.0
```

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-4.3.21.0`.

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 4.3.21.0 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 4.3.21.0 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. Rebuild any modules that were built against the provided C language header files for use with Greenplum Database 4.3.5.0 or later. (for example, any shared library files for user-defined functions in `$GPHOME/lib`). See your operating system documentation and your system administrator for information about rebuilding and compiling modules such as shared libraries.
4. Use the Greenplum Database `gppkg` utility to install Greenplum Database extensions. If you were previously using any Greenplum Database extensions such as `pgcrypto`, `PL/R`, `PL/Java`, `PL/Perl`, and `PostGIS`, download the corresponding packages from [Pivotal Network](#), and install using this utility. See the *Greenplum Database Utility Guide* for `gppkg` usage details.

- 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](#).

- (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
```

- Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.3.21.0 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 4.3.21.0. Detailed upgrade instructions are provided in [Upgrading from 4.2.x.x to 4.3.21.0](#).

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 4.3.21.0.

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 4.3.21.0](#).

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.3.21.0.

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 4.3.21.0*.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.3.21.0.

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 4.3.21.0*.

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_diag` Issue

In Greenplum Database 4.3.19.0 or an earlier 4.3.x release, Greenplum Database returned an integer out of range error in some cases when performing a query against the `gp_toolkit.gp_bloat_diag` view. The issue was resolved in Greenplum Database 4.3.20.0 (resolved issue [26518](#)).

When updating Greenplum Database, the `gp_toolkit.gp_bloat_diag` function and view must be updated in databases created with a Greenplum Database 4.3.19.0 or an earlier 4.3.x release. This issue has been fixed in databases created with Greenplum Database 4.3.20.0 and later.

To check whether the `gp_toolkit.gp_bloat_diag` function and view in a database requires an update, run the `psql` command `\df` to display information about the `gp_toolkit.gp_bloat_diag` function.

```
\df gp_toolkit.gp_bloat_diag
```

If the data type for `btdexppages` is `integer`, an update is required. If the data type is `numeric` an update is not required. In this example, the `btdexppages` data type is `integer` and requires an update.

```
List of functions
-[ RECORD 1 ]-----
+-----+-----+-----+-----+-----+-----+
Schema          | gp_toolkit
Name             | gp_bloat_diag
Result data type | record
Argument data types | btdreldpages integer, btdexppages integer, aotable
boolean, OUT bltidx integer, OUT bltdiag text
Type            | normal
```

Run the following script to update the function and view to fix the issue on each database that was created with Greenplum Database 4.3.19.0 or an earlier 4.3.x release.

As the `gpadmin` user, follow these steps.

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_diag` function and view in the database `mytest` and assumes that the script is in the `update_bloat_diag.sql` in the `gpadmin` home directory.

```
psql -f /home/gpadmin/update_bloat_diag.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_diag` Function and View

```
BEGIN;
CREATE OR REPLACE FUNCTION gp_toolkit.gp_bloat_diag(btdrelpages int,
  btdexppages numeric, aotable bool,
  OUT bltidx int, OUT bltdiag text)
AS
$$
  SELECT
    bloatidx,
    CASE
      WHEN bloatidx = 0
        THEN 'no bloat detected'::text
      WHEN bloatidx = 1
        THEN 'moderate amount of bloat suspected'::text
      WHEN bloatidx = 2
        THEN 'significant amount of bloat suspected'::text
      WHEN bloatidx = -1
        THEN 'diagnosis inconclusive or no bloat suspected'::text
    END AS bloatdiag
  FROM
    (
      SELECT
        CASE
          WHEN $3 = 't' THEN 0
          WHEN $1 < 10 AND $2 = 0 THEN -1
          WHEN $2 = 0 THEN 2
          WHEN $1 < $2 THEN 0
          WHEN ($1/$2)::numeric > 10 THEN 2
          WHEN ($1/$2)::numeric > 3 THEN 1
          ELSE -1
        END AS bloatidx
      ) AS bloatmapping
$$
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(btdreldpages::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 4.3.20.0 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 4.3.20.0.

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 2.0.1 for Greenplum Database 4.3.x.x	PostGIS	2.0.3
	Proj	4.8.0
	Geos	3.3.8
PL/Java 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
PL/Java 1.3 for Greenplum Database 4.3.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 2.3 for Greenplum Database 4.3.x.x	PL/R	8.3.0.16
	R	3.1.1
PL/R 2.2 for Greenplum Database 4.3.x.x	PL/R	8.3.0.16
	R	3.1.1
PL/R 2.1 for Greenplum Database 4.3.x.x	PL/R	8.3.0.15
	R	3.1.0
PL/R 1.0 for Greenplum Database 4.3.x.x	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.2 for Greenplum Database 4.3.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.16.3 on RHEL 7.x 5.12.4 on RHEL 6.x 5.5.8 on RHEL 5.x
PL/Perl 1.1 for Greenplum Database	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4 on RHEL 5.x

Greenplum Database Extension	Extension Components	
	Name	Version
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 (1.13, 1.12, 1.11, 1.10, 1.9.1, 1.9)

Note: Greenplum Database 4.3.20.0 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 4.3.20.0 supports these minimum Greenplum Database extensions package versions.

Table 7: Greenplum Database 4.3.20.0 Package Version

Greenplum Database Extension	Minimum Package Version
PostGIS	2.0.1 and release <code>gpdb4.3orca</code>
PL/Java	1.3.1 and release <code>gpdb4.3orca</code>
PL/Perl	1.2 and release <code>gpdb4.3orca</code>
PL/R	2.3 and release <code>gpdb4.3orca</code>
Pgcrypto (see Note)	1.3 and release <code>gpdb4.3orca</code>
MADlib	1.9 and release <code>gpdb4.3orca</code>
Python Data Science Modules	1.0.0 and release <code>gp4</code>
R Data Science Libraries	1.0.0 and release <code>gp4</code>

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 4.3.20.0, you must install and use Greenplum Database extension packages (gppkg files and contrib modules) that are built for Greenplum Database 4.3.5.0 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 4.3.20.0.

For the pgcrypto extension, these restrictions apply.

- The pgcrypto extension package version pv1.2 and earlier are not compatible with Greenplum Database 4.3.20.0.

When you upgrade to Greenplum Database 4.3.20.0 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.

pvpkg-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 ⁴	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 ⁴	Greenplum HD 1.2	gphd-1.2
	Greenplum HD 1.1	gphd-1.1 (default)
Cloudera ⁴	CDH 5.2, 5.3, 5.4.x - 5.8.x	cdh5
	CDH 5.0, 5.1	cdh4.1

Hadoop Distribution	Version	gp_hadoop_target_version
	CDH 4.1 ² - CDH 4.7	cdh4.1
Hortonworks Data Platform	HDP 2.1, 2.2, 2.3, 2.4, 2.5	hdp2
MapR ^{3,4}	MapR 4.x, MapR 5.x	gpmr-1.2
	MapR 1.x, 2.x, 3.x	gpmr-1.0
Apache Hadoop	2.x	hadoop2

Notes:

1. Pivotal HD 1.0 is a distribution of Hadoop 2.0
2. For CDH 4.1, only CDH4 with MRv1 is supported
3. MapR requires the MapR client. For MapR 5.x, only TEXT and CSV are supported in the FORMAT clause of the CREATE EXTERNAL TABLE command.
4. Support for these Hadoop distributions have been deprecated and will be removed in a future release: Pivotal HD, Greenplum HD, Cloudera CDH 4.1 - CDH 4.7, and MapR 1.x, 2.x, 3.x.

Greenplum Database 4.3.21.0 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 4.3.21.0 Release Notes</i>	A02
<i>Greenplum Database 4.3 Installation Guide</i>	A28
<i>Greenplum Database 4.3 Administrator Guide</i>	A36
<i>Greenplum Database 4.3 Reference Guide</i>	A37
<i>Greenplum Database 4.3 Utility Guide</i>	A36
<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>	A08
<i>Greenplum Database 4.3 Connectivity Tools for Windows</i>	A07
<i>Greenplum Database 4.3 Load Tools for UNIX</i>	A13

Title	Revision
<i>Greenplum Database 4.3 Load Tools for Windows</i>	A13
<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 © 2018 Pivotal Software, Inc. All rights reserved.

Pivotal Software, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." PIVOTAL SOFTWARE, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.