Rev: A01
Updated: November 14, 2012

Welcome to Greenplum Database 4.2.3

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

Note: This document contains pertinent release information about Greenplum Database 4.2.3. For previous versions of the release notes for Greenplum Database, go to Powerlink.

About Greenplum Database 4.2.3

Greenplum Database 4.2.3 is a maintenance release that introduces resolved issues and some performance/feature enhancements; there are no new features in this release. Please refer to the following sections for more information about this release.

- Supported Platforms
- Product Enhancements
- 4.2.3 Documentation Set
- Resolved Issues in Greenplum Database 4.2.3
- Known Issues in Greenplum Database 4.2.3
- Upgrading to Greenplum Database 4.2.x.x
- Greenplum Database Tools Compatibility
- Greenplum Database Extensions Compatibility
- Hadoop Distribution Compatibility
- Greenplum Database 4.2 Documentation

Product Enhancements

Updating Data Domain Boost

Greenplum database versions 4.2.x support dynamically updating the natively installed Data Domain Boost database plug-in via gppkg.

Prior to the availability of the Data Domain Boost gppkg package, a database upgrade was required to update the Data Domain Boost plug-in.

Performance Improvement

Greenplum Database version 4.2.2 had removed the hard limit on sub-transactions within a transaction block. In this release, the performance of transactions with large numbers of sub-transactions has been significantly improved.

4.2.3 Documentation Set

This release includes a revamped Greenplum Database documentation set that makes it easier to find the information you need. Here's what you get:

- *Database Administrator Guide*: Everyday DBA tasks, from accessing the database to defining database objects, loading, querying, and managing data, extensibility, and performance and tuning.
- *System Administrator Guide*: Greenplum Database system configuration, monitoring, maintenance, and system backup and restore.
- *Reference Guide*: Greenplum Database reference material, including SQL commands, system catalog reference, server parameters, and environment variables.
- *Utility Guide*: Management and client utilities, as well as Oracle functions.
- *Installation Guide*: Information and instructions for installing and initializing a Greenplum Database system.

All the above documentation is available from Powerlink.

Coming in a later release:

- *Developer's Guide*
- *Greenplum Database Concepts Guide*
- *Security Configuration Guide*: Greenplum Database security configuration and security adherence information.
- Troubleshooting wiki

Supported Platforms

Greenplum Database 4.2.3 runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 5.5, 5.6, 5.7, 6.1, and 6.2
- SuSE Linux Enterprise Server 64-bit 10 SP4, 11 SP1
- Solaris x86 64-bit v10 U7, U8, U9, U10
- Oracle Unbreakable Linux 64-bit 5.5
- CentOS 64-bit 5.5, 5.6, 5.7, 6.1, and 6.2

Note: For the most recent server platform support information for Greenplum Database, check the EMC Support Matrix (ESM).

To access the ESM, go to the Powerlink home page, select:

Home > **Support** > **Interoperability and Product Lifecycle Information** > **E-Lab Interoperatability Navigator**, then search for Greenplum Database.

Resolved Issues in Greenplum Database 4.2.3

The table below lists issues that are now resolved in Greenplum Database 4.2.3.

For issues resolved in prior releases, refer to the corresponding release notes available from Powerlink.

Table 1 Resolved Issues in 4.2.3

Issue Number	Category	Description
18415	Query Execution	When queries met several simultaneous conditions, incorrect results or SIGSEGV errors could occur. Queries that simultaneously met several predicate, partitioning, and redistribution requirements could generate an error.
17030	DDL and Utility Statements	Owners of user-defined types were unable to drop them without super-user permissions. Owners are now properly allowed to drop user-defined types without additional permissions.
17727	Client Access Methods	A "current transaction was aborted, commands ignored until end of transaction block" error message was generated as an error when the Autocommit option in JDBC was set to false and a rollback occurred in a transaction that has already been aborted. JDBC rollbacks are now properly managed to finish the current transaction block as required and no error messages are displayed.
16219	Client Access Methods, Dispatch	JDBC statements sometimes failed with Unexpected internal error when unanalyzed tables were accessed. JDBC statements sometimes failed with Unexpected internal error when unanalyzed tables were accessed. The issue occurred when using protocol version 3 for communication through the JDBC driver in conjunction with implicit statistics gathering for unanalyzed tables involved in the current query.
17481	Catalog and Metadata	Queries on the system view `pg_partitions` could fail to return when DDL statements on partitioned tables were running concurrently.
17896	Catalog and Metadata	Queries on views with columns defined with an implicit type cast could sometimes generate an "Unexpected internal error" message.

Table 1 Resolved Issues in 4.2.3

Issue Number	Category	Description
17995	DDL and Utility Statements	In rare circumstances, pg_cancel_backend and pg_terminate_backend did not terminate sessions as expected.
17907	Query Execution	Queries that included hashjoins created too many files when the following server parameters were set: `gp_workfile_type_hashjoin='BFZ'` and `gp_workfile_compress_algorithm='ZLIB'`. When this occured and the system reached the maximum number of open files per process, the query aborted and it tried to close all files, causing the clean-up code to fail and creating a PANIC.
18334	Backup and Restore	Direct I/O feature did not support SuSe. This feature now supports SuSe.
17848	Query Optimizer	Queries using a `WITH` clause can generate incorrect results when common table sharing is disabled (through the server parameter `gp_cte_sharing`) and several rare query plan patterns are met simultaneously.
18452	Backup and Restore	When using `gp_dump`/`pg_dump` to backup external tables with custom formatters with blank space in one or more values of the formatter definition, the dump would be generated incorrectly. Attempting to restore the affected dump would result in an error.
18130	Backup and Restore	Backups using `gpcrondump` could report an error message when the backup spanned midnight and the `-z` flag was used.
18161	Backup and Restore	When using `gpcrondump` to backup a schema that doesn't exist, a generic error message indicating the dump failed was produced instead of a specific error messaging indicating that the schema didn't exist.
17733	Loaders	On AIX platforms some cryptographic functions could fail because of a library path issue.
17875	Loaders	When reading from an external table on Hadoop, a `SELECT` with a `LIMIT` clause specifying a small number of rows could sometimes fail.
17061	DDL/DML	Queries against an append-only, column-oriented table could encounter the following generic error message: "Bad append-only storage header". This occurred only for AO/CO tables, during an index scan, in the presence of aborted rows, and for a specific set of on-disk access patterns
17593	DDL/DML	In some isolated scenarios memory freed incorrectly by an internal database process could result in a "FATAL", "53300", "sorry, too many clients" error message.
17817	Replication/Crash Recovery	During a FATAL failure, the code aborted already-committed transactions. As a result, the xlog contains committed and aborted records for the same transaction. The issue has been resolved in this release.
17422	Backup and Restore	Tables leveraging column compression that have incurred one or more `ALTER TABLE` modifications may arrive at a state in which the table-level compression settings conflict with those of individual columns. In such cases, the schemas of these tables would not be correctly dumped; while this would result in a proper dump of the data, an error would be observed during restore as the database attempts to recreate the table.
17970	Interconnect	When the value of the `gp_interconnect_type` was set to UDP, an "ack:short read recvfrom() call" error could be prematurely generated in some situations, aborting the query generating the error.
14367	DDL/DML	`ALTER TABLE ADD COLUMN` with default `NULL` was not supported for append-only tables. This syntax is now supported.

Table 1 Resolved Issues in 4.2.3

Issue Number	Category	Description
17606	DDL/DML - Partitioning	When the type of column contained in a partitioning key was altered from timestamp to date it was no longer possible to add partitions to the table.
15243	Management Scripts Suites	Under some circumstances, `gpssh` was leaking ssh processes, causing the max ptys limit to be reached. Child processes spawned from `gpssh` command are now terminated after a given interval even if the `gpssh` is under high load or was forced to shut down.
15989	Backup and Restore	`gpcrondump` failed to dump the "template1" and "postgres" databases.

Known Issues in Greenplum Database 4.2.3

This section lists the new known issues in Greenplum Database 4.2.3. A workaround is provided where applicable.

For known issues discovered in previous releases, including patch releases to Greenplum Database 4.1 or 4.0.x, see the corresponding release notes, available from Powerlink:

Table 2 All Known Issues in 4.2.3

Issue	Category	Description
16129	Management Scripts Suite	**`gpkill` does not run on the Solaris platform.** The gpkill utility is using an internal tool called “glider” to introspect processes and glean/archive some relevant information before actually killing processes. In some cases, our invocation of this tool fails to yield the desired introspective information.
15692 17192	Backup and Restore	**Greenplum Database’s implementation of RSA lock box for Data Domain Boost changes backup and restore requirements for customers running SuSE.** The current implementation of the RSA lock box for Data Domain Boost login credential encryption only supports customers running on Red Hat Enterprise Linux. **Workaround**: If you run Greenplum Database on SuSE, use NFS as your backup solution. See the *Greenplum Database System Administrator Guide* for information on setting up a NFS backup.
18850	Backup and Restore	Data Domain Boost credentials cannot be set up in some environments due to the absence of certain libraries (for example, `libstdc++`) expected to reside on the platform. **Workaround**: Install the missing libraries manually on the system.
18851	Backup and Restore	When performing a data-only restore of a particular table, it is possible to introduce data into Greenplum Database that contradicts the distribution policy of that table. In such cases, subsequent queries may return unexpected and incorrect results. To avoid this scenario, we suggest you carefully consider the table schema when performing a restore.
18785	Loaders	External web tables that use IPv6 addresses must include a port number.
18713	Catalog and Metadata	Drop language plpgsql cascade results in a loss of `gp_toolkit` functionality. **Workaround**: Reinstall `gp_toolkit`.

Table 2 All Known Issues in 4.2.3

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.
18834	Documentation	Description in the Greenplum Utility Guide and the command line help of `-B parallel_processes` option for `gpstart` and `gpstop` is incorrect. Correct description is: The number of worker threads used for parallel operations. If not specified, the utility will use up to 64 worker threads as needed.
12468	Management Scripts Suite	`gpexpand --rollback` fails if an error occurs during expansion such that it leaves the database down `gpstart` also fails as it detects that expansion is in progress and suggests to run `gpexpand --rollback` which will not work because the database is down. **Workaround**: Run `gpstart -m` to start the master and then run rollback,
18785	Loaders	Running `gpload` with the `--ssl` option and the relative path of the source file results in an error that states the source file is missing. **Workaround**: Provide the full path in the yaml file or add the loaded data file to the certificate folder.
18414	Loaders	Unable to define external tables with fixed width format and empty line delimiter when file size is larger than gpfdist chunk (by default, 32K).
14640	Backup and Restore	`gpdbrestore` outputting incorrect non-zero error message. When performing single table restore, `gpdbrestore` gives warning messages about non-zero tables however prints out zero rows.
17285	Backup and Restore	NFS backup with `gpcrondump -c` can fail. In circumstances where you haven't backed up to a local disk before, backups to NFS using `gpcrondump` with the `-c` option can fail. On fresh systems where a backup has not been previously invoked there are no dump files to cleanup and the `-c` flag will have no effect. **Workaround**: Do not run `gpcrondump` with the `-c` option the first time a backup is invoked from a system.
17837	Upgrade/ Downgrade	Major version upgrades internally depend on the `gp_toolkit` system schema. The alteration or absence of this schema may cause upgrades to error out during preliminary checks. **Workaround**: To enable the upgrade process to proceed, you need to reinstall the `gp_toolkit` schema in all affected databases by applying the SQL file found here: `$GPHOME/share/postgresql/gp_toolkit.sql`.
17513	Management Scripts Suite	Running more than one `gpfilespace` command concurrently with itself to move either temporary files (`--movetempfilespace`) or transaction files (`--movetransfilespace`) to a new filespace can in some circumstances cause OID inconsistencies. **Workaround**: Do not run more than one `gpfilespace` command concurrently with itself. If an OID inconsistency is introduced `gpfilespace --movetempfilespace` or `gpfilespace --movetransfilespace` can be used to revert to the default filespace.

Table 2 All Known Issues in 4.2.3

Issue	Category	Description
17780	DDL/DML: Partitioning	ALTER TABLE ADD PARTITION inheritance issue When performing an ALTER TABLE ADD PARTITION operation, the resulting parts may not correctly inherit the storage properties of the parent table in cases such as adding a default partition or more complex subpartitioning. This issue can be avoided by explicitly dictating the storage properties during the ADD PARTITION invocation. For leaf partitions that are already afflicted, the issue can be rectified through use of EXCHANGE PARTITION.
17795	Management Scripts Suite	Under some circumstances, gppkg on SuSe is unable to correctly interpret error messages returned by rpm. On SuSE, gppkg is unable to operate correctly under circumstances that require a non-trivial interpretation of underlying rpm commands. This includes scenarios that result from overlapping packages, partial installs, and partial uninstalls.
17604	Security	A Red Hat Enterprise Linux (RHEL) 6.x security configuration file limits the number of processes that can run on gpadmin. RHEL 6.x contains a security file (/etc/security/limits.d/90-nproc.conf) that limits available processes running on gpadmin to 1064. **Workaround**: Remove this file or increase the processes to 131072.
17415	Installer	When you run gppkg -q -info<some gppkg>, the system shows the GPDBversion as main build dev.
17334	Management Scripts Suite	You may see warning messages that interfere with the operation of management scripts when logging in. Greenplum recommends that you edit the /etc/motd file and add the warning message to it. This will send the messages to are redirected to stdout and not stderr. You must encode these warning messages in UTF-8 format.
17221	Resource Management	Resource queue deadlocks may be encountered if a cursor is associated with a query invoking a function within another function.
17113	Management Scripts Suite	Filespaces are inconsistent when the Greenplum database is down. Filespaces become inconsistent in case of a network failure. Greenplum recommends that processes such as moving a filespace be done in an environment with an uninterrupted power supply.
17189	Loaders: gpfdist	gpfdist shows the error “Address already in use” after successfully binding to socket IPv6. Greenplum supports IPv4 and IPv6. However, gpfdist fails to bind to socket IPv4, and shows the message “Address already in use”, but binds successfully to socket IPv6.
16278	Management Scripts Suite	gpkill shows that it failed to kill the gpload process, but in fact the process was successfully aborted with all the data loaded correctly.
16269	Management Scripts Suite	gpkill should attempt to kill each given pid. gpkill accepts the list of pids but only shows that one of the processes may not be killed.

Table 2 All Known Issues in 4.2.3

Issue	Category	Description

16519	Backup and Restore	Limited data restore functionality and/or restore performance issues can occur when restoring tables from a full database backup where the default backup directory was not used.
		In order to restore from backup files not located in the default directory you can use the `−R` to point to another host and directory. This is not possible however, if you want to point to a different directory on the same host (NFS for example).
		Workaround: Define a symbolic link from the default dump directory to the directory used for backup, as shown in the following example:
		1. Perform a full Database Backup to a specific NFS directory:
		`$ gpcrondump -x <db_name> -z -u /backup/DCA-93 -a`
		2. Create a file listing the segment servers:
		`$ vi /home/gpadmin/segments`
		`sdw1`
		`sdw2`
		`sdw3`
		`….`
		3. Remove the relevant date folder from db_dumps directories on the master and segments:
		`$ rm -r /data/master/gpseg-1/db_dumps/20120619`
		`$ gpssh -f segments 'rm -r /data1/primary/gpseg*/db_dumps/20120619'`
		`$ gpssh -f segments 'rm -r /data2/primary/gpseg*/db_dumps/20120619'`
		4. Create a symbolic link between the master and segment directories and the directory to which you backed up in step 1. Only the master and sdw1 was shown here, write a script for the remaining segments:
		`$ ln -s /backup/DCA-93/db_dumps/20120619 /data/master/gpseg-1/db_dumps/20120619`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data1/primary/gpseg0/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data1/primary/gpseg1/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data1/primary/gpseg2/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data2/primary/gpseg3/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data2/primary/gpseg4/db_dumps/20120619'`
		`$ gpssh -h sdw1 'ln -s /backup/DCA-93/db_dumps/20120619 /data2/primary/gpseg5/db_dumps/20120619'`
		5. Restore from backup files:
		`$ gpdbrestore -t 20120619061835 -T <schema.table> -a`
		6. Remove the symbolic links
		`$ rrm -r /data/master/gpseg-1/db_dumps/20120619`
		`$ gpssh -f segments 'rm -r /data1/primary/gpseg*/db_dumps/20120619'`
		`$ gpssh -f segments 'rm -r /data2/primary/gpseg*/db_dumps/20120619'`

Table 2 All Known Issues in 4.2.3

Issue	Category	Description
16267 15954	Management Scripts Suite	`gpkill` cannot kill processes that are deemed STUCK. **Workaround**: Kill the STUCK processes using OS kill.
16067	Management Scripts Suite	`gpkill` does not validate the user input for `password_hash_algorithm` The current behavior shows a success message for any input value. However, the server configuration parameter value is not updated if the input is invalid. When the user tries to set the value for a session from within psql, it fails with the appropriate error message.
16064	Backup and Restore	Restoring a compressed dump with the `--ddboost` option displays incorrect dump parameter information. When using `gpdbrestore --ddboost` to restore a compressed dump, the restore parameters incorrectly show "`Restore compressed dump = Off`". This error occurs even if `gpdbrestore` passes the `--gp-c` option to use gunzip for in-line de-compression.
15899	Backup and Restore	When running `gpdbrestore` with the list (`-L`) option, external tables do not appear; this has no functional impact on the restore job.

Upgrading to Greenplum Database 4.2.x.x

The upgrade path supported for this release is Greenplum Database 4.1.x.x to Greenplum Database 4.2.x.x. The minimum recommended upgrade path for this release is from Greenplum Database version 4.1.1.5. If you have an earlier major version of the database, you must first upgrade to version 4.1.x.

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

- Upgrading from 4.2.x.x to 4.2.3
- Upgrading from 4.1.x.x to 4.2.x.x
- For Users Running Greenplum Database 4.0.x.x
- For Users Running Greenplum Database 3.3.x.x
- Troubleshooting a Failed Upgrade

Upgrading from 4.2.x.x to 4.2.3

An upgrade from 4.2.x.x to 4.2.3 involves stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

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

   ```
   $ su - gpadmin
   ```

2. Perform a smart shutdown of your current Greenplum Database 4.2.x.x system (there can be no active connections to the database):

   ```
   $ gpstop
   ```

3. Run the installer for 4.2.3 on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example:

   ```
   /usr/local/greenplum-db-4.2.2.x
   ```

4. 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-4.2.x.x/greenplum_path.sh
   ```

   to:

   ```
   source /usr/local/greenplum-db-4.2.3.x/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-4.2.3.x
   /usr/local/greenplum-db
   ```

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

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

6. Run the `gpseginstall` utility to install the 4.2.3.x binaries on all the segment hosts specified in the *`hostfile`*. For example:

   ```
   $ gpseginstall -f hostfile
   ```

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 utilizing Data Domain Boost, you have to re-enter your DD Boost credentials after upgrading from Greenplum Database 4.2.1 to 4.2.3, as follows:

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

   Note that 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 4.1.x.x to 4.2.x.x

This section describes how you can upgrade from Greenplum Database 4.1.x.x or later to Greenplum Database 4.2.x.x. For users running versions prior to 4.1.x.x of Greenplum Database, see the following:

- For Users Running Greenplum Database 4.0.x.x
- For Users Running Greenplum Database 3.3.x.x

Planning Your Upgrade

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

Prior to upgrading your database, Greenplum 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
```

Some of the rules for partitioned tables are different in 4.2 than in previous releases. `gpmigrator` detects partitioned tables that violate these new rules and aborts the upgrade. In most cases, `gpmigrator` will create a repair script you can run to bring your 4.1 Greenplum Database into line with the new rules in 4.2. See Upgrading Partitioned Tables with Constraints for more details.

Upgrading Partitioned Tables with Constraints

Partition tables with `CHECK`, `PRIMARY KEY`, or `UNIQUE` constraints must be updated prior to upgrading:

- Regular `CHECK`, `PRIMARY KEY`, or `UNIQUE` constraints added by database users usually appear on every sub-table of the partitioned table. Their names may need to be adjusted. If necessary, `gpmigrator` creates a repair script to do this.
- `PRIMARY KEY` and `UNIQUE` constraints on partitioned tables that do not include all the columns of the partition key need to be removed. If needed, `gpmigrator` creates a repair script to do this. Note that the unique index underlying the constraint remains and provides the same protection against duplicate keys as did the constraint. As was previously the case, it is possible for different parts of the partitioned table to contain the same key value.
- If the added constraints appear on some but not all of the sub-tables that make up a partitioned table, they cannot be updated automatically. In this case, you need to either drop the irregular constraints or add the missing constraints. Other causes of irregularity may exist but are rare. Greenplum recommends that you contact support if you encounter any issues with partitioned tables that cannot be resolved automatically with `gpmigrator`.

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 4.1.x system)

Perform these steps on your current 4.1.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'
```

Note: 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 Software Binaries

1. Download or copy the installer file to the Greenplum Database master host.

2. Unzip the installer file. For example:

   ```
   # unzip greenplum-db-4.2.x.x-PLATFORM.zip
   ```

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

   ```
   # /bin/bash greenplum-db-4.2.x.x-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-4.2.x.x`), or enter an absolute path to an install location. You must have write permissions to the location you specify.

6. The installer will install the Greenplum software and create 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 4.2.x.x installation. For example:

   ```
   $ source /usr/local/greenplum-db-4.2.x.x/greenplum_path.sh
   ```

8. Run the `gpseginstall` utility to install the 4.2.3.x binaries on all the segment hosts specified in the *`hostfile`*. For example:

   ```
   $ gpseginstall -f hostfile
   ```

Upgrade Execution

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

9. Source the path file from your old 4.1.x.x installation. For example:

   ```
   $ source /usr/local/greenplum-db-4.1.1.5/greenplum_path.sh
   ```

10. (*optional but strongly recommended*) Back up all databases in your Greenplum Database system using `gpcrondump` (or zfs snapshots on Solaris systems). 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.

11. If your system has a standby master host configured, remove the standby master from your system configuration. For example:

```
$ gpinitstandby -r
```

12. Perform a clean shutdown of your current Greenplum system. For example:

```
$ gpstop
```

13. Source the path file from your new 4.2.x.x installation. For example:

```
$ source /usr/home/greenplum-db-4.2.x.x/greenplum_path.sh
```

14. Update your environment so it is sourcing your new 4.2.x.x installation.

a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.2.x.x installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.2.x.x
/usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

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

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.2.x.x
/usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```

15. (*optional but recommended*) Prior to running the migration, perform a pre-upgrade check to verify that your database is healthy by executing the 4.2.x.x version of the `gpmigrator` utility with the `--check-only` option. For example:

```
# gpmigrator_mirror --check-only
/usr/local/greenplum-db-4.1.1.5
/usr/local/greenplum-db-4.2.x.x
```

16. As `gpadmin`, run the 4.2.x.x version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ su - gpadmin
$ gpmigrator_mirror /usr/local/greenplum-db-4.1.1.5
/usr/local/greenplum-db-4.2.x.x
```

Note: If the migration does not complete successfully, contact Customer Support (see "Troubleshooting a Failed Upgrade" on page 17).

17. The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.2.x.x system will be running and accepting connections.

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

Post-Upgrade (on your 4.2.x.x system)

18. If your system had a standby master host configured, reinitialize your standby master using `gpinitstandby`:

```
$ gpinitstandby -s standby_hostname
```

19. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the compatible Greenplum Database 4.2.x.x Load Tools package. Application Packages are available at the EMC Download Center.

20. Rebuild any custom modules against your 4.2.x.x installation (for example, any shared library files for user-defined functions in `$GPHOME/lib`).

21. Greenplum Database 4.2.x.x introduced the `gppkg` utility to install Greenplum Database extensions. If you were previously using any PostgreSQL extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, download the corresponding packages from Powerlink, and install using this new utility. See the *Greenplum Database Administrator Guide 4.2* or later for usage details.

22. If you want to utilize the Greenplum Command Center management tool, install the latest Command Center Console and update your environment variable to point to the latest Command Center binaries ((source the `gpperfmon_path.sh` file from your new installation).

Note that the Greenplum Command Center management tool replaces Greenplum Performance Monitor.

Command Center Console packages are available from the EMC Download Center.

23. Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.2.x.x installation (if necessary).

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.1.x.x to 4.2.x.x. Detailed upgrade instructions are provided in the Upgrade Procedure section.

Pre-Upgrade Preparation (on your current system)	
* 4.1.x.x system is up and available	
□	Log in to your master host as the `gpadmin` user (your Greenplum superuser).
□	(Optional) Run `VACUUM` on all databases,
□	(Optional) Remove old server log files from `pg_log` in your master and segment data directories.
□	Check for and recover any failed segments (`gpstate`, `gprecoverseg`).
□	Copy or preserve any additional folders or files (such as backup folders).
□	Install the Greenplum Database 4.2.x.x binaries on all Greenplum hosts.
□	Inform all database users of the upgrade and lockout time frame.
Upgrade Execution	
* The system will be locked down to all user activity during the upgrade process	
□	Backup your current databases.
□	Remove the standby master (`gpinitstandby -r`).
□	Do a clean shutdown of your current system (`gpstop`).
□	Update your environment to source the new Greenplum Database 4.2.x.x installation.
□	Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
□	After the upgrade process finishes successfully, your 4.2.x.x system will be up and running.
Post-Upgrade (on your 4.2.x.x system)	
* The 4.2.x.x system is up	
□	Reinitialize your standby master host (`gpinitstandby`).

□	Upgrade `gpfdist` on all of your ETL hosts.
□	Rebuild any custom modules against your 4.2.x.x installation.
□	Download and install any Greenplum Database extensions.
□	(Optional) Install the latest Command Center Console and update your environment to point to the latest Command Center binaries.
□	Inform all database users of the completed upgrade.

For Users Running Greenplum Database 4.0.x.x

Users on a release prior to 4.1.x.x cannot upgrade directly to 4.2.x.x.

- 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 Powerlink).
- Follow the upgrade instructions in these release notes for Upgrading from 4.1.x.x to 4.2.x.x.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.1.x.

- Upgrade from your current release to 4.0.x.x (follow the upgrade instructions in the latest Greenplum Database 4.0.x.x release notes available on Powerlink).
- Upgrade the 4.0.x.x release to 4.1.x.x (follow the upgrade instructions in the latest Greenplum Database 4.1.x.x release notes available on Powerlink).
- Follow the upgrade instructions in these release notes for Upgrading from 4.1.x.x to 4.2.x.x.

Troubleshooting a Failed Upgrade

If you experience issues during the migration process, go to the Support page at Powerlink or contact Greenplum customer support at one of the following numbers:

United States: 800-782-4362 (1-800-SVC-4EMC)

Canada: 800-543-4782

Worldwide: +1-508-497-7901

Be prepared to provide the following information:

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

Greenplum Database Tools Compatibility

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.

Note: Release and support for AIX Clients/Connectivity/Loaders is pending. We will update the Release Notes and the EMC Download Center when available.

Tool packages are available from the EMC Download Center.

Table 3 Greenplum Database Tools Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql) Greenplum MapReduce (gpmapreduce)[1]	4.2.3.0	4.2.3.0
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.2.3.0	4.2.3.0
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.2.2.0	4.2.3.0
Greenplum Command Center	Greenplum Database management tool.	1.2	4.2.3.0

[1] gpmapreduce is not available on Windows.

Greenplum Database Extensions Compatibility

Greenplum Database delivers an agile, extensible platform for in-database analytics, leveraging the system's massively parallel architecture. With Release 4.2.x.x, Greenplum enables turn-key in-database analytics via Greenplum Extensions.

You can download Greenplum extensions packages from the EMC Download Center and install them using the Greenplum Packager Manager (`gppkg`). See the *Greenplum Database Administrator Guide 4.2* or later for details.

Note that Greenplum Package Manager installation files for extension packages may release outside of standard Database release cycles. Therefore, for the latest install and configuration information regarding any supported database package/extension, go to the `Support` site and download `Primus Article 288189` from our knowledge base.

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

Note that the PL/Python database extension is already included with the standard Greenplum database distribution.

Table 4 Greenplum Database Extensions Compatibility

Greenplum Database Extension	Extension Components	
	Name	**Version**
PostGIS 1.0 for Greenplum Database 4.2.x.x	PostGIS	1.4.2
	Proj	4.7.0
	Geos	3.2.2
PL/Java 1.0 for Greenplum Database 4.2.x.x	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26 Update 31
PL/R 1.0 for Greenplum Database 4.2.x.x	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.1 and 1.2 for Greenplum Database 4.2.x.x	PL/Perl	Based on PostgreSQL 9.1
	Perl	5.12.4
Pgcrypto 1.0 for Greenplum Database 4.2.x.x	Pgcrypto	Based on PostgreSQL 8.3
Greenplum Hadoop File System	gphdfs	1.1
	gphdfs	1.2
MADlib 1.0 for Greenplum Database 4.2.x.x	MADlib	0.5.0

Hadoop Distribution Compatibility

Use the `gppkg` utility to install the gNet package containing the jar file for the extensions, the libraries, and the documentation for the gphdfs extensions. To install the correct distribution, refer to the following Hadoop extensions compatibility matrix:

Table 5 Hadoop Extensions Compatibility

Hadoop Distribution	Version
Greenplum HD	Greenplum HD 1.1
	Greenplum HD 1.2
Cloudera	cdh302
Greenplum MR	Greenplum MR 1.0
	Greenplum MR 1.2

Greenplum Database 4.2 Documentation

For the latest Greenplum Database documentation go to Powerlink. Greenplum documentation is provided in PDF format.

Note that our documentation set has been restructured with this release, see "4.2.3 Documentation Set" on page 2 for more details.

Table 6 Greenplum Database Documentation

Title	Revision
Greenplum Database 4.2.3 Release Notes	A01
Greenplum Database 4.2 Installation Guide	A04
Greenplum Database 4.2 Database Administrator Guide	A01
Greenplum Database 4.2 System Administrator Guide	A01
Greenplum Database 4.2 Reference Guide	A01
Greenplum Database 4.2 Utility Guide	A01
Greenplum Database 4.2 Client Tools for UNIX	A01
Greenplum Database 4.2 Client Tools for Windows	A01
Greenplum Database 4.2 Connectivity Tools for UNIX	A01
Greenplum Database 4.2 Connectivity Tools for Windows	A01
Greenplum Database 4.2 Load Tools for UNIX	A01
Greenplum Database 4.2 Load Tools for Windows	A01
Greenplum Command Center 1.2 Administrator Guide	A01