



P/N: 300-013-603

Rev: A01

Updated: February 29, 2012

Welcome to Greenplum Database 4.2.1

Greenplum Database is a massively parallel processing (MPP) database server that supports next generation data warehousing and large-scale analytics processing. By automatically partitioning data and running parallel queries, it allows a cluster of servers to operate as a single database super computer 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, to multiple petabytes.

Note: This document contains pertinent release information about Greenplum Database 4.2.1. For previous versions of the release notes for the Greenplum database, go to [Powerlink](#).

About Greenplum Database 4.2.1

Greenplum Database 4.2.1 is the first service pack release for version 4.2. This service pack release includes new issue resolutions and minor feature enhancements. Please refer to the following sections for more information about this release.

- [Supported Platforms](#)
- [New Functionality in Greenplum Database 4.2.1](#)
- [Resolved Issues in Greenplum Database 4.2.1](#)
- [Known Issues in Greenplum Database 4.2.1](#)
- [Upgrading to Greenplum Database 4.2.x.x](#)
- [Greenplum Database Client Tool Packages and Performance Monitor](#)
- [Greenplum Database Extensions Compatibility](#)
- [Hadoop Distribution Compatibility](#)
- [Greenplum Database 4.2, Documentation](#)

Supported Platforms

Greenplum Database 4.2.1 runs on the following platforms:

- Red Hat Enterprise Linux 64-bit 5.5, 5.6, 5.7, 6.1
- 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

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 Interoperability Navigator, then search for Greenplum Database.

New Functionality in Greenplum Database 4.2.1

The following new functionality is being introduced in this release of Greenplum Database. See the Greenplum Database documentation for more details about each feature. For a complete list of relevant documentation, see [Greenplum Database 4.2, Documentation](#).

Column-level Compression for Append-only Tables

In previous releases, Greenplum Database provided support for table-level compression for append-only tables, which allows the user to specify a compression algorithm to be specified for the entire table object. In 4.2.1, Greenplum Database introduces support for column-level compression, where users can specify different compression algorithms for different columns in the same table object. The data stored in columns is of uniform data type, and thus several storage optimizations are available for column-oriented data which are not available for row-oriented data. For example, a column of data type Month has a cardinality of 12, and certain algorithms such as RLE (Run Length Encoding) make use of similarity of adjacent data to compress data sets.

New Column-level Compression Algorithm

In 4.2.1 Greenplum Database provides support for RLE (Run Length Encoding) algorithm. RLE is particularly suited for column compression. RLE achieves maximum compression on data with long runs of duplicate elements, for example: A 'date' column values as date '1/1/01' repeated 100 times in a column will be stored as "1/1/01 100". A best practice while using RLE is to sort data on input to achieve the maximum compression.

SSL Support for Parallel Data Loading

In 4.2.1, Greenplum Database introduces support for SSL encryption for gpfdist - Greenplum's Parallel File Distribution server. Customers often load data into Greenplum environments from other encrypted data sources. With SSL Support for gpfdist, customers data can be encrypted in transit between the gpfdist server and Greenplum Database segment hosts.

New management utility gckill

In 4.2.1, Greenplum Database provides a new utility, `gckill`, which protects end-users from accidentally killing key Greenplum Database processes and bringing their clusters down.

New password encryption algorithm SHA-256

In 4.2.1, Greenplum Database provides support for a new algorithm SHA-256 for encrypting passwords stored in the database. SHA-256 algorithm ensures compliance with FIPS-140-2 standard required by Federal Customers. The Federal Information Processing Standard Publication 140-2 is a U.S. government computer security standard used to accredit cryptographic modules. The SHA-256 algorithm is provided by RSA BSAFE toolkit bundled with the Greenplum Database.

Secure Connection between GPDB and Hadoop

In 4.2.1, Greenplum Database supports the capability of a secure connection with Hadoop. gNet™ Hadoop extensions can be configured to ensure that customer data is transferred in an encrypted fashion between Hadoop and Greenplum Database.

Certification of Greenplum gNet Hadoop Extensions with additional Hadoop distributions

Greenplum Database 4.2.1 release will certify `gp_hdfs` protocol for the following Hadoop distributions, the following are available from the [EMC Download Center](#).

- Greenplum HD: GPHD 1.0, GPHD 1.1,
- Greenplum MR: GPMR 1.0, GPMR 1.2

Note: You can also use Cloudera `cdh3u2`

Enhanced Server Configuration Parameter

To connect to a Greenplum Database 4.2.0, customers had to specify the correct Hadoop extension for the distribution. They could accomplish this using the server configuration parameter, `gp_hadoop_connector_version`. Greenplum Database 4.2.1 provides an enhancement for this server configuration parameter configuration parameter. The parameter requires that the user specify the preferred Hadoop distribution:

Table 1 Enhanced Server Configuration Parameters in 4.2

Parameter Name	Value Range	Default Value	Description	Set Classifications
<code>gp_hadoop_target_version</code>	gphd-1.0 gphd-1.1 gpmr -1.0 gpmr -1.2 cdh3u2	gphd-1.1	This parameter specifies the target Hadoop distribution.	master session reload

Greenplum Package Manager support for MADlib

Greenplum Package Manager (`gppkg`) ensures automatic installation and updates of functional extensions like in-database GeoSpatial functions, PL/R, PL/Java, PL/Perl and `pgcrypto`, along with their dependencies, across an entire cluster. The package

manager also integrates with existing scripts so that any packages are automatically installed on any new hosts introduced into the system following cluster expansion or segment host recovery.

In 4.2.1, the `gppkg` utility provides support for deploying MADlib in a Greenplum Database MPP cluster. MADlib is an open-source library for scalable in-database analytics. It provides data-parallel implementations of mathematical, statistical and machine learning methods for structured and unstructured data. For more information, see <http://madlib.net>. The database extension for MADlib is available from the [EMC Download Center](#).

Greenplum Database 4.2 - Extensions - MADlib 1.0

- MADlib version 0.3.0

Resolved Issues in Greenplum Database 4.2.1

Table 2, “Resolved Issues in 4.2.1 (previously documented as Known Issues)” lists customer reported issues that are now resolved in Greenplum Database 4.2.1.

Note: This document contains pertinent release information about Greenplum Database 4.2.1. For previous versions of the release notes of the Greenplum database, go to the [Powerlink](#) home page.

The following Resolved Issues were documented as Known Issues in earlier release notes.

Table 2 Resolved Issues in 4.2.1 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
16474 16480	4.2.1.0	Query Optimizer	An issue in the query planner occasionally yielded unexpected results. This issue has been resolved in this release.
16107 15208	4.2.1.0	SQL Parser	In 4.2, the upgrade routine did not check to see if the database had been stopped before proceeding, causing multiple errors. This issue has been resolved in this release.
15967 16056	4.2.1.0	Catalog and Metadata	Concurrent truncate operations may lead to tuple errors. Systems running concurrent truncate operations may have experienced tuple errors in persistent tables, such as TID persistent relation file <filename> tuple is invalid. This issue has been resolved in this release.
15962, 16009	4.2.1.0	DDL and Utility Statements	When you DROP a parent partition index, the child partition index does not DROP as well. Now, when you drop an index on a parent partition, you see the following warning: <code>WARNING: Only dropped the index "i1"</code> <code>HINT: To drop other indexes on child partitions, drop each one explicitly.</code> This issue has been resolved in this release.
15816, 15479	4.2.1.0	DDL and Utility Statements	Attempting to change user parameters with non-numerical statements such as '150MB' errored out. This issue has been resolved in this release; non-numerical expressions now perform as expected.

Table 2 Resolved Issues in 4.2.1 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
15815 15205	4.2.1.0	Query Optimizer	No available memory In very rare circumstances, the Greenplum Performance Monitor process <code>gpmonws</code> used all available memory on the master node. This issue has been resolved in this release.
15761 15762	4.2.1.0	None	Performance monitor now monitors ETL and Hadoop nodes in its Health Monitor The health monitoring software and Performance Monitor can now monitor DIA (ETL) nodes and Hadoop nodes. DIA and Hadoop nodes are options you can include in a DCA, in addition to the standard Greenplum Database nodes. This issue has been resolved in this release.
15662 15642	4.2.10	Upgrade/ Downgrade	Previously, the upgrade path from 4.1.1.7 to 4.2.0.0 did not work. This issue has been resolved in this release.
15620 15640	4.2.1.0	None	SAS; guc.h should not include the header file, emconnect/api.h This issue has been resolved in this release.
15478 15714	4.2.1.0	None	Greenplum was not handling transitive closures of non-equal conditions. This issue has been resolved in this release.
15316 14770	4.2.1.0	DDL and Utility statements	At times expansions failed when a distributed column was not unique in a table. This issue has been resolved in this release.
15303 13491	4.2.1.0	Catalog and Metadata	Trying to insert more than approximately 400,000 rows into a message column causes errors. This issue has been resolved in this release.
15222 15441	4.2.1.0	Catalog and Metadata	Dropping the grantor role causes gpcheckcat to report foreign key errors This issue has been resolved in this release.
15221 15588	4.2.1.0	Replication: Crash Recovery	Rebooting the server while the Greenplum database is still running maintenance tasks that include vacuum prevents the database from starting up after the reboot This issue has been resolved in this release.
15208 16073	4.2.1	SQL Parser	The upgrade routine did not check that the database had been stopped. This caused errors. This issue has been resolved in this release.
15205 15815	4.2.1.0	Resource Management	The ALTER RESOURCE QUEUE command does not update the resource queue property, COST_OVERCOMMIT. This issue has been resolved in this release.
15103 15342	4.2.1.0	None	At times temp files were not cleaned after a database restart. This issue has been resolved in this release.
14704 15365	4.2.1.0	OS Abstraction	Stuck spinlock caused by issuing a COPY statement too frequently This issue has been resolved in this release.
14362 15587	4.2.1.0	Management Scripts Suite	During some upgrades, gpexpand reported that it ran analyze on tables when it had not. This issue has been resolved in this release.

Table 2 Resolved Issues in 4.2.1 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
14329 15374	4.2.1.0	Functions and Languages	Some users experienced slow query times when issuing via a SQL function. This issue has been resolved in this release.
12365 14813 15376 15417	4.2.1.0	DDL/DML: Heap, DML	Database sizing metric based on pg_database_size provides unexpected results. This issue has been resolved in this release.
13987 15248	4.2.1.0	DDL/DML: AO - row	If the Greenplum database is running at high concurrency while migrating an application from the Sy base IQ, the Greenplum System crashes This issue has been resolved in this release
11377 15312	4.2.1.0	DML	In previous versions, SAS unions would fail because columns were treated as different data types (text and integer) This issue has been resolved in this release.

Known Issues in Greenplum Database 4.2.1

This section lists the new known issues in Greenplum Database 4.2.1. 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 3 All Known Issues in 4.2.1

Issue	Category	Description
16577	Upgrade and Downgrade	<p>gpmigrator --check-only option is not documented in the online help or in the <i>Greenplum Database Administrator Guide</i>, or later</p> <p>Prior to upgrading your Greenplum Database, Greenplum recommends that you run a pre-upgrade check to verify that your database is healthy. You can perform a pre-upgrade check by executing the new gpmigrator (_mirror) utility with the --check-only option.</p> <p>For example:</p> <pre>source \$new_gphome/greenplum_path.sh; gpmigrator_mirror --check-only \$old_gphome \$new_gphome</pre>
16556 16321 16129	Management Scripts Suite	<p>gpkill does not run on the Solaris platform.</p> <p>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.</p>
16546	Documentation	<p>Need to add the following changes to the Backup and Restore sections of the <i>Greenplum Database Administrator Guide</i>, Revision A02</p> <ul style="list-style-type: none"> • Using the Data Domain Boost: The gpccrondump functionality was designed for one network connection to the Data Domain system. If the customer is using two or more connections, you will need to change the host name manually on the segment servers in the .ddconfig file. • Configuring Advanced Load Balancing and Link Failover in Data Domain: The Data Domain group should only be used for IPs on the same network subnet. • Enabling Encrypted Optimized Duplication: We only support Data Domain Collection replication in 4.2.0 and 4.2.1. The Low-Bandwidth option does not work with Collection replication for DDBoost. • Incorrect gpccrondump option in the section Backup Options for Data Domain Boost and Local Backups: The correct option is -o. This section lists one of the gpccrondump options as -s, which only works for schema backups.
16520	Documentation	<p>gpkill can accept a list of pids, not just a single pid</p> <p>The description of gpkill in the <i>Greenplum Database Administrator Guide</i>, or later is not complete. The information should be as follows:</p> <p>The gpkill utility checks or terminates a Greenplum process or a list of Greenplum processes. If the process is a critical Greenplum Database process or a system process (not part of Greenplum), gpkill does not terminate it. If you are a Superuser, you can only use gpkill to terminate your own processes.</p>

Table 3 All Known Issues in 4.2.1

Issue	Category	Description
16515 13669	Documentation	The section, “Creating Queues with Memory Limit”, needs clarification The MEMORY_LIMIT setting is determined on a per-segment basis, not on a per-segment-host basis. The information should be edited as follows: “Resource queues that offer a MEMORY_LIMIT setting, control the amount of memory for all the queries submitted through the queue. The total memory should not exceed the physical memory available per-segment. Greenplum recommends that you set MEMORY_LIMIT to 90% of memory available on a per-segment basis. For example, if a host has 48 GB of physical memory and 6 segments, then the memory available per segment is 8 GB. You can calculate the recommended MEMORY_LIMIT for a single queue as 0.90*8=7.2 GB. If there are multiple queues created on the system, their total memory limits must also add up to 7.2 GB.”
16425 15886	Query Execution	Queries may fail when run within user-defined functions, even with very high memory statement settings You may see the error message: 56: ERROR: insufficient memory reserved for statement Workaround: To turn off memory quotas before running queries within UDFs, set gp_resqueue_memory_policy to none.
16421	Performance	Setting the server parameter values for max_connections and max_prepared_connections to high value creates performance degradation These new server configuration parameters are only set during gpinitssystem, and an upgrade to 4.2.x does not change it. Existing 4.1.x customers will not see a change in behavior. Also impacts autostats.
16402	Loader: gpfdist	gpfdist_ssl: cannot upload big files from the ETL server running on different platform (Linux) to the Greenplum cluster running on Solaris When running gpfdist with the --ssl option on Linux, while the master and segments running on Solaris, the gpfdist returns an error.
16310	Functions and Languages	plpython cannot handle the SPI exception correctly If plpython encounters the SPI exception it aborts the transaction and discards the changes. However the error message does not indicate this.
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.
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	gpconfig 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	Cannot restore a compressed dump with the --ddbboost option When using gpdbrestore --ddbboost 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.
15903	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.

Table 3 All Known Issues in 4.2.1

Issue	Category	Description
15769	Documentation	Data Domain Boost recommends new sizing recommendations For detailed sizing recommendations when using Data Domain Boost, contact Data Domain customer support or go to https://my.datadomain.com .
15719	Documentation	Changes related to the file replication feature in GPDB The Greenplum Database Administrator Guide states that when using the UPDATE and DELETE command the following restriction applies: "Cannot use STABLE or VOLATILE functions in UPDATE (or DELETE) statement if mirrors are enabled." This is not correct.
15458	Replication - Master Mirroring	Documentation not clear about -t and -n option for pg_dump The current description of the -t option is as follows: -t, --table=TABLE dump the named table(s) only This description should be changed to -t, --table=TABLE dump the named table(s) only, schema.tablename.

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 version of the database, you must first upgrade to version 4.1.x.

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

- [Planning Your Upgrade](#)
- [Upgrading from 4.2.0 to 4.2.1.x](#)
- [For Users Running Greenplum Database 4.0.x.x](#)
- [For Users Running Greenplum Database 3.3.x.x](#)
- [Troubleshooting a Failed Upgrade](#)

Planning Your Upgrade

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

Prior to upgrading your Greenplum Database, we recommend that you run a pre-upgrade check to verify that your database is healthy.

You can perform a pre-upgrade check by executing the new `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 that 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](#), below, 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 have to be adjusted. If needed, `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. We recommend that you contact support if you encounter any issues with partitioned tables that cannot be resolved automatically with `gpmigrator`.

Upgrading from 4.2.0 to 4.2.1.x

These instructions are for users currently running 4.2.0. An upgrade from 4.2.0 to 4.2.1 involves removing the standby master from your Greenplum system (if configured), 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 - gadmin
```

If you have a standby master configured, remove it from your Greenplum configuration:

```
$ gpinitstandby -r
```

2. Perform a smart shutdown of your current Greenplum Database 4.2.0 system (there can be no active connections to the database):

```
$ gpstop
```

3. Run the installer for 4.2.1.x on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example:

```
/usr/local/greenplum-db-4.2.1.x
```

4. Edit the environment of the Greenplum Database superuser (gadmin) 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-4.2.0.0/greenplum_path.sh
to:
```

```
source /usr/local/greenplum-db-4.2.1.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-4.2.1.x
/usr/local/greenplum-db
```

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


```
$ source ~/.bashrc
```
6. Log in as root and install the 4.2.1.x binaries on all of the segment hosts. Tar the Greenplum installation directory, use `gpscp` to copy it to the segment hosts, and use `gpssh` to untar it at the segment hosts. Make sure that the `gadmin` user owns the new installation directory. Also update any symbolic links to point to the new version. For example:

```
$ su -
# gtar -cvf gp.tar /usr/local/greenplum-db-4.2.1.x
# gpscp -f seghostname_file gp.tar =:/
# gpssh -f seghostname_file
=> gtar --directory / -xvf /gp.tar
=> chown -R gadmin /usr/local/greenplum-db-4.2.1.x
=> rm /gp.tar
=> rm /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.2.1.x
/usr/local/greenplum-db
```

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

```
$ su - gadmin
$ gpstart
```

8. If your original system had a standby master, add the standby master back into your Greenplum configuration:

```
$ gpinitstandby -s standby_hostname
```

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](#)

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 also provided in the [Upgrade Procedure](#) section.

Pre-Upgrade Preparation (on your current system)	
<i>* 4.1.x.x system is up and available</i>	
<input type="checkbox"/>	Log in to your master host as the <code>gpadmin</code> user (your Greenplum superuser).
<input type="checkbox"/>	Check for and recover any failed segments (<code>gpstate</code> , <code>gprecoverseg</code>).
<input type="checkbox"/>	Install the Greenplum Database 4.2.x.x binaries on all Greenplum hosts.
<input type="checkbox"/>	Copy or preserve any additional folders or files (such as backup folders) that you have added in the Greenplum data directories or <code>\$GPHOME</code> directory. Only files or folders strictly related to Greenplum Database operations are preserved by the migration utility.
<input type="checkbox"/>	(Optional) Run <code>VACUUM</code> on all databases, and remove old server log files from <code>pg_log</code> in your master and segment data directories. This is not required, but will reduce the size of Greenplum Database files to be backed up and migrated.
<input type="checkbox"/>	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.
Upgrade Execution	
<i>* The system will be locked down to all user activity during the upgrade process</i>	
<input type="checkbox"/>	Backup your current databases (<code>gpcrondump</code> or ZFS snapshots) and secure backup files in a location outside of your Greenplum data directories.
<input type="checkbox"/>	Remove the standby master from your system configuration (<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 your Greenplum Database 4.2.x.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 4.2.x.x system will be up and running.
Post-Upgrade (on your 4.2.x.x system)	
<i>* The 4.2.x.x system is up</i>	

<input type="checkbox"/>	Reinitialize your standby master host (<code>gpinitstandby</code>).
<input type="checkbox"/>	Upgrade <code>gpfdist</code> on all of your ETL hosts by installing the compatible version 4.2.x.x Load Tools package.
<input type="checkbox"/>	Rebuild any custom modules against your 4.2.x.x installation (for example any shared library files for user-defined functions in <code>\$GPHOME/lib</code>).
<input type="checkbox"/>	Greenplum Database 4.2.x.x introduced the <code>gppkg</code> utility to install Greenplum Database extensions. If you were previously using any PostgreSQL extensions such as <code>pgcrypto</code> , <code>PL/R</code> , <code>PL/Java</code> , <code>PL/Perl</code> , and <code>PostGIS</code> , download the corresponding packages from Powerlink , and install using this new utility. See the <i>Greenplum Database Administrator Guide 4.2</i> or later for usage details.
<input type="checkbox"/>	Install the new 4.1.1.x Performance Monitor Web Application and update your environment to point to the 4.1.1.x Performance Monitor binaries. See “ Greenplum Database Client Tool Packages and Performance Monitor ” on page 17 for detailed compatibility information.
<input type="checkbox"/>	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 Procedure

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



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 (`gadmin`).

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

```
$ su - gadmin
```
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 all existing log files from your segment hosts:

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

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. Create a `hostfile` file that has the machine configured host names and host addresses (interface names) for each host in your Greenplum system (master, standby master and segments). Make sure there are no blank lines or extra spaces. For example, if you have a master, standby master and three segments with two network interfaces per host, your file would look something like this:

```
mdw
mdw-1
mdw-2
smdw
smdw-1
smdw-2
sdw1
sdw1-1
sdw1-2
sdw2
sdw2-1
sdw2-2
sdw3
sdw3-1
```

```
sdw3-2
```

8. 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
```
9. Run the `gpsegininstall` utility referencing the `hostfile` file you just created. Use the `-u` and `-p` options to specify the name and password of your Greenplum administrative user (`gpadmin`). For example:


```
$ gpsegininstall -f hostfile -u gpadmin -p P@$$word
```

Upgrade Execution

During upgrade, all client connections to the master will be locked out.

10. 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
```
11. (*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.
12. If your system has a standby master host configured, remove the standby master from your system configuration. For example:


```
$ gpinitstandby -r
```
13. Perform a clean shutdown of your current Greenplum system. For example:


```
$ gpstop
```
14. 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
```
15. Update your environment so that 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
```

- 16.** (*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.
- 17.** As `gppadmin`, 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 - gppadmin
$ 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).
- 18.** 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)

- 19.** If your system had a standby master host configured, reinitialize your standby master using `gpinitstandby`:
- ```
$ gpinitstandby -s standby_hostname
```
- 20.** 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](#).
- 21.** Rebuild any custom modules against your 4.2.x.x installation (for example any shared library files for user-defined functions in `$GPHOME/lib`).
- 22.** 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.
- 23.** If you are using the Greenplum Performance Monitor, install the latest 4.1.1.x Performance Monitor Web Application and update your environment to point to the new 4.1.1.x Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.2.x.x installation). Greenplum Performance Monitor 4.1.1.x Web Application Packages are available at the [EMC Download Center](#).

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 4.2.0 to 4.2.1.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 4.2.0 to 4.2.1.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 Checklist](#).
- Log output from `gpmigrator` and `gpcheckcat` (located in `~/gpAdminLogs`)

Greenplum Database Client Tool Packages and Performance Monitor

Greenplum releases the Greenplum Database Performance Monitor and a number of client tool packages on various platforms that can be used to connect to Greenplum Database. The following table describes the Performance Monitor and client tool packages compatible with this Greenplum Database release. Client tool packages are available at the [EMC Download Center](#).

Table 4 Greenplum Database Performance Monitor and Client Tools Version Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql) Greenplum MapReduce (gpmapreduce) ¹	4.2.1.0	4.2.1.0
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.2.1.0	4.2.1.0
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.2.1.0	4.2.1.0
Greenplum Database Performance Monitor	Greenplum Database monitor for query and system performance	4.1.2.0	4.2.1.0

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.

Greenplum Extensions can be downloaded from [EMC Download Center](#) and installed using the new `gppkg` utility. See the *Greenplum Database Administrator Guide 4.2* or later for details.

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 5 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
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 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	1.0
	gphdfs	1.1
MADlib 1.0 for Greenplum Database 4.2.x.x	MADlib	0.3.0

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 6 Hadoop Extensions Compatibility

Hadoop Distribution	Version
Greenplum HD	Greenplum HD1.0
	Greenplum HD1.1
Cloudera	cdh302
Greenplum MR	Greenplum MR 1.0
	Greenplum MR 1.2

Greenplum Database 4.2, Documentation

For the latest Greenplum Database documentation, at the time of publishing, go to [Powerlink](#). Greenplum documentation is provided in PDF format.

Table 7 Greenplum Database Documentation

Title	Part Number	Revision
Greenplum Database 4.2 Installation Guide	300-013-162	A02
Greenplum Database 4.2 Administrator Guide	300-013-163	A02
Greenplum Database 4.2 Client Tools for UNIX	300-013-190	A01
Greenplum Database 4.2 Client Tools for Windows	300-013-191	A01
Greenplum Database 4.2 Connectivity Tools for UNIX	300-013-192	A01
Greenplum Database 4.2 Connectivity Tools for Windows	300-013-193	A01
Greenplum Database 4.2 Load Tools for UNIX	300-013-194	A01
Greenplum Database 4.2 Load Tools for Windows	300-013-195	A01
Greenplum Database 4.2.1 Release Notes	300-013-603	A01
Greenplum Database 4.2 Release Notes	300-013-265	A01
Greenplum Database Performance Monitor 4.1 Administrator Guide	300-012-430	A04

Copyright © 2012 EMC Corporation. All rights reserved.

EMC 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." EMC CORPORATION 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 EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

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