Greenplum
The Data Computing Division of EMC

EMC²
where information lives®

P/N: 300-013-261
Rev: A01
Updated: November 23, 2011

Welcome to Greenplum Database 4.2

Greenplum Database is a massively parallel processing (MPP) database server designed to support the next generation of data warehousing and large-scale analytics processing. It allows a cluster of servers to operate as a single database super computer — automatically partitioning data and parallelizing queries — to achieve performance tens or hundreds times faster than traditional databases. It supports SQL and MapReduce parallel processing and data volumes that range from hundreds of gigabytes, to tens to hundreds of terabytes, to multiple petabytes.

About Greenplum Database 4.2

Greenplum Database 4.2 introduces a number of significant new features, as well as performance and stability enhancements.

Please refer to the following sections for more information about this release.

- Supported Platforms
- New Functionality in Greenplum Database 4.2
- Changed Features in Greenplum Database 4.2
- Resolved Issues in Greenplum Database 4.2
- Known Issues in Greenplum Database 4.2
- Upgrading to Greenplum Database 4.2
- Greenplum Database Client Tool Packages and Performance Monitor
- Greenplum Database Extensions Compatibility
- Greenplum Database 4.2 Documentation

Release Notes for earlier releases of Greenplum Database are available on Powerlink.

Supported Platforms

Greenplum Database 4.2 is supported 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
- Oracle 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 Interoperatability Navigator, then search for Greenplum Database.

New Functionality in Greenplum Database 4.2

The following new functionality is being introduced in this release of Greenplum Database. See the *Greenplum Database Administrator Guide 4.2* for more details about each feature.

High Performance Hadoop Connector

Greenplum Database 4.2 Hadoop Connector enables high performance parallel import and export of compressed and uncompressed data from Hadoop clusters.

To further streamline resource consumption during load times, custom-format data (binary, Pig, Hive, etc.) in Hadoop can now be converted to Greenplum Database Format using MapReduce, and then imported into Greenplum Database. This provides significantly more efficient and flexible data exchange between Hadoop and Greenplum Database.

Greenplum Database 4.2 Hadoop Connectors are available for GP HD, GP MR.

Simpler and scalable back-up with Data Domain Boost

Greenplum Database achieves enterprise level backup through the integration with Data Domain using Data Domain Boost. The Database is now Data Domain aware and performs backup deduplication in each database segment. As a result, it only sends the changed data to Data Domain for backup. This dramatically increases throughput, reduces the data transferred over the network on subsequent backups, and increases the efficiency of back-up servers CPU utilization. Furthermore, with the use of Data Domain Boost, it eliminates the need for NFS mount management.

Greenplum backup using Data Domain Boost supports Data Domain Collection replication.

Greenplum backup using Data Domain Boost also supports simultaneous backup and restore when the backup and restore operations run on different databases.

For sizing recommendations, please contact Data Domain customer support at `http://www.datadomain.com/support/contact-support.html`.

Limitations:

- Only 1:1 (one Greenplum Database to one Data Domain system) configuration is allowed for backup and restore using Data Domain Boost.
- Backup and restore using Data Domain Boost is available:

  —to Data Domain 890 if Greenplum Database does not exceed 180 primary segment instances plus the Master.

  —to Data Domain 670, 860, and 880 if Greenplum Database does not exceed 90 primary segment instances plus the Master.
- Greenplum Backup and Restore with Data Domain Boost does not support Data Domain Global Deduplication Array (GDA) and Data Domain Archiver.
- Supported Platforms: Greenplum currently does not support backup and restore using Data Domain Boost on Solaris.

  Please consult the Data Domain Compatibility Guide to confirm Data Domain Boost host OS support for your Greenplum host platform.

  The Data Domain Compatibility Guide is available here: `https://my.datadomain.com/US/en/compatibility.jsp`

Language and Compatibility Enhancements for Faster Migrations to Greenplum

Greenplum Database offers enhanced SQL support, including native support of more than 20 Oracle functions, correlated subqueries, non-recursive `WITH` clause, and fixed-width data format. These enhancements streamline support of 3rd party tools that generate such queries and make migration from other databases faster and simpler.

- **Oracle Functions**

  Addition of more than 20 Oracle-compatible functions.
- **Correlated Subqueries Support**

  Correlated subqueries provide an efficient way to express results in terms of results of another query.

  Correlated subqueries are already widely used.
- **WITH Clause (Common Table Expression) Support**

  Non-recursive `WITH` clauses are supported to define one or more subqueries that can be used in a query. Common Table Expressions are part of SQL standard.

  Benefits:

  Enables you to reference the same subquery in a query without repeating the same subquery text.

  Enables you to break complex queries into multiple pieces

  Avoids the need to rewrite `WITH` queries

  Limitations:

  Greenplum Database only supports the *non-recursive* `WITH` clause.

- **Fixed-Width Format Support**

  Greenplum Database can read fixed-width format files using the external table functionality.

  Benefits:

  Data files with fixed-width format allow for quick parsing.

  No longer need an intermediary utility to transfer data in a fixed-width format to a Greenplum Database supported format.

  External Table extensions allow for Oracle fixed-width format data files to be exported seamlessly to the Greenplum Database.

- **XML Support**

Greenplum Database 4.2 includes the following XML support:

XML Functions—Greenplum Database 4.2 includes support for XML data type and the XML Path language (`xpath`) version 1.0, enabling high performance parallel load of XML documents into the database. The following XML support is included:

- XML data type

  ```
  xml(text)  RETURNS xml
  ```

- XPath Expressions

  ```
  xpath(text, xml, text array)   RETURNS xml array
  xpath_exists(text, xml, text array)  RETURNS bool
  xpath(text, xml) RETURNS xml array
  ```

- XML Support Functions

  ```
  xml_is_well_formed_document(text)  RETURNS bool
  xmlagg(xml) RETURNS xml
  xmlconcat2(xml, xml) RETURNS xml
  ```

- Indexes on XML Metadata for faster retrieval

XML Transformation—Greenplum Database data loader *gpfdist* now provides transformation features to load XML data into a table and to write data from the Greenplum Database to XML files.

Performance and Memory Optimizations

Greenplum Database Release 4.2 supports dynamic partition elimination and query memory optimization. Dynamic partition elimination disregards irrelevant partitions in a table and allows for significant reduction in amount of data scanned and results in faster query execution times. The query memory optimization feature intelligently frees and reallocates memory to different operators during query processing, allowing for better memory utilization, higher throughput, and higher concurrency.

- **Dynamic Partition Elimination**

  Performance of queries with equijoins on the partition key and restrictions on other dimensions is improved by pruning partitions with no matching data.

- **Memory Optimization**

  Greenplum Database's use of eager-free memory optimization results in better concurrency:

  In Greenplum Database 4.2, a new distribution algorithm, `eager_free`, has been introduced to take advantage of the fact that not all operators execute at the same time. The query plan is divided into stages and Greenplum Database *eagerly frees* memory allocated to a previous stage at the end of that stage's execution, then allocates the eagerly freed memory to the new stage.

  Note that the `eager_free` algorithm is enabled automatically in the database and is transparent to the user.

Greenplum Database Extension Framework and Turnkey In-Database Analytics

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

Greenplum Extensions can be downloaded from EMC Download Center and installed using the new Greenplum Package Manager. This new Greenplum Database utility ensures automatic installation and updates of functional extensions like in-database GeoSpatial functions, PL/R, PL/Java, PL/Perl and pgcrypto.

Greenplum extensions dramatically simplify the task of enabling and managing advanced in-database functionality across a cluster. For example, extensions automatically get deployed on new nodes during expansions of Greenplum clusters.

- **PostGIS**

  PostGIS is a spatial database extension for PostgreSQL that allows GIS (Geographic Information Systems) objects to be stored in the database. PostGIS includes support for GiST-based R-Tree spatial indexes, and functions for analysis and processing of GIS objects. The PostGIS extension is available from the EMC Download Center and is installed using the Greenplum Package Manager (`gppkg`), see below.

- **New Greenplum Package Manager**

  A new package manager (`gppkg`) allows the user to install the Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, and PostGIS, 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.

  The following extensions are available from the EMC Download Center.

  - **Greenplum Database 4.2 - Extensions - PostGIS 1.0**

    Postgis version 1.4.2

    Proj version 4.7.0

    Geos version 3.2.2

  - **Greenplum Database 4.2 - Extensions - PL/Java 1.0**

    PL/Java version (based on 1.4.0)

    Java JDK version 1.6.0_26

  - **Greenplum Database 4.2 - Extensions - PL/R 1.0**
    PL/R version 8.3.0.12
    R version 2.13.0
  - **Greenplum Database 4.2 - Extensions - PL/Perl 1.0**
    PL/Perl (shipped with the database – based on GPDB)
    Perl (current version 5.8.8)
  - **Greenplum Database 4.2 - Extensions - Pgcrypto 1.0**
    Pgcrypto (based on Postgresql 8.3).

IPv6 support

As the address exhaustion of Internet Protocol version 4 (IPv4) approaches, support of its successor, Internet Protocol version 6 (IPv6), has become more and more important.

This release of Greenplum Database provides support for IPv6 as well as IPv4 addressing. This new support is transparent to end-users.

Greenplum Database 4.2 supports IPV6 only, IPV4 only, as well as dual stack environments for client-master connections, as follows:

- Exclusive IPv6 environment (Master Server, Segment Hosts, Clients)
- Exclusive IPv4 environment (Master Server, Segment Hosts, Clients)
- IPv6 with dual-stack master:
  - Master Server and Standby Server support both IPv4 and IPv6 client connections (psql, odbc, jdbc, etc.)
  - All the Segment Servers must be exclusive IPv6
  - gpfdist Server supports the same network protocol as Segment Servers (IPv6)

Note: Performance Monitor Web Server 4.1.1.8 supports Greenplum Database running on either IPv4 or IPv6 environments noted above.

When using IPv6, any references to IP addresses must be enclosed in brackets [ ] as shown in the following examples:

`gpfdist` example:

```
CREATE EXTERNAL TABLE ext_gpfdist_table
(num int)
LOCATION ('gpfdist://[2620:0:170:610::11]:7777/ex.txt')
FORMAT 'TEXT' (DELIMITER '|');
```

Management utility (`gpdbrestore`) example:

```
gpdbrestore -R
[2620:0:170:610::11]:/path_to/master/gpseg-1/db_dumps/201108
```

Note: Data Domain Boost does not support IPv6.

New Performance Monitor Tables

In order to facilitate the collection of network statistical data, four new Performance Monitor tables have been added, they are:

- `socket_stats`
- `interface_stats`
- `udp_stats`
- `tcp_stats`

gpfdist as a Windows Service

In previous versions of the Greenplum Loaders, `gpfdist` could not be run as a Windows service. The current Greenplum Loaders allow `gpfdist` to run as a Windows service.

See "Greenplum Database Client Tool Packages and Performance Monitor" on page 26 for detailed compatibility information for all the Greenplum client tool packages.

See the *Management Utility Reference* Appendix of the *Greenplum Database Administrator Guide 4.2* for information about running `gpfdist` as a Windows Service.

Time-Based Authentication

Greenplum Database enables the administrator to restrict access to the database based on time. The restrictions are set on a role basis as part of the `CREATE` or `ALTER ROLE` command.

Access can be restricted by day or by day and time. The constraints are removable, without deleting and recreating the role.

Changed Features in Greenplum Database 4.2

- New Reserved Word
- SQL Commands
- Management Utilities
- Server Configuration Parameters
- System Catalogs

Refer to the *Greenplum Database Administrator Guide 4.2* for details about usage changes.

New Reserved Word

`decode` will be converted into a reserved word in the next Service Pack release of Greenplum Database, version 4.2.1.

If you want to use `decode` as a function you must invoke it using the full schema-qualified syntax or by enclosing the function name in quotes, `"decode"()`. Greenplum recommends customers who wish to retain the use of the `decode` function to take necessary steps to prepare for the change.

For example: Once `decode` becomes a reserved word, if you have a function `decode(a, b, c)` you should invoke it either:

without specifying its schema:

```
SELECT “decode”(1, 2, 1);
```

or, if specifying the schema for `decode`, you can omit the double quote, as follows:

```
SELECT public.decode(1, 2, 1);
```

SQL Commands

Table 1 SQL Command Changes in 4.2

SQL Command	Description of Change
`ALTER ROLE protocol='gphdfs'`	Starting in Greenplum Database 4.2, gphdfs protocol privileges can no longer be granted using `ALTER ROLE protocol='gphdfs'`. gphdfs privileges are now granted using: `GRANT INSERT ON PROTOCOL gphdfs TO username;` See the *Grant Privileges for the HDFS Protocol* section in the *Greenplum Database 4.2 Administrator Guide* for more details.

`CREATE ROLE`
`ALTER ROLE`
	New syntax added to support new Time-Based Authentication feature.

Management Utilities

Table 2 Management Utility Changes in 4.2

Utility Name	Description of Change
`gppkg`	A new management utility has been created with which to install Greenplum Database extensions such as pgcrypto, PL/R, PL/Java, PL/Perl, PL/R, along with their dependencies, across an entire cluster.
`gpfdist`	In previous versions of the Greenplum Loaders, `gpfdist` could not be run as a Windows service. The Greenplum Loaders version 4.1.1.3 and above allow `gpfdist` to run as a Windows service.
`gp_dump`	Deprecated. gp_dump will be deprecated in the next major release of Greenplum Database.
`gpcrondump`	The following options were removed: `-w, -i, -p` New Data Domain Boost support added. Note that some options work differently or are not supported when `-ddboost` is specified. See the *Greenplum Database Administrator Guide 4.2* for details.
`gpinitsystem`	`-m` option removed

Server Configuration Parameters

- New Parameters
- Parameters with Changed Defaults
- Removed Parameters

New Parameters

Table 3 New Server Configuration Parameters in 4.2

Parameter Name	Value Range	Default Value	Description	Set Classifications
gp_dynamic_partition_pruning	on/off	on	This parameter enables plans that can dynamically eliminate scanning of partitions.	master session reload
gp_filerep_tcp_keepalives_count	number of lost keepalives	2	Maximum number of TCP keepalive retransmits for FileRep connection.	local system restart
gp_filerep_tcp_keepalives_idle	number of seconds	1 min	Seconds between issuing TCP keepalives for FileRep connection.	local system restart
gp_filerep_tcp_keepalives_interval	number of seconds	30 sec	Seconds between TCP keepalive retransmits for FileRep connection.	local system restart
gp_hadoop_xnet_version	`gphd-xnet-1.0.0.0` or `gpmr-xnet-1.0.0.0`	`gphd-xnet-1.0.0.0`	The installed Hadoop Cross Connect version.	local session reload
gp_hadoop_home	Same value as `HADOOP_HOME`	NULL	The location where Hadoop is installed in each segment.	local session reload

Parameters with Changed Defaults

Table 4 Changed Server Configuration Parameters in 4.2

Parameter Name	Old Value	New Value	Description
from_collapse_limit	16	20	Sets the FROM-list size beyond which subqueries are not collapsed.
gp_connections_per_thread	512	64	Sets the number of client connections handled in each thread.
gp_fts_probe_timeout	3 mins	10 s	Maximum time (in seconds) allowed for FTS to complete probing a segment.
gp_interconnect_setup_timeout	20 s	5 min	Timeout (in minutes) on interconnect setup that occurs at query start
gp_resqueue_memory_policy	auto	eager_free	Sets the policy for memory allocation of queries.
gp_segment_connect_timeout	3 mins	10 min	Maximum time (in minutes) allowed for a new worker process to start or a mirror to respond.
join_collapse_limit	16	20	Sets the FROM-list size beyond which JOIN constructs are not collapsed.

Table 4 Changed Server Configuration Parameters in 4.2

Parameter Name	Old Value	New Value	Description
max_appendonly_tables	2048	10000	Maximum number of different (unrelated) append only tables that can participate in writing data concurrently.
max_connections	25	250 on Master, 750 on Segments	Sets the maximum number of concurrent connections.
max_prepared_transactions	50	250	Sets the maximum number of simultaneously prepared transactions.

Removed Parameters

The following server configuration parameters were deprecated in Greenplum Database 4.1 and have been removed in release 4.2.

Table 5 Removed Server Configuration Parameters

Parameter Name
maintenance_work_mem
max_work_mem
work_mem

System Catalogs

Removed Catalog Objects

The following catalog objects were deprecated in Greenplum Database 4.1 and have been removed in release 4.2.

Table 6 Removed Catalog Objects

Object Type	Schema	Catalog Name	Description
table	pg_catalog	pg_resqueue	Functionality moved to the `pg_resqueuecapability` table
view	pg_catalog	pg_resqueue_status	Use the `gp_toolkit.gp_resqueue_status` view instead

Resolved Issues in Greenplum Database 4.2

The following tables list customer reported issues that are now resolved in Greenplum Database 4.2.

- Table 7, "Resolved Issues in 4.2 (not previously documented)"

  This table lists issues that were not previously documented in the release notes for any patch or feature release of Greenplum Database. For issues resolved in previous releases, including patch releases to Greenplum Database 4.1 or 4.0.x, see the corresponding release notes, available from Powerlink.
- Table 8, "Resolved Issues in 4.2 (previously documented as Known Issues)"

  This table lists resolved issues that were documented as Known Issues in previous release notes.

Table 7 Resolved Issues in 4.2 (not previously documented)

Issue Number	Resolved In	Category	Description
14667	4.2	Loaders	**gpload: not-null constraint violations** gpload was throwing not-null constraint violations when using reusable temporary tables. This issue has been resolved in this release.
14545	4.2	Management Scripts	**gpexpand failed with tablenames that contain spaces** gpexpand did not handle spaces in tablenames for partition tables. This issue has been resolved in this release.
14244	4.2	Query Execution	**Window function with EXCLUDE returned incorrect results.** The EXCLUDE clause was not implemented and returned incorrect results rather than the appropriate error message. This issue has been resolved in this release.
14124	4.2	Query Execution	**Memory usage and performance issues with Nested Loop Join** The performance of Nested Loop Join against an append-only column-oriented table was much slower than the performance of a Nested Loop Join on a heap table. This issue has been resolved in this release.
14092	4.2	Query Execution	**Queries with LIMIT and ORDER BY returned truncated results** Queries with LIMIT and ORDER BY returned truncated results when the number of produced tuples (LIMIT predicate) is high relatively to the used memory (statement_mem). This issue has been resolved in this release.
14001	4.2	Functions and Languages	**Auto-analyze did not work from within PL/pgSQL functions** In a PL/pgSQL Function, after truncating a table and inserting into it, the table statistics remained empty. This issue has been resolved in this release.
13818	4.2	Backup Restore	**gp_dump failed during backup** gpcrondump (and the underlying gp_dump) failed when excluding (using the -T option) tables that contain sequences. This issue has been resolved in this release.

Table 7 Resolved Issues in 4.2 (not previously documented)

Issue Number	Resolved In	Category	Description
13806	4.2	DDL/DML: Partitioning	**Adding unbound partition to a table failed** Adding an unbound partition to the beginning of a partitioned table failed if the end value is next to the start of the first partition, but succeeded if there was a gap. This issue has been resolved in this release.
13750	4.2	DDL/DML: Partitioning	**Adding a partition failed if a partial index existed** Adding a partition to a table failed with a `relation ... does not exist` error if there was a partial index on the partitioned table. Without the partial index (if there was no index or the index was not partial) the statement succeeded. This issue has been resolved in this release.
13718	4.2	Dispatch	**Session deadlock** In rare circumstances, a session deadlock occurred (only one session affected) when the handling server process on the master tried to acquire a lock it already held. This issue has been resolved in this release.
13592	4.2	Query Execution	**`ERROR: seqserver socket is closed`** When a sequence reached its limit (max or min), the sequence server process would error out. This was closing the connection with the client, causing the client to print out a "socket closed" error. Next time the client would try to connect to the sequence server, seqserver would crash because of uncleaned state left from erroring out. This issue has been resolved in this release.
13576	4.2	Query Execution	**Running a query with concurrency caused database lockdown** In rare circumstances, concurrently run queries encountered an internal deadlock. This issue has been resolved in this release.
13452	4.2	Backup and restore	**Postgresql reserved words** Using reserved postgresql keywords such as `range` in column names was causing failures in backup / restore and analyze operations. This issue has been resolved in this release.
13399	4.2	Loaders	**PWX/gpload did not create table including defaults** When adding a column with default values into a table, the loader was not using the `INCLUDING DEFAULTS` option, causing `NULL` to get loaded into the column. This issue has been resolved in this release.
13395	4.2	DDL/DML: Partitioning	**`ALTER TABLE DROP PARTITION` failed** Unable to drop a partition when bitmap indexes were present. This issue has been resolved in this release.
13273	4.2	Client Tools	**SELECT statement from an external hadoop table returned wrong number of rows** This issue has been resolved in this release.
13264	4.2	Query Execution	**Incorrect data distribution when using window function** In some circumstances, using a window function within a `CREATE TABLE AS SELECT` statement led to incorrect data distribution. This issue has been resolved in this release.

Table 7 Resolved Issues in 4.2 (not previously documented)

Issue Number	Resolved In	Category	Description
13214	4.2	SQL Parser	**Difference in behavior of `to_date` WITH `to_char` built-in functions** In some releases of Greenplum Database the `to_date` and `to_char` functions didn't process leading spaces the same way. This issue has been resolved in this release.
13190	4.2	Functions and Languages	**Timezone type not working with WIT (Asia/Jakarta)** For systems configured with specific timezones (like WIT), FTS failed to update catalog tables. This issue has been resolved in this release.
13153	4.2	Backup and Restore	**`gpdbrestore -L` failed** `gpdbrestore -L` failed for dump files with excessively long lines, due to a limitation of awk on the Solaris platform. This issue has been resolved in this release.
13119	4.2	OLAP	**`ROLLUP` query failed** Queries with `ROLLUP` on a constant expression in the `FROM` clause failed. This issue has been resolved in this release.
12803	4.2	Catalog and Metadata	**The `pg_catalog.pg_max_external_files` view was incorrectly based on an obsolete table** The system configuration table `gp_configuration` was obsoleted in Greenplum Database 4.0 and superseded by new table `gp_segment_configuration`. The view now refers to the correct table. This issue has been resolved in this release.
12791	4.2	Dispatch	**Segment failures incorrectly reported** Under certain circumstances, Greenplum Database incorrectly reported segment failures when the real issue was a network connection problem.
12767	4.2	gpmapreduce	**`gpmapreduce` returned incorrect exit code** `gpmapreduce` did not always return non-zero exit code on failure. This issue has been resolved in this release.
12663	4.2	Query Execution	**Unexpected internal error: Segment process received signal SIGSEGV** In rare circumstances, queries with more than one sort key in the execution plan resulted in this error. This issue has been resolved in this release.
12637	4.2	Query Execution	**Constraint error caused query to fail** Queries failed when trying to coerce a value to a domain (`COERCETODOMAIN)` and the value violated the domain constraint. This issue has been resolved in this release.
12531	4.2	Query Execution	**SQL UDA extremely slow** Performance issue when using SQL user-defined aggregate with perfmon enabled. Performance monitor statistics were sent out for each row processed by the aggregate, which slowed down the query. This issue has been resolved in this release.

Table 7 Resolved Issues in 4.2 (not previously documented)

Issue Number	Resolved In	Category	Description
12421	4.2	Query Optimizer	**Unexpected cache lookup failure in `EXPLAIN`** In rare circumstances, `EXPLAIN` would report: `NOTICE: cache lookup failed for attribute A of relation NNNNN` and then misreport the query plan. Queries triggering this issue had an index scan directly under a nested loop. This issue has been resolved in this release.
12135	4.2	Backup and Restore	**`gpcrondump -q` not suppressing output as expected** When `-q` flag (no screen output) was set, `gpcrondump` did not suppress output for `-k` and `-j` flags as expected. This issue has been resolved in this release.
11865	4.2	Backup and Restore	**Tables with indexes slow to restore** The restore operation on a table with an index was taking place after the index was created, which slowed the restore process. The index is now created after restoring the data. The `CREATE INDEX` DDL is saved in a separate `post_data` backup file. This issue has been resolved in this release.
10341	4.2	Management Scripts	**Read operation failed on external table** Unable to query gp_skew_coefficients on external tables. This issue has been resolved in this release.
3759	4.2	DDL and Utility Statements	**Dropping a primary key or unique constraint did not cascade** Adding a primary key or unique constraint to a partitioned table would correctly cascade the constraint to its parts, but dropping the same constraint did not cascade. This issue has been resolved in this release. Now dropping such a constraint from a partitioned table, eliminates the constraint from all parts. Note, however, that simple unique indexes (those not associated with constraints) behave as before.

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

Table 8 Resolved Issues in 4.2 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
12919 12918	4.2	Loading Data	**Using TRUNCATE option with MERGE or UPDATE lost data** If `TRUNCATE` was set to true and the mode was `UPDATE` in the `YAML` file, all existing data was wiped out, and nothing updated when `gpload` was executed. All data in the target table was lost, and the target table was empty. If `TRUNCATE` was set to true and the mode was `MERGE` in the `YAML` file, all existing records were wiped out when `gpload` was executed, and all records in the source file were inserted. The target table only contained the new records, but all the previously existing data was lost. This issue has been resolved in this release.
12917 12916	4.2	Loading Data	**Table names could not contain special characters in the YAML file** If a table name contained a special character or space and the `YAML` configuration file set `REUSE_TABLES` to `true`, the `gpload` utility failed. This issue has been resolved in this release.
12915 12914	4.2	Loading Data	**Count of data formatting errors was incorrect** If an error tolerance was set in a `YAML` file, errors were successfully written to an error table, but the log and output messages incorrectly displayed zero errors, as shown below: `data formatting errors = 0` This issue has been resolved in this release.
14219	4.2	Documentation	**Incorrect default batch size for `gpexpand` in documentation** The default batch size (-B option) for the gpexpand utility was incorrectly documented in the help file and the Greenplum Administrator Guide as 8. The actual default batch size is 16. This issue has been resolved in this release.
13425	4.2	Documentation	**Documentation error** The GPDB Administrator Guide incorrectly stated that `gpfdist` supported compression on Windows. This issue has been resolved in this release.

12846	4.2	Documentation	**Incorrect/missing OS system settings in GPDB 4.1 Installation Guide**
			The following OS System settings were missing from the installation guide:
			`kernel.msgmni = 2048`
			`net.ipv4.ip_local_port_range = 1025 65535`
			The `kernel.sem` OS System setting had an incorrect value in the Installation Guide. The correct value is:
			`kernel.sem = 250 512000 100 2048`
			This issue has been resolved in this release.

12472		Memory	**No available memory** In very rare circumstances, the Greenplum Performance Monitor process `gpmonws` used all available memory on the master node. This issue has been resolved in this release.

Table 8 Resolved Issues in 4.2 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
12334, 12604	4.2	Fault Detection and Recovery	**Shared memory error caused some segment mirrors to become unresponsive** In very rare circumstances, an issue with shared memory in the Greenplum Database resulted in the following error: `"FATAL","XX000","semop(id=21495876,num=13) failed: Numerical result out of range (pg_sema.c:499)"` When this error occurred, mirror segments could become unresponsive, and the system went into change-tracking mode. This issue has been resolved in this release.
12190	4.2	Query Optimizer	**Query plan was not created when redistribution was on a column that does not support hashing** The Greenplum Database query planner did not generate a query plan if the plan requires redistribution based on a column that contains a data type that does not support hashing. This issue has been resolved in this release.
12165, 12173	4.2	Query Execution	**Queries with multi-row subqueries were not supported** Greenplum Database did not support queries that contain multi-row subqueries. This issue has been resolved in this release.
12120	4.2	Installation	**Greenplum Database directories showed the incorrect group for the owner user if the owner name and the group name were different** The installation package assumed the user name and group are the same; that is, `gpadmin` is part of the group named `gpadmin`. If this is not the case, the group was incorrect. This issue has been resolved in this release.

Table 8 Resolved Issues in 4.2 (previously documented as Known Issues)

Issue Number	Resolved In	Category	Description
12082	4.2	Query Execution	**Error when GROUP BY and a SELECT clauses referred to a column in different ways** Greenplum Database interprets queries that mix windowing (OVER clause) and aggregation (GROUP BY clause and/or aggregate functions) as if the grouping operation is performed first, and the windowing operation is performed on the result. In cases where the FROM clause specifies a join with aliases on the join terms, a query might refer to a column in a join term or to the same column in the result of the join. In very rare cases, an internal error occurred when a GROUP BY clause referred to a column in one way and a SELECT clause referred to it in a different way. This issue has been resolved in this release.
5647	4.2	Upgrade	**Upgrade utility required standard prefixes for segment data directories** The gpmigrator utility requires a single, standard segment prefix to perform a successful upgrade. System configurations that have been modified with multiple or omitted segment prefix values in segment data directories were not recognized as valid by gpmigrator, and could not be upgraded using this utility. The naming convention for data directories in a Greenplum Database system is *SEG_PREFIXnumber* where number starts with 0 for segment instances (the master is always -1). So for example, if you choose the prefix gp, your master instance data directory would be named gp-1,and the segment instances would be named gp0, gp1, gp2, and so on. This issue has been resolved in this release.

Known Issues in Greenplum Database 4.2

This section lists the new known issues in Greenplum Database 4.2. A work-around 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 9 All Known Issues in 4.2

Issue	Category	Description
15596	Installer	**Tools package installer incorrectly referencing documentation** After installing any of the Client, Load, or Connectivity tools packages, the installer incorrectly references product documentation in the installation directory. Documentation is no longer installed as part of these packages. Download the appropriate documentation from Powerlink.
15443	Installer	**Directory paths containing spaces are not supported by the installer** The installer may fail to extract into a directory path that contains an embedded space.
15379	Query Optimizer	**Dynamic partition elimination not applied** Dynamic partition elimination is not applied when the partition key is timestamp and joining attribute is of date type. **Workaround**: Explicitly cast the joining attribute to a timestamp.
15254	Management Scripts Suite	**`pljava_classpath` cannot be set using `gpconfig`** **Workaround:** Manually set `pljava_classpath` in `postgres.conf file`. Put the value within quotes, for example: `pljava_classpath='examples.jar'`
15098	Management Scripts Suite	**`gpseginstall` has an 8 character limitation on passwords** The `gpseginstall` utility is used to create and set the system user on all hosts as part of the installation of the database software on all hosts. gpseginstall limits the password length to 8 characters. Use the workarounds below to avoid this limitation and create stronger passwords. **Workaround:** We recommend creating stronger passwords by either: • Pre-creating users and passwords on all the hosts prior to running `gpseginstall`. • Changing the password either manually or by using gpssh after running `gpseginstall`.
14538	Query Optimizer	**Correlated subqueries can fail with ERROR: more than one row returned by a subquery used as an expression** In very rare circumstances when a correlated subquery in the `WHERE` clause has another correlated subquery in its `WHERE` clause that cannot be unnested to a join, the query can fail. **Workaround:** Rewrite the query.
14537	Backup and Restore	**Data Domain Boost does not support IPV6**
14077	Backup and Restore	**Backup and Restore: case sensitive database names not handled well** When a database name is case sensitive (like GreenplumDB), `gpdbrestore -e` option (recreate database) fails because it does not handle the database name correctly.

Table 9 All Known Issues in 4.2

Issue	Category	Description
13810	Loaders	**`gpfdist` on Windows 2008 connection error** On Windows 2008 server, when started either as a service or via the command-line, `gpfidst` fails due to a connection error: ERROR: connection with gpfdist failed for gpfdist://10.110.121.161:9090/*.csv. effective url: http://10.110.121.161:9090/*.csv. error code = 115 (Operation now in progress) (seg1 slice1 rh55-qavm62:6101 pid=3313) This is caused by the first `gpfdist` instance binding to the IPv6 address, even if IPv6 is disabled on the machine. The Greenplum Database process that issues the query for `gpfdist` is trying to connect using an IPv4 address (in the referenced error message, 10.110.121.161). **Workaround**: Start a second instance of `gpfdist` using the same port, so that it will bind to the IPv4 address.
13342	Fault Detection and Transitions	**`gpstate` can report unreliable status information** Under circumstances where there is already a fault condition and communication with FTS is already unreliable, `gpstate -s` can report unreliable status information. As part of its process, gpstate –s examines all it learns about each segment. The process that examines the segment state relies on `gp_primarymirror’s` `getVersion` and `getMirrorStatus` messages and on a check of the postmaster’s pid file, meaning that problems not detectable via those checks are not considered segment failures by `gpstate -s`.
10640	Query Optimizer	**No pruning with `COALESCE`** Partitions are not pruned when `WHERE` clause filters on a `COALESCE` expression on the partition key. **Workaround**: Explicitly include the filter on the partition key without the `COALESCE`.
9549	Management Scripts Suite	**`gprecoverseg -F` failed** In cases where `gprecoverseg -F` is run at the same time a segment is marked down, `gprecoverseg -F` fails with the following error: gprecoverseg failed. (Reason='error 'can't start transaction' in 'BEGIN'') exiting... **Workaround**: Rerun `gprecoverseg -F` if you see this error.

Upgrading to Greenplum Database 4.2

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

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

- Planning Your Upgrade
- Upgrading from 4.1.x.x to 4.2.0.0
- 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 filespace) 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 4.1.x.x to 4.2.0.0

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.1.x.x to 4.2.0.0. Detailed upgrade instructions are also 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).

□	Check for and recover any failed segments (`gpstate`, `gprecoverseg`).
□	Install the Greenplum Database 4.2.0.0 binaries on all Greenplum hosts.
□	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.
□	(Optional) Run `VACUUM` on all databases, and remove old server log files from `pg_log` 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.
□	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

* The system will be locked down to all user activity during the upgrade process

□	Backup your current databases (`gpcrondump` or ZFS snapshots) and secure backup files in a location outside of your Greenplum data directories.
□	Remove the standby master from your system configuration (`gpinitstandby -r`).
□	Do a clean shutdown of your current system (`gpstop`).
□	Update your environment to source your Greenplum Database 4.2.0.0 installation.
□	Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
□	After the upgrade process finishes successfully, your 4.2.0.0 system will be up and running.
	Post-Upgrade (on your 4.2 system)

* The 4.2.x system is up

□	Reinitialize your standby master host (`gpinitstandby`).
□	Upgrade `gpfdist` on all of your ETL hosts by installing the compatible version 4.2 Load Tools package.
□	Rebuild any custom modules against your 4.2 installation (for example any shared library files for user-defined functions in `$GPHOME/lib`).

□	Greenplum Database 4.2 introduced the Greenplum Package Manager (`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* for usage details.
□	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 26 for detailed compatibility information.
□	Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.2 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 (`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 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-4.2.0.0`), 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 installation. For example:

   ```
   $ source /usr/local/greenplum-db-4.2.0.0/greenplum_path.sh
   ```

9. Run the gpseginstall 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:

   ```
   $ gpseginstall -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 installation. For example:

    ```
    $ source /usr/home/greenplum-db-4.2.0.0/greenplum_path.sh
    ```

15. Update your environment so that it is sourcing your new 4.2 installation.

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

       ```
       # rm -rf /usr/local/greenplum-db
       # ln -s /usr/local/greenplum-db-4.2.0.0
       /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-4.2.0.0
       /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 version of the gpmigrator utility with the --check-only option.

17. As gpadmin, run the 4.2 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-4.2.0.0
```

Note: If the migration does not complete successfully, contact Customer Support (see "Troubleshooting a Failed Upgrade" on page 26).

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 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 Load Tools package. Application Packages are available at the EMC Download Center.

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

22. Greenplum Database 4.2 introduced the Greenplum Package Manager (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* 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 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.

- 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 4.2.0.0.

For Users Running Greenplum Database 3.3.x.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.1.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 4.2.0.0.

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 Perfomance Monitor and client tool packages compatible with this Greenplum Database release. Client tool packages are available at the EMC Download Center.

Table 10 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]	4.2.0.0	4.2.0.0
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.2.0.0	4.2.0.0
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.2.0.0	4.2.0.0
Greenplum Database Performance Monitor	Greenplum Database monitor for query and system performance	4.1.1.8	4.2.0.0

[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, Greenplum enables turn-key in-database analytics via Greenplum Extensions.

Greenplum Extensions can be downloaded from EMC Download Center and installed using the new Greenplum Package Manager (`gppkg`. See the *Greenplum Database Administrator Guide 4.2* for usage 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 11 Greenplum Database Extensions Compatibility

Greenplum Database Extension	**Extension Components**	
	Name	**Version**
PostGIS 1.0 for Greenplum Database 4.2	Postgis	1.4.2
	Proj	4.7.0
	Geos	3.2.2
PL/Java 1.0 for Greenplum Database 4.2	PL/Java	Based on 1.4.0
	Java JDK	1.6.0_26
PL/R 1.0 for Greenplum Database 4.2	PL/R	8.3.0.12
	R	2.13.0
PL/Perl 1.0 for Greenplum Database 4.2	PL/Perl	Based on Greenplum Database version
	Perl	5.12.4
Pgcrypto 1.0 for Greenplum Database 4.2	Pgcrypto	Based on Postgresql 8.3

Greenplum Database 4.2 Documentation

The following Greenplum Database documentation is available in the `$GPHOME/docs` directory of your Greenplum installation and on Powerlink. Greenplum documentation is provided in PDF format.

Table 12 Greenplum Database Documentation

Title	**Part Number**	**Revision**
Greenplum Database 4.2 Installation Guide	300-013-162	A01
Greenplum Database 4.2 Administrator Guide	300-013-163	A01]
Greenplum Database 4.2 Client Tools for UNIX	300-013-190	A01
Greenplum Database 4.2 Client Tools for Windows	300-013-191	A01

Table 12 Greenplum Database Documentation

Title	Part Number	Revision
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 Release Notes	300-013-265	A01
Greenplum Database Performance Monitor 4.1 Administrator Guide	300-012-430	A04