GREENPLUM®
A DIVISION OF EMC

EMC²®

P/N: 300-013-342
Rev: A02
Updated: January 23, 2012

Welcome to Greenplum Database 4.1.1.8

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 4.1.1.8

This release includes new issue resolutions and minor feature enhancements. Please refer to the following sections for more information about this release.

- New Functionality in Greenplum Database 4.1.1.x
- Resolved Issues in Greenplum Database 4.1.x.x
- Known Issues in Greenplum Database 4.1.1.8
- Upgrading to Greenplum Database 4.1.1.x
- Greenplum Database Client Tool Packages and Performance Monitor
- Greenplum Database 4.1.1.8 Documentation

Release Notes for previous 4.1.1.x and earlier releases are available on Powerlink.

New Functionality in Greenplum Database 4.1.1.x

In previous versions of the Greenplum Loaders, `gpfdist` could not be run as a Windows service. The Greenplum Loaders version 4.1.1.3 allow `gpfdist` to run as a Windows service.

Greenplum Loaders version 4.1.1.3 are compatible with all 4.1.x versions of the Greenplum database. See "Greenplum Database Client Tool Packages and Performance Monitor" on page 26 for detailed compatibility information for all the Greenplum client tool packages.

Follow the instructions below to download, register and activate `gpfdist` as a service:

1. Update your Greenplum Loader package to version 4.1.1.3. This package is available from the EMC Download Center.

2. Register `gpfist` as a Windows service:

   a. Open a Windows command window

b. Run the following command:

```
sc create gpfdist binpath= "<path_to>gpfdist.exe -p 8081
-d External\load\files\path -l Log\file\path"
```

You can create multiple instances of `gpfdist` by running the same command again, with a unique name and port number for each instance, for example:

```
sc create gpfdistN binpath= "<path_to>gpfdist.exe -p 8082 -d
External\load\files\path -l Log\file\path"
```

3. Activate the `gpfdist` service:

a. Open the Windows Control Panel and select **Administrative Tools>Services**.

b. Highlight then right-click on the `gpfdist` service in the list of services.

c. Select **Properties** from the right-click menu, the **Service Properties** window opens.

Note that you can also stop this service from the **Service Properties** window.

d. Optional: Change the **Startup Type** to **Automatic** (after a system restart, this service will be running), then under **Service status**, click **Start**.

e. Click **OK**.

Repeat the above steps for each instance of `gpfdist` that you created.

Resolved Issues in Greenplum Database 4.1.x.x

This following table lists the customer reported issues that are now resolved in Greenplum Database 4.1.x.x.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
15439	4.1.1.8	Catalog and Metadata	**`gpcheckcat` foreign key errors** Dropping the grantor role caused `gpcheckcat` to report foreign key errors. This issue has been resolved in this release.
15292	4.1.1.8	Catalog and Metadata	**`gpcheckcat` slow peformance** In a large scale environment `gpcheckcat` in some cases was taking a long time to complete. This issue has been resolved in this release.
14396	4.1.1.8	Query Execution	**Error: DELETE fails with"PANIC","XX000","Waiting on lock already held!'** Updates and deletes resulted in a segment restart and transactions being aborted. This issue has been resolved in this release.
13624	4.1.1.8	Replication/ Crash Recovery	**Error: Shutdown fails with '"PANIC","XX000","Waiting on lock already held!' error** A forced shutdown during concurrent activity on the system resulted in the above error and the master was unable to restart. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
13420	4.1.1.8	Monitoring: Performance Monitor	**Performance Monitor performance issues** Performance Monitor's python process was using large amount of memory which was causing other python-based management utilities to throw various errors. This issue has been resolved in this release.
12614	4.1.1.7	Replication/ Crash Recovery	**Error: 'Cannot read status of transaction from pg_clog file'** In some circumstances, the database could not restart, because the transaction status was not available. This issue has been resolved in this release.
14283	4.1.1.6	Loaders	**Error: 'Address already in use'** In some circumstances when mulitple gpload sessions were started at the same time using the same `PORT_RANGE`, some sessions might fail with "Address already in use" error. This issue has been resolved in this release.
14667	4.1.1.6	Loaders	**gpload: not-null constraint violations** gpload was throwing not-null constraint violations when using reusuable temporary tables. This issue has been resolved in this release.
14519	4.1.1.6	Functions and Languages	**Error: 'relation with OID xxxx does not exist'** The above error was generated in cases where tables were being created and dropped frequently. This was a result of an issue with earlier versions of Postgres. This issue has been resolved in this release.
14125	4.1.1.5	Query Execution	**Query with AGGREGATE functions failed** Some queries that called `AGGREGATE` functions failed with a 'Segment process received signal SIGSEGV (postgres.c:3360)' error. This issue has been resolved in this release.
14037	4.1.1.5	Backup/ Restore	**Slow multiple-table restore from backup** When specifying multiple tables to be restored from a preexisting dump file, a portion of the restore operation was being repeated unnecessarily, generally resulting in longer completion times. This issue has been resolved in this release.
14011	4.1.1.5	Resource Management	**Resource queue initialization error** A rare condition in resource management initialization may have lead to the Greenplum Database being unable to start. When this occurred, the system failed with the message "insufficient resource queues available". This issue has been resolved in this release.
13999/ 13973	4.1.1.5	DDL/DML: Heap	**Insufficient memory error** Under rare circumstances, updates involving very large numbers of records can result in an 'insufficient memory' error. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
13969	4.1.1.5	DML	**Window function returning incorrect results** A window function that specifies `RANGE BETWEEN interval '0 seconds' FOLLOWING` returns incorrect results. This issue is resolved with the following workaround: Use `'0 seconds' PRECEDING` to specify the interval.
13845	4.1.1.5	Query Optimizer	**SQLPrepare did not throw exceptions for unsupported query constructs** The ODBC function, SQLPrepare, used to test the validity of SQL statements prior to sending the statement to GPDB for execution, incorrect indicated that unsupported query constructs were supported. This issue has been resolved in this release.
11911	4.1.1.5	Monitoring: Performance Monitor	**Performance Monitor historical metrics not displaying correctly** In the Performance Monitor user interface, system metrics graphs covering durations of 1hr (the default) or 6hrs did not display properly. This issue has been resolved in this release.
14145	4.1.1.5	Upgrade/ Downgrade	**gpexpand failed due to a preexisting, inconsistent gpexpand schema** The existence of an inconsistent 'gpexpand' schema caused a failure in the setup phase of gpexpand, thereby requiring a rollback of several preceding steps. This issue has been resolved in this release.
14144	4.1.1.5	Management Tools	**Scalability issue during subprocessing in Management Tools** File descriptors used during the creation of subprocesses in the Managment Tools were not being properly recycled. At large enough scale, this was causing exhaustion of available file descriptors, preventing any further subprocesses from being spawned. In rare cases, this could cause failures of in-progress operations like starting, stopping, expanding, etc. This issue has been resolved in this release.
14010	4.1.1.4	Upgrade/ Downgrade	**Log files removed after upgrade** Running gpmigrator to upgrade removed the master log files. This issue has been resolved in this release.
12659	4.1.1.4	Query Optimizer	**"ERROR","XX000", stanumbers is not a 1-D float4 array** Under rare circumstances, invalid statistics caused some statements to fail with the above error. This issue has been resolved in this release.
PRFMN-4	4.1.1.4	Performance Monitor	**Performance Monitor errors when using IE8** When connecting to Performance Monitor using Internet Explorer 8 the web server started streaming errors as soon as a connection to the web server from the browser was made. These errors were being logged in lighttpd-errors.log, potentially causing the partition to run out of disk space. This issue has been resolved in this release.
13023	4.1.1.4	Monitoring: Performance Monitor	**Performance Monitor performance issues** Performance Monitor’s python process was using large amount of memory which was causing other python-based management utilities to throw various errors. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
13879	4.1.1.4	Query Execution	**Window function using order by/range returned incorrect results** `max() over (order by ... range ...)` returned incorrect results when applied to a numeric column. This issue has been resolved in this release.
13817	4.1.1.4	Loaders: gpfdist	**Write performance issues when using writable external table** gpfdist (loader) was causing performance issues when opening the external writable table target. Prior to this release the default behavior for gpfdist was to open files for synchronous I/O. This meant that any writes on the resulting file descriptor would block gpfdist until the data has been physically written to the underlying hardware. With this release, opening files for synchronous I/O must be explicitly specified by using a new option that has been added to gpfdist (-S). This issue has been resolved in this release.
13596	4.1.1.3	Catalog and Metadata	**pg_type corruption** Some queries were causing corruption of the pg_type system catalog table that resulted in the database being unusable. This issue has been resolved in this release.
13414	4.1.1.3	Query execution	**Out of memory error** In some queries that involve Append-only indexes on large number of partition tables, these queries might fail with "out of memory" error. This issue has been resolved in this release.
12617	4.1.1.3	Query Optimizer	**Problems with queries with nest loop joins** When BitmapAppendOnlyPath is in the inner side of nested loop join, an incorrect query plan was generated causing queries to fail. This issue has been resolved in this release
11695	4.1.1.3	Performance/ Query Execution	**Error: “Too many files open”** In earlier versions of 4.0.x, some queries that contained multi-way HashJoin in the plan, errored out with a "too many files open" error when HashJoin needed to create a lot of temporary work files for its intermediate state. If you encounter this error, set the following GUC: `SET gp_workfile_type_hashjoin=buffile;` then re-run the query.
13461	4.1.1.2	Loaders	**Wildcard Character in External Table Name Caused a 404 Not Found Error** When loading an external table on Windows systems with `gpfdist`, if the file name contained a wildcard character, the system returned a 404 Not Found error. This issue has been resolved in this release.
13433	4.1.1.2	Loaders	**Bad Rows Might Cause Error** When accessing data with `gpfdist`, bad rows in the last file on the list might cause the system to return the error `segment reject limit reached`. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
13130, 13098	4.1.1.1	Segment Mirroring	**Segment Failures with Concurrent Activity** Segments failed with "stuck spinlock" error during concurrent IO activity on the database. This issue has been resolved in this release.
12868	4.1.1.0	PostgreSQL	**Executing more than one window function causes an error** In some prior releases, executing more than one window function would sometimes fail with the following error: `ERROR: Unexpected internal error` This issue has been resolved in this release.

12854
12765
12754
	4.1.1.0	Upgrade	**The Greenplum Database Upgrade from 4.0.4.0 to 4.1.0.0 Fails**
			The upgrade from 4.0.4.0 to 4.1.0.0 would sometimes not pass the memory limit to the upgraded version or fail with one of the following error:
			`ERROR: heap_inplace_update: wrong tuple length`
			`CRITICAL: Startup failed with error code 2`
			This issue has been resolved in this release.

12774	4.1.1.0	Query Planning	**BitmapAppendOnlyPath on the Inner Side of a Nested Loop Join Produces Invalid Query Plan** This issue has been resolved in this release.

12766	4.1.1.0	Transaction Processing	**Executing PREPARE TRANSACTION Causes System Failure**
			In some prior releases, executing a PREPARE TRANSACTION command returned the following warnings and error followed by a system failure.
			`WARNING: Greenplum Database detected segment failure(s), system is reconnected`
			`WARNING: Greenplum Database detected segment failure(s), system is reconnected`
			`ERROR: No primary gang allocated`
			This issue has been resolved in this release.

12645	4.1.1.0	Upgrade	**The Greenplum Database Upgrade from 4.0.5.1 to 4.1.0.0 Failed** The upgrade from 4.0.5.1 to 4.1.0.0 would sometimes fail on systems with the Greenplum Performance Monitor installed and returned the following error: `gpmigrator_mirror:fdwd-mdw0000:gpadmin-[CRITICAL]:-Error executing SQL` This issue has been resolved in this release.
12578	4.1.1.0	Table Management	**The gpexpand Utility Expands the Same Table Multiple Times** In some prior releases, tables with the same name in different namespaces were expanded multiple times. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
12577	4.1.1.0	Expanding Partitioned Tables in Multiple Schemas	**Systems with Partitioned Tables Experienced Very Slow Expansion** In systems with partitioned tables with the same name in multiple schemas, `gpexpand` issues multiple commands to redistribute the data for each partitioned table. This causes the expansion process to become very slow, as each table is expanded multiple times. For example, in the following setup schema_one.part_table_1 schema_two.part_table_1 schema_three.part_table_1 running `gpexpand` issues a total of nine `ALTER TABLE SET DISTRIBUTED BY` statements. This issue has been resolved in this release.
12573	4.1.1.0	Query	**The Explain Analyze Command Fails on Some Tables** In some prior releases, the Expand Analyze command would fail on some queries with the following error: `ERROR: Unexpected internal error` This issue has been resolved in this release.
12566	4.1.1.0	Log Messages	**Multiple Log Messages** On a mirrorless setup, FTS prints multiple log messages on the master log that segments have not reached the expected state. This issue has been resolved in this release.
12530, 12547	4.1.1.0	Unsuccessful create index, reindex, or vacuum	**Inconsistent Results after unsuccessful create index, reindex, or vacuum with a bitmap index** If the system crashes when running an index, reindex, or vacuum command with a bitmap index you might see inconsistent results in the data. This issue has been resolved in this release.
12499 12190	4.1.1.0	Query Execution	**ERROR: Type 19258 is not hashable** Certain data types such as `ip4r` are not fully supported in Greenplum Database because the query planner does not know how to hash them for distributed query operations. Joining on columns of such data types will give an error such as: `ERROR: Type 19258 is not hashable` This issue has been resolved in this release.
12495	4.1.1.0	Crash Recovery	**Crash Recovery Included Dropped Objects** Dropped objects were not removed from the file system during crash recovery. This issue has been resolved in this release.
11734	4.1.1.0	Memory Quota	When scanning a highly partitioned table with low reserved memory that was set by memory quota, an insufficient memory reserved error can appear. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
10028	4.1.1.0	Fault Detection and Recovery	**After Running gprecoverseg, Segments May Not Be Running in Their Preferred Role** In 4.x releases, when a primary segment goes down, the mirror is activated and becomes the primary segment. After running `gprecoverseg`, the currently active segment remains the primary and the failed segment is then brought up as the mirror. The segment instances are not returned to the preferred role that they were given at system initialization time. This can leave the system in a potentially unbalanced state, as some segment hosts may still have more active segments that is optimal for top system performance. After recovering a failed segment, the roles may be reversed (the original mirror is now the primary and vice-versa). The `gpstate -e` command will show you if you have segments in this condition. This issue has been resolved in this release.
12346	4.1.0.0	Data Loading	**COPY Error: No partition for partitioning key...** In prior releases, using the `COPY` command to load certain columns from standard input into a partitioned table would sometimes fail with the following error: `ERROR: no partition for partitioning key.` This issue has been resolved in this release.
12339	4.1.0.0	Backup and Restore	**Partitioned Table Names Different Following a Restore** In prior 4.0.x releases, the backup utilities (`gp_dump` and `gpcrondump`) would add extra quote characters to partitioned table names whenever the table names contained capital letters or special characters. After a restore operation, these partitioned table objects would be recreated with the extra quote characters in the table name, thereby changing the name of the table from what it was prior to backup/restore. This issue has been resolved in this release - the original table names are now restored as expected.
12287	4.1.0.0	Management Utilities	**Cannot Start Greenplum Database when a Segment is Down** In prior releases, the Greenplum Database startup utility, `gpstart`, failed to start Greenplum Database if a segment was not available instead of starting the system with the available mirror copies. This issue has been resolved in this release.
12203	4.1.0.0	Query Execution / Append-Only Tables	**ERROR: Bad append-only storage header...** In prior 4.0.x releases, the following error could sometimes cause a query to fail when multiple concurrent queries were accessing the same append-only table: `ERROR: Bad append-only storage header. Header check error 1, detail 'Append-only storage header is invalid...` This issue has been resolved in this release.
12124	4.1.0.0	Fault Detection and Recovery	**New Server Configuration Parameter for Fault Detection Timeout** In prior releases, the interconnect timeout and the fault detection timeout were controlled by the same parameter (`gp_segment_connect_timeout`). Having too small of a fault detection timeout increased segment failures on busy systems. A new parameter has been added (`gp_fts_probe_timeout`) that allows users to control the fault detection timeout separately. The default is 3 minutes.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
12105	4.1.0.0	Fault Detection and Recovery	**Intermittent Network Outages Causing Segment Failures** In prior 4.0.x releases, intermittent network outages could prevent connections between the master and segments, thereby causing segment failures. In this release, the master will now retry a connection if the first attempt fails, thereby decreasing the likelihood of segment failures.
12103	4.1.0.0	Query Planning and Dispatch (JDBC)	**Error When Using JDBC: "Statement Error, writer gang busy..."** In prior 4.0.x releases, certain SELECT and DELETE statements would fail with the following error when using the JDBC driver: `Statement Error, writer gang busy: possible attempt to execute volatile function in unsupported context...` This issue has been resolved in this release.
12028	4.1.0.0	Workload Management	**Function gp_adjust_priority() Does Not Impact CPU Utilization** In prior 4.0.x releases, using the `gp_adjust_priority()` function to change the priority of an active query would change the query's priority, however CPU utilization and query runtime was not adjusted as expected. This issue has been resolved in this release.
11999	4.1.0.0	Workload Management	**Functions Not Evaluated Against Resource Queue Limits** In prior releases, resource queues did not evaluate queries that executed functions against the limits of the resource queue. Functions containing `SELECT` statements were allowed to run, even if they exceeded the resource queue limits. This issue has been resolved in this release.
11986	4.1.0.0	Backup and Restore	**Cannot Restore a Single Table Using gpdbrestore Parallel Restore Utility** In prior releases, when using the parallel restore utility `gpdbrestore` with the `-T` option to restore a single table, data would not always be restored to the target table as expected. This issue has been resolved in this release. **Note:** When using the `-T` option, `gpdbrestore` does not truncate the table before restoring the data from the backup. If your intention is to replace existing data in the table from backup, truncate the table prior to running `gpdbrestore`.
11985	4.1.0.0	Backup and Restore	**Cannot Restore a Single Table Using gpdbrestore Parallel Restore Utility** In prior releases, when using the parallel restore utility `gpdbrestore` with the `-T` option to restore a single table, data would not always be restored to the target table as expected. This issue has been resolved in this release. **Note:** When using the `-T` option, `gpdbrestore` does not truncate the table before restoring the data from the backup. If your intention is to replace existing data in the table from backup, truncate the table prior to running `gpdbrestore`.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
11916	4.1.0.0	Backup and Restore	**Error Restoring Tables with Triggers** In prior releases, when restoring a table from backup that had triggers associated with it, the restore would fail during `COPY` with the following error (even when the table was not an append-only or column-oriented table): `ERROR: AOCS Does not do trigger yet (copy.c:4007)` This issue has been resolved in this release by 1) allowing restore of heap tables that have user-defined triggers, and 2) providing a better error message when the table is an append-only or column-oriented table. **Note:** Although Greenplum Database does not fully support user-defined triggers, certain uses of triggers on heap storage tables have been verified for specific customers by Greenplum Technical Support. Triggers are disabled in Greenplum Database for append-only (AO) storage tables.
11891	4.1.0.0	Data Loading	**Unhelpful gpfdist Error Message Upon Load Failure** In prior releases, when loading data using the `gpfdist` parallel file distribution server, the following error message would display upon load failure: `[INTERNAL ERROR gpfdist.c:1575] cannot listen all the time...` The `gpfdist` parallel file distribution server has been enhanced in this release to print more diagnostic information in its error messages providing administrators with more information in diagnosing the cause of load failures.
11843	4.1.0.0	Fault Detection and Recovery	**Primary Segment Failure Does Not Activate Mirror** In prior 4.0.x releases, a system crash (such as a power failure) could result in a failed segment being unable to transition operations to its mirror. When this occurred, segment logs would have a messages such as: `PANIC","58P01","could not open control file ""global/pg_control"": No such file or directory...` When this error occurred, Greenplum Database was unable to restart. This issue has been resolved in this release.
11841	4.1.0.0	Transaction Management	**FATAL: the limit of xx distributed transactions has been reached** In prior 4.0.x releases, if a segment failure occurred while many concurrent `COPY` or load operations were running, additional connections would sometimes fail with the following error: `FATAL: the limit of xx distributed transactions has been reached...` This issue has been resolved in this release.
11800	4.1.0.0	Query Execution	**EXPLAIN ANALYZE Not Printing Detailed Statistics on Large Queries** In prior releases, the `EXPLAIN ANALYZE` command would not always print out detailed statistics (such as memory usage, spill files, timing, and so on) for queries involving very large data sets. This issue has been resolved in this release.
11773	4.1.0.0	Data Loading	**gpload Error on Tables with Added or Dropped Columns** In prior releases, data loads would fail with the following error if the target table had ever been modified using `ALTER TABLE...ADD COLUMN` or `ALTER TABLE...DROP COLUMN`: `ERROR: syntax error at or near "-"` This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description
11752, 11886	4.1.0.0	Query Execution	**Information in Stack Dump and Core Dump Files Could Become Unreadable** In rare cases, certain SIGSEGV and operating system related errors could cause stack and core dump files to become unusable. This prevented the diagnosis of the underlying cause of the SIGSEGV error and/or other conditions in the Greenplum Database environment. This issue has been resolved in this release.
11743	4.1.0.0	Fault Detection and Recovery	**Unrecoverable Segment After a System Crash** In prior 4.0.x releases, there was an issue that prevented a primary segment and its mirror from resynchronizing following a system crash (such as a power failure). This issue typically occurred when several concurrent transactions were updating an append-only (AO) table at the time of the failure. If this issue was encountered, administrators were unable to recover failed segments using the `gprecoverseg` utility, and the `gpstop` /`gpstart` utilities were unable to stop/restart the failed segments. Symptoms of this issue included errors such as the following in the Greenplum Database log files: `"PANIC","XX000","Append-Only Mirror Resync EOFs intent count would go negative..."`
11732	4.1.0.0	Upgrade	**gpupgrademirror Errors when Upgrading Greenplum Database from 3.3.x to 4.0.x** When running the Greenplum Database upgrade utility (`gpmigrator`), the `gpupgrademirror` sub-utility uses the system catalog tables to determine the database files it needs to copy from each primary segment to its respective mirror segment. In rare cases, index file records in the catalog did not match the actual index files found on the segments. When this type of mismatch occurred and an expected index file did not exist on a segment, users received the following error message: `Exception: There was a problem with one of the gpupgrademirror sub processes.` This issue has been resolved in this release.
11728	4.1.0.0	DML/DDL	**Slow TRUNCATE Performance on Large Objects** In prior 3.3.x and 4.0.x releases, users experienced decreasing performance when truncating several large tables in succession. This issue has been resolved in this release.
11727	4.1.0.0	Fault Detection and Recovery	**NIC Failure on Master Host Causes Greenplum Database to Become Unresponsive** In prior releases, if the primary network interface (NIC) used for interconnect traffic was disabled on the master host, the fault detection process (`ftsprobe`) was not able to connect back to the master resulting in unresponsive queries. When this occurred, the master log file would have messages such as: `"FATAL","XX000","FTS: setting segDB state failed, error connecting to entry db..."` This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.x.x

Issue Number	Resolved In	Category	Description

11713	4.1.0.0	Fault Detection and Recovery	**Failed Segment Unable to Failover to Mirror**
			In prior 4.0 releases, certain heavy load conditions could cause primary-to-mirror segment communications to become unresponsive. When this occurred, Greenplum Database was unable to transition operations over to the mirror when a primary segment failed, leaving Greenplum Database in an unresponsive state. This release introduces a fix to the primary-to-mirror segment communication processes to prevent this issue in the future.
			In addition to this fix, customers experiencing this issue should also set the `tcp_keepalives` server configuration parameters to the recommended values by running the following series of commands on the master host:
			`$ gpconfig -c tcp_keepalives_idle -v 180`
			`$ gpconfig -c tcp_keepalives_count -v 2`
			`$ gpconfig -c tcp_keepalives_interval -v 75`
			`$ gpstop -r`

11707, 11006	4.1.0.0	Management Utilities	**Cannot Start Greenplum Database when a NIC is Down** In prior releases, the Greenplum Database startup utility, `gpstart`, failed to start Greenplum Database if a segment host had a failed network interface (NIC). When this occurred, the `gpstart` utility became unresponsive. This issue has been resolved in this release.
11696	4.1.0.0	Fault Detection and Recovery	**gprecoverseg: Multi-Segment Recovery Process Unresponsive** In prior releases, in cases where there were multiple segment primary/mirror pairs to recover (such as in a multi-host failure), the ordering of the segment recovery could prevent a primary and its corresponding mirror from establishing a connection to each other in the allowed timeout. When this occurred, the `gprecoverseg` utility became unresponsive and segments were not recovered as expected. This issue has been resolved in this release.
10848	4.1.0.0	Management Utilities	**Management Utilities Should Place Mirror on Different Subnet than its Primary** In prior releases, the Greenplum Database management utilities such as `gpinitsystem`, `gpaddmirrors` and `gpexpand` did not always place a primary and its corresponding mirror segment on different subnets (interconnect networks) to ensure network switch redundancy. This issue has been resolved in this release.
6571	4.1.0.0	Performance Monitor	**Missing or Incorrect Performance Monitor System Metrics** In prior releases of Greenplum Performance Monitor, some rows in the `system_history` table would have incorrect numbers or missing data for the `net_rb_rate`, `net_wb_rate`, `net_rp_rate` and `net_wp_rate columns`. This issue has been resolved in this release.
6567	4.1.0.0	Data Loading	**gpload Error when Using password Authentication** In prior releases, the `gpload` utility did not support `password` authentication (only `md5`). When password authentication was used, `gpload` would fail with the following error: `ERROR|unexpected error - backtrace written to log file` This issue has been resolved in this release - password authentication is now supported.

Known Issues in Greenplum Database 4.1.1.8

This section lists the known issues in Greenplum Database 4.1.1.8. A work-around is provided where applicable.

Table 2 All Known Issues in 4.1.1.8

Issue	Category	Description
N/A	Performance Monitor	Performance Monitor 4.1.1.8 asks for IPv6 support during setup. IPv6 is not supported by Greenplum Database 4.1.x IPv6 is supported by Greenplum Database 4.2. **Workaround**: Enter No to continue in the IPv4 environment.
11297	Query Optimizer	**Aggregate functions without Preliminary functions slowing query execution** In SQL, an aggregate function definition can optionally be supplemented with a preliminary function that describes how intermediate results are combined. Not providing a preliminary function in the definition can in some circumstances affect performance. The causes of the loss of performance are: • The planner may choose GroupAgg instead of HashAgg • More data may need to be moved between nodes. **Workaround:** If you encounter slow query execution when using aggregate functions, add a preliminary function to the definition.
14219	Documentation	**Incorrect default batch size for gpexpand in documentation** The default batch size (-B option) for the gpexpand utility is incorrectly documented in the help file and the Greenplum Administrator Guide as 8. The actual default batch size is 16.
12919	Loading Data	**Using TRUNCATE option with MERGE or UPDATE loses data** If `TRUNCATE` is set to true and the mode is `UPDATE` in the `YAML` file, all existing data is wiped out, and nothing is updated when `gpload` is executed. All data in the target table is lost, and the target table is empty. If `TRUNCATE` is set to true and the mode is `MERGE` in the `YAML` file, all existing records are wiped out when `gpload` is executed, and all records in the source file are inserted. The target table only contains the new records, but all the previously existing data is lost.
12917	Loading Data	**Table Names Cannot Contain Special Characters in the YAML File** If a table name contains a special character or space and the `YAML` configuration file sets `REUSE_TABLES` to `true`, the `gpload` utility fails.
12915	Loading Data	**Count of Data Formatting Errors Is Incorrect** If an error tolerance is set in a `YAML` file, errors are successfully written to an error table, but the log and output messages incorrectly display zero errors, as shown below: `data formatting errors = 0`

12846	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`

Table 2 All Known Issues in 4.1.1.8

Issue	Category	Description
12823	Backup and Restore	**Cannot Restore Functional Index and Functional Trigger** When performing a partial schema backup and restore, such as gp_dump using the -t option to dump only the selected tables, a function index or functional trigger cannot be restored, since the underlying functions are not missing from the dump files. **Workaround: Restore the required functions separately, prior to restoring the tables than contain a functional index or functional trigger**.
12788	Recovery	**Mirror Location Not Validated** The `gprecoverseg` utility does not validate the input file for mirrors on same host as the primary segment.
12646	Loaders	**MATCH_COLUMNS requires declared distribution key columns** For update or delete operations using `gpload`, the `MATCH_COLUMNS` section in the YAML file still requires all distribution key columns to be declared.
12590	Email Alerts	**Invalid SMTP Server for Email Alerts Causes Degraded Performance** When you configure the Greenplum Database to send email alerts, you must supply a value for the parameter `gp_email_smtp_server`. If the specified email server is invalid or unreachable from the Greenplum master host, then Greenplum Database can experience degraded performance or become unresponsive. **Work-around:** If you notice degraded performance after configuring email alerts, either disable the email alerting feature, or make sure that the specified email server is valid and accessible from the Greenplum master host.
12472	Memory	**No Available Memory** In very rare circumstances, the Greenplum Performance Monitor process `gpmonws` can use all available memory on the master node. **Work-around:** Restart the Greenplum Performance Monitor.
12334, 12604	Fault Detection and Recovery	**Shared Memory Error Can Cause Some Segment Mirrors to Become Unresponsive** In very rare circumstances, an issue with shared memory in the Greenplum Database can result in the following error: `"FATAL","XX000","semop(id=21495876,num=13) failed: Numerical result out of range (pg_sema.c:499)"` When this error occurs, mirror segments can become unresponsive, and the system goes into change-tracking mode. **Work-around:** To resolve this issue, run the `gprecoverseg` utility.
12288	System Catalogs and Metadata	**gpcheckcat Error: gp_persistent_* state check found** The `gpcheckcat` utility is used to detect potential inconsistencies in the system catalog tables of the master and segments. In certain rare circumstances, an inconsistency to the `gp_persistent_*` catalog tables may be introduced if a segment experiences a system outage (such as a power failure) during certain `CREATE` or `DROP` operations. When this occurs, the `gpcheckcat` utility will detect and report errors. `[ERROR]:-gp_persistent_tablespace_node state check found` **Work-around**: Contact EMC Technical Support.

Table 2 All Known Issues in 4.1.1.8

Issue	Category	Description
12250	Data Loading	**Connections to Greenplum Database Remain Open After an ETL Operation** On rare occasions while performing an ETL operation using a third-party ETL tool, the following error might display: `"53300","sorry, too many clients already.","There are no more available slots in the sharedSnapshotArray..."` This indicates that there are no more available client connection slots to the database. This issue is caused by previous ETL operations not properly closing their connections to Greenplum Database. **Work-around**: Manually close the open connections.
12233	Query Execution / Partitioning	**DEFAULT Partition Always Scanned Regardless of Query Predicate** The default partition of a partitioned table is always scanned during query execution, even when the query predicate has explicit conditions that match other partitioned tables.This may impact query performance when the default partition contains significant amounts of data. **Work-around:** After loading data into a partitioned table, make sure that any data landing in the default partition is split out into a new partition instead of remaining in the default partition.
12192	Query Execution	**Sub-optimal Performance when Joining on Columns of Different Data Types** When you join on columns of different datatypes, the query planner implicitly casts the columns to a common datatype before it can execute the join. In some cases, this implicit casting causes the query planner to choose a sub-optimal query plan, thereby increasing query execution time. **Work-around**: Columns that are joined frequently should be of the same datatype. If this is not possible, explicitly cast joined columns to the same datatype as part of your SQL statement.
12165, 12173	Query Execution	**Queries with Multi-row Subqueries Are Not Supported** Currently, Greenplum Database does not support queries that contain multi-row subqueries. **Work-around:** Rewrite to query statement to avoid using multi-row subqueries.
12120	Installation	**Greenplum Database directories show the incorrect group for the owner user if the owner name and the group name are different** The installation package assumes 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 is incorrect. **Work-around:** Change the group ownership manually by running `chgrp`.
12099	DDL / Partitioned Tables	**Using ALTER TABLE...DROP COLUMN to Alter the Schema of a Partitioned Table Causes Errors** Currently, adding additional partitions to the hierarchy after altering the table design by using `ALTER TABLE...DROP COLUMN` can cause those new partitions to become corrupted. In this scenario, the following error message might appear: `ERROR: distribution policy for "new_partition" must be the same as that for "parent_partition"` **Work-around:** If you need to drop columns on a partitioned table and want to avoid this issue, run `ALTER TABLE...DROP COLUMN`, `CTAS` the partitioned table, `DROP` the old table, and rename the new table to the old table name.

Table 2 All Known Issues in 4.1.1.8

Issue	Category	Description
12082	Query Execution	**Error When GROUP BY and a SELECT Clauses Refer 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 can occur when a `GROUP BY` clause refers to a column in one way and a `SELECT` clause refers to it in a different way. **Work-around**: Supply an alias for the join result (either in `ANSI JOIN` syntax in the `FROM` clause or by expressing the `JOIN` as a subquery in the `FROM` clause), and refer to columns by this alias for clauses outside the `FROM` and `WHERE` clauses of the query.
11690	Fault Detection and Recovery	**Greenplum Database Self-Healing Process Can Create Zero Byte Files** When Greenplum Database automatically recovers from errors and other system problems (self-healing), the following error message might display when accessing the database: `"Could not read from file ""pg_clog/0004"` This occurs because, in rare cases, zero byte files are created during the self-healing process. **Work-around**: Contact EMC Technical Support.
10278	System Catalogs and Metadata	**pg_relation_size and pg_database_size do not Account for Custom Filespaces** The `pg_relation_size` and `pg_database_size` functions are used to calculate the size of a relation or database in a distributed Greenplum Database system. These functions do not currently account for relations stored in any filespaces besides the default `pg_system` filespace.

9968	Backup and Restore	**Slow gp_dump / gpcrondump Performance**
		Prior to executing a dump operation, the dump utilities need to look up metadata information in the system catalogs. On databases with thousands of tables and millions of column attributes, this portion of the dump operation can take a long time.
		Work-around: Run `VACUUM ANALYZE` on the system catalog tables before running a dump operation. Prior to running a dump operation, run the following commands for the role that is executing the dump utilities. For example:
		`ALTER ROLE gpadmin SET enable_nestloop = on;`
		`ALTER ROLE gpadmin SET random_page_cost = 10;`
		After the dump is complete, return to the default settings:
		`ALTER ROLE gpadmin RESET enable_nestloop;`
		`ALTER ROLE gpadmin RESET random_page_cost;`

8445	DDL	**CREATE DATABASE ERROR: "template1" is being accessed by other users** When the system has segments in `resynchronizing` mode, administrators will not be able to create a new database. Running a `CREATE DATABASE` command when segments are in the process of resynchronizing will result in the following error: `ERROR: source database "template1" is being accessed by other users...` **Work-around:** Use `gpstate -e` to check the status. When all segments are `synchronized`, you may retry the `CREATE DATABASE` command.

Table 2 All Known Issues in 4.1.1.8

Issue	Category	Description
6279	DDL	**Append-Only Table Error After ALTER TABLE...DROP COLUMN** After altering an append-only table to drop a column, users encounter the following error when trying to access the table: `ERROR: cache lookup failed for type 0` **Work-around:** If you need to drop a column from an append-only table, do not use `ALTER TABLE`. Instead recreate the append-only table with the desired columns using `CREATE TABLE ...AS SELECT`.
5647	Upgrade	**Upgrade Utility Requires 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 are not recognized as valid by `gpmigrator`, and cannot 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.
5517	DDL	**Deadlock Detected when Concurrently Altering Tables with Bitmap Indexes** Concurrent `ALTER TABLE` commands on tables that have bitmap indexes may encounter a deadlock and be unable to proceed. This issue occurs with concurrent `ALTER TABLE` commands that set distribution policies on tables that have bitmap indexes. **Work-around:** if you need to run concurrent `ALTER TABLE` commands on tables with bitmap indexes (as when running `gpexpand` with `-n` for multiple parallel processes), first drop the bitmap indexes and reinstate them when the concurrent `ALTER TABLE` commands are completed.

5517, 3213	Query Execution	**Transaction Within a Function Not Recognized as a Sub-Transaction**
		When a function containing multiple transaction blocks is run and an error occurs in one transaction block, the entire function exits with the errors:
		`ERROR: The distributed transaction 'Prepare' broadcast failed to one or more segments`
		`ERROR: current transaction is aborted, commands ignored until end of transaction block`

1589	System Catalogs and Metadata	**PostgreSQL Usage Statistics Views and Functions do not Work as Expected** PostgreSQL has a number of views (`pg_stat_*`, `pg_statio_*`) for showing usage statistics. All of these views only report on the usage of the master (system catalogs), not the usage of user data on the segments. Many of the PostgreSQL statistics functions have the same problem. For example, `pg_stat_get_tuples_inserted()` shows only those inserts into the master (usually 0), not the number inserted into the table in question.

Upgrading to Greenplum Database 4.1.1.x

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

- Upgrading from 4.1.0.x to 4.1.1.x
- Upgrading from 4.0.x.x to 4.1.1.x
- Upgrading From 3.3.x.x to 4.1.1.x
- Server Configuration Parameters Changes

Upgrading from 4.1.0.x to 4.1.1.x

These instructions are for users currently running 4.1.0.x. An upgrade from 4.1.0.x to 4.1.1.x 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 - gpadmin
   ```

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

   ```
   $ gpinitstandby -r
   ```

3. Perform a smart shutdown of your current Greenplum Database 4.1.0.x system (there can be no active connections to the database):

   ```
   $ gpstop
   ```

4. Run the installer for 4.1.1.8 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-4.1.1.8`

5. Edit the environment of the Greenplum Database superuser (`gpadmin`) and make sure you are sourcing the `greenplum_path.sh` file for the new installation. For example change the following line in `.bashrc` or your chosen profile file:

   ```
   source /usr/local/greenplum-db-4.1.0.0/greenplum_path.sh
   ```

   to:

   ```
   source /usr/local/greenplum-db-4.1.1.8/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-4.1.1.8
   /usr/local/greenplum-db
   ```

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

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

7. Log in as root and install the 4.1.1.8 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 gpadmin 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-4.1.1.8
# gpscp -f seghostname_file gp.tar =:/
# gpssh -f seghostname_file
   => gtar --directory / -xvf /gp.tar
   => chown -R gpadmin /usr/local/greenplum-db-4.1.1.8
   => rm /gp.tar
   => rm /usr/local/greenplum-db
   => ln -s /usr/local/greenplum-db-4.1.1.8
   /usr/local/greenplum-db
```

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

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

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

```
$ gpinitstandby -s standby_hostname
```

Note: If you want to rollback to the previous version of Greenplum Database, stop the database (gpstop) and change the symbolic link to refer to the previous installation. Then, restart the database (gpstart).

10. Make sure you have the latest versions of Greenplum Performance Monitor, Greenplum Client, Loader, and Connectivity installed. See "Greenplum Database Client Tool Packages and Performance Monitor" on page 26 for the latest version number. Installation packages are available for download from the EMC Download Center.

    To update the Greenplum Performance Monitor, install the new 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 4.1.1.x installation). Greenplum Performance Monitor 4.1.1 Web Application Packages are available at the EMC Download Center. See the *Greenplum Performance Monitor 4.1 Administrator Guide* for additional installation information.

Upgrading from 4.0.x.x to 4.1.1.x

For detailed upgrade procedures, see the following sections:

- Upgrade Procedure

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.0.x.x to 4.1.1.x. Detailed upgrade instructions are also provided in the Upgrade Procedure section.

Pre-Upgrade Preparation (on your current system)

* 4.0.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 4.1.1.x binaries on all Greenplum hosts.
- □ Copy any custom modules from your existing installation to your 4.1.1.x installation on all Greenplum hosts. For example, shared library files for user-defined functions in `$GPHOME/lib` or PostgreSQL add-on modules (such as `plr.so` or `pgcrypto.so`) in `$GPHOME/lib/postgresql`.
- □ 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 4.1.1.x installation.
- □ Run the upgrade utility (`gpmigrator_mirror` if you have mirrors, `gpmigrator` if you do not).
- □ After the upgrade process finishes successfully, your 4.1.1.x system will be up and running.

Post-Upgrade (on your 4.1.1.x system)	
* The 4.1.1.x system is up	
□	Reinitialize your standby master host (`gpinitstandby`).
□	Upgrade `gpfdist` on all of your ETL hosts by installing the compatible version 4.1.1.x Load Tools package.
□	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.
□	Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.1.1.x installation (if necessary).

Upgrade Procedure

This section divides the upgrade into three 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 4.0.x system)

Perform these steps on your current 4.0.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. (*optional*) Run the `gpcheckcat` utility to validate your system catalogs. For example:

   ```
   $ $GPHOME/bin/lib/gpcheckcat
   ```

   If any errors are reported, contact Greenplum Customer Support.

5. Run `gpstate` to check for failed segments.

   ```
   $ gpstate
   ```

6. 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-4.1.1.x-PLATFORM.zip
   ```

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

   ```
   # /bin/bash greenplum-db-4.1.1.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-4.1.1.8`), 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 4.1.1.x installation. For example:

```
$ source /usr/local/greenplum-db-4.1.1.8/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
   ```

10. Copy any custom modules from your existing 4.0.x.x installation to your 4.1.1.x installation on all Greenplum hosts. For example, shared library files for user-defined functions in `$GPHOME/lib` or PostgreSQL add-on modules (such as `plr.so` or `pgcrypto.so`) in `$GPHOME/lib/postgresql`.

Upgrade Execution

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

11. Source the path file from your old 4.0.x.x installation. For example:

    ```
    $ source /usr/local/greenplum-db-4.0.4.0/greenplum_path.sh
    ```

12. (*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.

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

    ```
    $ gpinitstandby -r
    ```

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

    ```
    $ gpstop
    ```

15. Source the path file from your new 4.1.1.x installation. For example:

    ```
    $ source /usr/home/greenplum-db-4.1.1.8/greenplum_path.sh
    ```

16. Update your environment so that it is sourcing your new 4.1.1.x installation.

    a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.1.1.x installation directory. For example (as root):

       ```
       # rm -rf /usr/local/greenplum-db
       # ln -s /usr/local/greenplum-db-4.1.1.8
       /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-4.1.1.8
       /usr/local/greenplum-db
       => chown -R gpadmin /usr/local/greenplum-db
       ```

```
    => exit
```

17. As `gpadmin`, run the 4.1.1.x version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ su - gpadmin
$ gpmigrator_mirror /usr/local/greenplum-db-4.0.4.0
/usr/local/greenplum-db-4.1.1.8
```

18. The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.1.1.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 4.1.1.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 4.1.1.x Load Tools package. Application Packages are available at the EMC Download Center.

21. If you are using the Greenplum Performance Monitor, install the new 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 4.1.1.x installation). Greenplum Performance Monitor 4.1.1.x Web Application Packages are available at the EMC Download Center.

22. See "Server Configuration Parameters Changes" on page 25 for parameter changes introduced in 4.1.0.0. Check your `postgresql.conf` server configuration files for any settings that should be removed or changed. You can use the `gpconfig` utility to check and set parameters.

Troubleshooting a Failed Upgrade

If you experience issues during the migration process, go to the Support page at http://powerlink.emc.com 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`)

Upgrading From 3.3.x.x to 4.1.1.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.1.1.x. You must first 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 http://powerlink.emc.com). After you have successfully upgraded to 4.0.x.x, follow the upgrade instructions for Upgrade Checklist.

Server Configuration Parameters Changes

The following sections list the Server Configuration Parameters that changed in Greenplum Database 4.1.0.0 and may need to be updated during an upgrade from 4.0.x.x.

- Deprecated Server Configuration Parameters
- New Server Configuration Parameters in 4.1.0.0

Deprecated Parameters

Table 3 Deprecated Server Configuration Parameters

Parameter Name	Description of Change
maintenance_work_mem	If using the new memory management features of 4.1.x.x (`gp_resqueue_memory_policy=auto`), these parameters are ignored. `max_statement_mem` and `statement_mem` are used instead to control per-query memory usage.
max_work_mem	
work_mem	

New Parameters

Table 4 New Server Configuration Parameters in 4.1.0.0

Parameter Names	Description
application_name	Sets the application name for a client session. For example, if connecting via `psql`, this parameter will be set to `psql`. Setting an application name allows it to be reported in log messages and statistics views.
gp_connectemc_mode	Controls the ConnectEMC event logging and dial-home capabilities of Greenplum Performance Monitor on the EMC Greenplum Data Computing Appliance (DCA). ConnectEMC must be installed in order to generate events. Allowed values are: • `on` (the default) - log events to the `gpperfmon` database and send dial-home notifications to EMC Support • `off` - turns off ConnectEMC event logging and dial-home capabilities • `local` - log events to the `gpperfmon` database only
gp_fts_probe_timeout	Sets the allowed timeout for the fault detection process (`ftsprobe`) to establish a connection to a segment instance. The default is 3 minutes. In prior releases, the interconnect timeout and the fault detection timeout were controlled by the same parameter (`gp_segment_connect_timeout`).
gp_resqueue_memory_policy	Enables the new memory management features of 4.1. When set to `none`, memory management is the same as in 4.0 (using `work_mem`). When set to `auto`, query memory usage is controlled by `statement_mem` and/or resource queue memory limits. The `work_mem` parameter becomes obsolete when this is enabled.

Table 4 New Server Configuration Parameters in 4.1.0.0

Parameter Names	Description
gp_workfile_checksumming	Adds a checksum value to each block of a work file (or spill file) used by `HashAgg` and `HashJoin` query operators. This adds an additional safeguard from faulty OS disk drivers writing corrupted blocks to disk. When a checksum operation fails, the query will cancel and rollback rather than potentially writing bad data to disk. Default is `on`.
max_statement_mem	Sets the maximum memory limit for a query. This parameter can only be set by database superusers. It prevents regular users from oversubscribing `statement_mem`. Default is 2000MB.
statement_mem	Sets a per-query memory limit. When `gp_resqueue_memory_policy` is enabled, this parameter replaces `work_mem`. Default is 125MB.
smdw_aliases	This is a new parameter added to the Greenplum Performance Monitor configuration file `gpperfmon.conf` (not `postgresql.conf`) that allows the monitor agents to use multiple interfaces to the standby master host (smdw). For example: `smdw_aliases= smdw-1,smdw-2`

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 from the EMC Download Center.

Table 5 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]	4.1.1.0	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4 4.1.1.3 4.1.1.2 4.1.1.1 4.1.1.0
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.1.1.0	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4 4.1.1.3 4.1.1.2 4.1.1.1 4.1.1.0

Table 5 Greenplum Database Performance Monitor and Client Tools Version Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.1.1.3 (windows platforms only)	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4 4.1.1.3 4.1.1.2 4.1.1.1 4.1.1.0
		4.1.1.2 (available platforms only)	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4 4.1.1.3 4.1.1.2 (available platforms only) 4.1.1.1
		4.1.1.0	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4 4.1.1.3 4.1.1.2 4.1.1.1 4.1.1.0
Greenplum Database Performance Monitor Greenplum Database monitor for query and system performance		4.1.1.8	4.1.1.8
		4.1.1.4	4.1.1.8 4.1.1.7 4.1.1.6 4.1.1.5 4.1.1.4
		4.1.1.3	4.1.1.3
		4.1.1.0	4.1.1.2 4.1.1.1 4.1.1.0

[1] gpmapreduce is not available on Windows.

Greenplum Database 4.1.1.8 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 6 Greenplum Database Documentation

Title	Part Number	Revision
Greenplum Database 4.1 Installation Guide	300-012-429	A02 - revised
Greenplum Database 4.1 Administrator Guide	300-012-428	A03 - revised
Greenplum Database Performance Monitor 4.1 Administrator Guide	300-012-430	A02 - revised
Greenplum Database 4.1 Client Tools for UNIX	300-012-432	A01
Greenplum Database 4.1 Client Tools for Windows	300-012-433	A01
Greenplum Database 4.1 Connectivity Tools for UNIX	300-012-434	A01
Greenplum Database 4.1 Connectivity Tools for Windows	300-012-435	A01
Greenplum Database 4.1 Load Tools for UNIX	300-012-436	A01
Greenplum Database 4.1 Load Tools for Windows	300-012-437	A01
Greenplum Database 4.1.1.8 Release Notes	300-013-342	A01