Greenplum
The Data Computing Division of EMC

EMC²
where information lives®

Welcome to Greenplum Database 4.0.5.0

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.

Greenplum Database 4.0.5 is the fifth service pack release for version 4.0. This service pack release includes the contents of 4.0.1.x, 4.0.2.x, 4.0.3.x and 4.0.4.x plus additional new issue resolutions and minor feature enhancements. Please refer to the following sections for more information about this release:

- Resolved Issues in Greenplum Database 4.0.5.0
- Known Issues in Greenplum Database 4.0.5.x
- Upgrading to Greenplum Database 4.0.5.0
- Greenplum Database 4.0 Client Tool Packages
- Greenplum Database Documentation

For the contents of previous 4.0.x releases, please see the release notes section of each corresponding release:

- Greenplum Database 4.0.4.x Release Notes
- Greenplum Database 4.0.3.x Release Notes
- Greenplum Database 4.0.2.x Release Notes
- Greenplum Database 4.0.1.x Release Notes

Feature Enhancements in 4.0.5.0

This section describes minor feature enhancements to Greenplum Database management utilities and server configuration parameters introduced in 4.0.5.0. Refer to the *Greenplum Database 4.0 Administrator Guide* for detailed descriptions of items listed in this section.

Server Configuration Parameters

Table 1 Server Configuration Parameter Enhancements in 4.0.5.0

Parameter Name	Description of Change
gp_fts_probe_timeout	New. Sets the allowed timeout for the fault detection process (`ftsprobe`) to establish a connection to a segment instance, and to receive information regarding the segment's current status. 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_segment_connect_timeout	Changed behavior. In releases prior to 4.0, this parameter controlled the connection timeout between the master and primary segments. In 4.0 and later releases, this parameter also controls the connection timeout between primary and mirror replication processes. Because of this change, this parameter is now classified as a *local* parameter, meaning it must now be set in the `postgresql.conf` files of the master and all segments (primary and mirror).
gp_resqueue_priority	Changed default to ON, which enables query prioritization for resource queues. In releases prior to 4.0.5.0, this parameter was OFF by default.
gp_resqueue_priority_cpucores_per_segment	Changed default to 4 on the segments and 24 on the master, which are the appropriate default settings for the Greenplum Data Computing Appliance.
gp_resqueue_priority_sweeper_interval	Changed default to 1000 (miliseconds), which is the appropriate default setting for the Greenplum Data Computing Appliance.

Management Utilities

Table 2 Management Utility Enhancements in 4.0.5.0

Management Utility Name	Description of Change
gpseginstall	New. The `gpseginstall` utility allows system administrators to install the Greenplum Database software on multiple hosts at once. When run as `root`, it can also automate other new installation system configuration tasks such as creating the Greenplum system user (`gpadmin`), setting the system user's password, setting the ownership of the Greenplum Database installation directory, and exchanging ssh keys between all specified host address names.
gpstop	New option: `-t <timeout_in_seconds>` When stopping Greenplum Database using `gpstop`, a segment may not complete shutdown in the default timeout of 60 seconds, especially if there are large transactions to roll back. To avoid shutdown errors, a new `-t` option was added to `gpstop` to allow the user to increase the allowed timeout.

Resolved Issues in Greenplum Database 4.0.5.0

This section lists the customer reported issues that are now resolved in Greenplum Database 4.0.5.x:

Table A.1 Resolved Issues in 4.0.5.x

Issue Number	Resolved In	Category	Description
12287	4.0.5.0	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.
12339	4.0.5.0	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.
12203	4.0.5.0	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	4.0.5.0	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.
12105	4.0.5.0	Fault Detection and Recovery	**Intermittent Network Outages Caused Master to Falsely Report Segment Failures** In prior 4.0.x releases, intermittent network-related communication failures between the master and a segment resulted in the master marking the segment as down. In this release, the master will now retry a connection if the first attempt fails (up to five retries). This will decrease the likelihood of the master incorrectly reporting a segment as failed when intermittent/temporary networking issues occur.
12103	4.0.5.0	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.

Table A.1 Resolved Issues in 4.0.5.x

Issue Number	Resolved In	Category	Description
12028	4.0.5.0	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	4.0.5.0	Workload Management	**Functions Are Not Restricted by Resource Queue Limits** In prior releases, resource queues did not restrict 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	4.0.5.0	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`.
11916	4.0.5.0	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 not supported for append-only (AO) storage tables.
11891	4.0.5.0	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.

Table A.1 Resolved Issues in 4.0.5.x

Issue Number	Resolved In	Category	Description
11843	4.0.5.0	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	4.0.5.0	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	4.0.5.0	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	4.0.5.0	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.
11752, 11886	4.0.5.0	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	4.0.5.0	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..."`

Table A.1 Resolved Issues in 4.0.5.x

Issue Number	Resolved In	Category	Description

11732	4.0.5.0	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	4.0.5.0	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	4.0.5.0	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.

11713	4.0.5.0	Fault Detection and Recovery	**Failed Segment Unable to Failover to Mirror**
			In prior 4.0.x 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	4.0.5.0	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.

Table A.1 Resolved Issues in 4.0.5.x

Issue Number	Resolved In	Category	Description
11696	4.0.5.0	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	4.0.5.0	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.

Known Issues in Greenplum Database 4.0.5.x

This section lists the known issues in Greenplum Database 4.0.5.x. A work-around is provided where applicable.

Table 2 Known Issues in 4.0.5.x

Issue	Description
12334	**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	**Error During CREATE or DROP of Pending Records in a Persistent Table Space** In rare circumstances, creating or dropping pending records in a persistent table space generates an error that can cause inconsistencies in the system catalog. **Work-around**: If possible, restore the catalog from back-up. Otherwise, contact EMC Technical Support.
12250	**Connections to Greenplum Database Remain Open After an ETL Operation** On rare occasions while performing an ETL operation, 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 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.

Table 2 Known Issues in 4.0.5.x

Issue	Description
12192	**Queries with Implicit Casts that Use Nested Loops Can Experience Sub-optimal Performance** Currently, queries with implicit casts that use nested loop in the query plan can experience reduced performance in Greenplum Database. This issue is related to the way in which Greenplum Database hashes different data types for columns that need to be joined.
12190	**Query Plan Is Not Created when Redistribution is on a Column that Does Not Support Hashing** Currently, the Greenplum Database query planner does not generate a query plan if the plan requires redistribution based on a column that contains a data type that does not support hashing. **Work-around**: Convert unsupported data types to data types supported by Greenplum Database.
12165, 12173	**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.
12099	**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.
12082	**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.
12050	**gp_toolkit Schema Has No Objects After Creating a New Database Instance** When a new database instance is created, the gp_toolkit schema is also created. Within this schema are automatically created objects. On rare occasions, creating a new database instance creates a gp_toolkit schema with no objects. **Work-around:** Execute the following command: `psql -f greenplum-db/share/postgresql/gp_toolkit.sql database_name` This populates the gp_toolkit schema with the required objects.
11690	**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.

Table 2 Known Issues in 4.0.5.x

Issue	Description
11668	**Append Only Table Performance Issues on Linux Platforms** XFS file system settings in the Red Hat Enterprise Linux 5.5 kernel can cause the fragmentation of the data files stored on disk for append-only (AO) tables. This fragmentation degrades query performance for queries that require AO table scan operations. **Work-around:** Update the Red Hat Enterprise Linux 5.5 kernel to version 2.6.18-194.26.1.el5 on all Greenplum hosts, and use the following recommended mount options for XFS: `rw,noatime,inode64,allocsize=16m`
10278	**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.
10028	**After Running gprecoverseg, Segments May Not Be Running in Their Preferred Role** In 4.0.5, 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. **Work-around:** Restart Greenplum Database (`gpstop -r`). A restart will bring up all segments in their preferred role. The Greenplum Database system will be unavailable for a brief period while it restarts.

9968	**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	**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.
6279	**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`.

Table 2 Known Issues in 4.0.5.x

Issue	Description
5647	**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	**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.

3125, 3213	**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	**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 4.0.5.0

For detailed upgrade procedures, see the following sections:

- Upgrading From 4.0.x.x to 4.0.5.0
- Upgrading From 3.3.x.x to 4.0.5.0
- Upgrading From 3.2.x.x to 4.0.5.0
- Upgrading from Releases Prior to 3.2.x

Upgrading From 4.0.x.x to 4.0.5.0

These instructions are for users currently running 4.0.x.x. An upgrade from 4.0.x.x to 4.0.5.0 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 4.0.x.x system (there can be no active connections to the database):

   ```
   $ gpstop
   ```

4. Run the installer for 4.0.5.0 on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example: `/usr/local/greenplum-db-4.0.5.0`

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-4.0.0.8/greenplum_path.sh
   ```

   to:

   ```
   source /usr/local/greenplum-db-4.0.5.0/greenplum_path.sh
   ```

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

   ```
   $ rm /usr/local/greenplum-db
   $ ln -s /usr/local/greenplum-db-4.0.5.0
   /usr/local/greenplum-db
   ```

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

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

7. Log in as root and install the 4.0.5.0 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-4.0.5.0
# gpscp -f seghostname_file gp.tar =:/
# gpssh -f seghostname_file
  => gtar --directory / -xvf /gp.tar
  => chown -R gpadmin /usr/local/greenplum-db-4.0.5.0
  => rm /gp.tar
  => rm /usr/local/greenplum-db
  => ln -s /usr/local/greenplum-db-4.0.5.0
  /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. If you are using the Greenplum Performance Monitor, install the new 4.0.5 Performance Monitor Web Application and update your environment to point to the new 4.0.5 Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.0.5 installation). Greenplum Performance Monitor 4.0.5 Web Application Packages are available at http://gpn.greenplum.com. See the *Greenplum Performance Monitor 4.0 Administrator Guide* for additional installation information.

11. (optional) EMC recommends that you update Greenplum Client, Loader, and Connectivity packages to version 4.0.5.0. These packages are available at http://gpn.greenplum.com.

Upgrading From 3.3.x.x to 4.0.5.0

You must be on release 3.3.x.x in order to use the 4.0 upgrade utility. The Greenplum upgrade utility (`gpmigrator`) makes the required changes to the system catalogs without requiring users to dump and restore their databases.

EMC strongly recommends that you perform a backup of your databases before running the upgrade utility. If you find issues when testing your upgraded system, you can restore this backup.

- Upgrade Checklist
- Upgrade Procedure
- Troubleshooting a Failed Upgrade

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 3.3.x.x to 4.0.5.0. Detailed upgrade instructions are also provided in the Upgrade Procedure section.

Pre-Upgrade Preparation (on your current system)
* 3.3.x.x system is up and available
☐ Log in to your master host as the `gpadmin` user (your Greenplum superuser).
☐ Install the Greenplum Database 4.0.5.0 binaries on all Greenplum hosts.
☐ Copy any custom modules from your current installation to your 4.0.5 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`.
☐ Drop the `gp_jetpack` schema from all databases (if it exists). Also note that `gp_` is now a reserved prefix for schema names in Greenplum Database 4.0 (as is `pg_`). If you have any user-created schemas named with a `gp_` prefix, Greenplum recommends that you rename them prior to upgrading.
☐ (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.
☐ Check for and recover any failed segments (`gpstate`, `gprecoverseg`).
☐ (Optional) Run the `gpcheckcat` utility to validate your system catalogs. If any errors are reported, contact Greenplum Customer Support.
☐ Run the `gpssh-exkeys` utility to exchange keys between all configured hostnames in your Greenplum Database system.
☐ 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). If you find issues when testing your upgraded system, you can restore using this backup.
☐	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 4.0.5 installation.
☐	Run `gpmigrator`
☐	After the upgrade process finishes successfully, your 4.0.5 system will be up and running.
Post-Upgrade (on your 4.0.5 system)	
* The 4.0.5 system is up	
☐	Reinitialize your standby master host (`gpinitstandby`).
☐	Upgrade `gpfdist` on all of your ETL hosts by installing the version 4.0.5 Load Tools package.
☐	Run the `gpbitmapreindex` utility to rebuild your bitmap indexes. The on-disk format of bitmap indexes has changed in 4.0.5.
☐	Install the new 4.0.5 Performance Monitor Web Application and update your environment to point to the 4.0.5 Performance Monitor binaries.
☐	Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.0.5 installation (if necessary).

Upgrade Procedure

This section divides the upgrade into three phases: pre-upgrade preparation, 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 3.3.x system)

Perform these steps on your current 3.3.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. Drop the `gp_jetpack` schema from all databases (if it exists). This Greenplum administrative schema has been renamed to `gp_toolkit` in 4.0. The `gp_toolkit` schema will be installed in all databases by default after upgrading to 4.0.5. For example, to drop `gp_jetpack`:

```
psql template1 -c 'DROP SCHEMA IF EXISTS gp_jetpack
CASCADE;'
psql mydatabase -c 'DROP SCHEMA IF EXISTS gp_jetpack
CASCADE;'
```

3. (*optional*) Vacuum all databases prior to upgrade. For example:

```
$ vacuumdb database_name
```

4. (*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'
```

5. Install the Greenplum Database 4.0.5.0 binaries on all Greenplum hosts. See the *Greenplum Database 4.0 Installation Guide* for detailed instructions.

6. If your current installation is utilizing any custom modules or PostgreSQL add-on modules, make sure the required library files are copied into the correct directory of your 4.0.5.0 installation on all Greenplum hosts. Shared library files for user-defined functions go in `$GPHOME/lib` and PostgreSQL add-on modules (such as `plr.so` or `pgcrypto.so`) go in `$GPHOME/lib/postgresql`. For example:

```
$ gpscp -f all_hosts_file \
  /usr/local/greenplum-db-3.3.7.2/lib/postgresql/plr.so \
  =:/usr/local/greenplum-db-4.0.5.0/lib/postgresql/plr.so
```

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

```
$ gpstate
```

8. If you have failed segments, you must recover them using `gprecoverseg` before you can upgrade.

```
$ gprecoverseg
```

9. Run the `gpssh-exkeys` utility referencing a host file (*all_hosts_file*) that has the host name for each master, standby master, and segment host. Include all configured interface names for each host. For example:

```
$ gpssh-exkeys -f all_hosts_file
```

10. (*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.

Upgrade Execution

During the migration process, all client connections to the master will be locked out. The migration utility locks out all client access to the master, however it does not block direct utility mode access to the segments. To ensure a safe upgrade, make sure users are aware that any connections to a Greenplum segment are not safe during the upgrade time frame.

11. (*optional but strongly recommended*) Back up all databases in your Greenplum Database system. If you find issues when testing your upgraded system, you can restore this backup.

You should also make a backup of your database roles and server configuration files. For example, using the 3.3.x version of `gpcrondump` (make sure you move the backup files out of your segment data directories):

```
$ /usr/local/greenplum-db-3.3.7.2/bin/gpcrondump -x database
-c -g -G -p -u /backup_target_directory
```

(**Solaris Users**) By default, the upgrade utility will pause on Solaris systems to allow for a ZFS snapshot to be taken. Make sure you take a snapshot of all segment file systems and the master file system. All Greenplum data directories must be on ZFS file systems. See the *Sun ZFS Administration Guide* for more information: http://opensolaris.org/os/community/zfs/docs/zfsadmin.pdf. If you already have a recent set of snapshots and do not want `gpmigrator` to pause mid-upgrade, run `gpmigrator` with the `--nosnapshot` option.

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 -f
```

14. Update your environment so that it is sourcing your new 4.0.5.0 installation.

a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.0.5.0 installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.0.5.0
/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-4.0.5.0
/usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```

    c. If your user profile file (such as `.bashrc`) does not use the `greenplum-db` symbolic link, you will need to update your profile file to source the new 4.0.5.0 installation. For example, update the following line in your profile file:

```
source /usr/local/greenplum-db-4.0.5.0/greenplum_path.sh
```

    And then source the updated file. For example:

```
source ~/.bashrc
```

15. As `gpadmin`, run the 4.0.5.0 version of `gpmigrator` specifying the following information:

- Your old and new `GPHOME` locations.

- A *new* port range for the 4.0 file *replication* processes. There are new processes in 4.0 used for primary to mirror communications. These must use a different port range from what you are already using for your primary and mirror segments. For example, if you are using 50000 and 60000 for your port ranges, use 53000 and 63000 for your replication port ranges.

- The mode by which to create the new 4.0 mirrors. The upgrade utility needs to drop and recreate all mirror segments.

- The path to your Greenplum Performance Monitor installation (if your system has the monitor program installed and enabled).

For example:

```
$ su - gpadmin

$ which gpmigrator
/usr/local/greenplum-db-4.0.5.0/bin/gpmigrator

$ gpmigrator --replication_port_base 53000 \
  --mirror_replication_port_base 63000 \
  --mirror_mode redundant \
  --gpperfmonhome /usr/local/greenplum-perfmon-2.1.0.0 \
  /usr/local/greenplum-db-3.3.7.2 \
  /usr/local/greenplum-db-4.0.5.0
```

16. (*Solaris only*) By default, the upgrade utility will pause on Solaris systems to allow for a ZFS snapshot to be taken. To resume the upgrade after the snapshots have been taken:

```
$ gpmigrator --resume
  /usr/local/greenplum-db-3.3.7.2 \
  /usr/local/greenplum-db-4.0.5.0
```

17. After the `gpmigrator` utility has completed successfully, your Greenplum Database 4.0 system will be up and running. The `gpmigrator` utility will take a while to complete (the time is proportional to the amount of data and the number of segments in your system).

Post-Upgrade (on your 4.0.5 system)

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

```
$ gpinitstandby -s standby_hostname
```

19. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the Greenplum Database 4.0.5 Load Tools package. Packages are available on http://gpn.greenplum.com.

20. The on-disk format of bitmap indexes has changed in 4.0.5. If your system has bitmap indexes, they must be reindexed (or dropped and recreated) after upgrading. Greenplum provides the `gpbitmapreindex` utility to facilitate the upgrade of any bitmap indexes in your system.

21. If you are using the Greenplum Performance Monitor, install the new 4.0.5 Performance Monitor Web Application and update your environment to point to the new 4.0.5 Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.0.5 installation). Greenplum Performance Monitor 4.0.5 Web Application Packages are available on http://gpn.greenplum.com. Note that the Performance Monitor data collection agents are integrated with the Greenplum Database server in 4.0.5, and are started automatically by the `postgres` server process. You no longer need a separate CRON job to monitor and restart the `gpmmon` data collection process.

22. Note that resource queues are enabled by default in 4.0.5. If you have any roles that are not assigned to a resource queue, they will be assigned to the `pg_default` resource queue after upgrade. For more information on configuring resource queues for workload management, see *Chapter 8* of the *Greenplum Database 4.0.5 Administrator Guide*.

23. Check your `postgresql.conf` server configuration files for any deprecated parameters you have set, and remove those settings. See "Deprecated Parameters" on page 33. Greenplum Database 4.0.5 will issue warning messages on startup if it finds any deprecated parameters set in your `postgresql.conf` files.

Troubleshooting a Failed Upgrade

If you experience issues during the migration process, contact Greenplum customer support at 1-866-410-6060 or open a support incident. Authorized Customer Administrators can log a support incident on the support portal. If you are a Greenplum Administrator at your company, and do not have access, please contact entitlement@greenplum.com.

Be prepared to provide the following information:

● A completed Upgrade Checklist.

● Log output from `gpmigrator` and `gpcheckcat` (located in `~/gpAdminLogs`)

Upgrading From 3.2.x.x to 4.0.5.0

Users on a release prior to 3.3.x.x cannot upgrade directly to 4.0.5. You must first upgrade from your current release to 3.3 (follow the upgrade instructions in the Greenplum Database 3.3 release notes available on http://gpn.greenplum.com). After you have upgraded to 3.3, follow the upgrade instructions for Upgrading From 3.3.x.x to 4.0.5.0.

Upgrading from Releases Prior to 3.2.x

If you are running a Greenplum Database release prior to 3.2.x.x, contact Greenplum customer support at 1-866-410-6060 or open a support incident. Authorized Customer Administrators can log a support incident on the support portal. If you are a Greenplum Administrator at your company, and do not have access, please contact entitlement@greenplum.com.

Greenplum Database 4.0 Client Tool Packages

Greenplum releases a number of client tool packages on various platforms that can be used to connect to Greenplum Database. The following table describes the client tool packages available, and their version compatibility with Greenplum Database server releases. Client tool packages are available for download on Greenplum Network (gpn.greenplum.com).

Table A.1 Greenplum Database Client and Server Version Compatibility

Client Package	Description of Contents	Client Versions	Server Versions
Greenplum Clients Greenplum Database Command-Line Interface (psql) Greenplum MapReduce (gpmapreduce)[1]		4.0.1	4.0.1.x 4.0.2.x
		4.0.3	4.0.3.x 4.0.4.x
		4.0.5	4.0.5.x
Greenplum Connectivity Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)		4.0.1	4.0.1.x 4.0.2.x
		4.0.3	4.0.3.x 4.0.4.x
		4.0.5	4.0.5.x
Greenplum Loaders Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)		4.0.1	4.0.1.x 4.0.2.x
		4.0.3	4.0.3.x 4.0.4.x
		4.0.5	4.0.5.x

1. gpmapreduce is not available on Windows.

Greenplum Database Documentation

The following Greenplum Database documentation is available in the `$GPHOME/docs` directory of your Greenplum installation, on Greenplum Network, or on Powerlink (powerlink.emc.com). Greenplum documentation is provided in PDF format.

Table 2 Greenplum Database Documentation

Title	Part Number	Revision
Greenplum Database 4.0 Administrator Guide	300-011-538	A03

Table 2   Greenplum Database Documentation

Title	Part Number	Revision
Greenplum Database 4.0 Installation Guide	300-011-541	A03
Greenplum Performance Monitor 4.0 Administrator Guide	300-011-542	A02

Be sure to periodically check Powerlink for documentation updates. Greenplum Database Release Notes can also be found on Powerlink.

Greenplum Database 4.0.4.x Release Notes

This section lists the customer reported issues that were resolved in Greenplum Database 4.0.4.x:

Table A.1   Resolved Issues in 4.0.4.x

Issue Number	Resolved In	Category	Description
11979	4.0.4.4	Fault Detection and Recovery	**Large Data Load Causes Segment Failures** In prior 4.0.x releases, the Greenplum Database fault detection process would sometimes mark segments as down during large data loading operations. This would typically occur when a segment was busy and unable to respond to the health polling process in the first try. This issue has been resolved in this release.
11818	4.0.4.3	Fault Detection and Recovery	**Segment Not Transitioning from Resynchronizing Mode, gpstate Status Not Available** When changes are being copied from a primary to a mirror segment, this is known as *resynchronizing* mode. In prior 4.0.x releases, a segment failure coinciding with multiple concurrent transactions could intermittently cause a mirror segment to remain in resynchronizing mode, never returning to a *synchronized* state. When this occurred, the `gpstate` utility reported the resynchronization progress as 'Not Available'. This issue has been resolved in this release.
11826	4.0.4.3	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.

Table A.1   Resolved Issues in 4.0.4.x

Issue Number	Resolved In	Category	Description
11814	4.0.4.3	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..."`

11673	4.0.4.2	Fault Detection and Recovery	**Failed Segment Unable to Failover to Mirror**
			
			In prior 4.0.x 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. Patch 4.0.4.2 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`

11667, 11686	4.0.4.1	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 is resolved in this release.
11595	4.0.4.0	Fault Detection and Recovery	**Failed Segments Not Marked as Down by Fault Detection Process** In prior 4.0.x releases, when a segment host went down due to file system problems (such as a corrupted/missing mount point or disk failure), the fault detection process of Greenplum Database did not mark all segments on that host down as expected. When a subsequent failure on another segment host occurred, Greenplum Database could potentially attempt to fail over to a mirror segment that was no longer available, leaving the system in an unresponsive state. This issue is now resolved in this release.

Table A.1   Resolved Issues in 4.0.4.x

Issue Number	Resolved In	Category	Description

11393	4.0.4.0	Data Loading	**COPY Errors when Loading Indexed Append-Only (AO) Tables**
			
			In prior 4.0.x releases, using concurrent COPY commands to load data into append-only tables with indexes could fail with errors such as:
			
			`Unexpected internal error: Master process x received signal SIGSEGV...`
			
			`...Error in copy_in_error_callback...`
			
			These errors could also cause primary segments to fail, requiring a segment recovery operation (gprecoverseg). This issue is now resolved in this release.

11354	4.0.4.0	Management Utilities	**Timeout Option Added to gpstop** When stopping Greenplum Database using gpstop, a segment may not complete shutdown in the allowed timeout of 60 seconds, especially if there are large transactions to roll back. To avoid shutdown errors, a new -t option was added to gpstop to allow the user to increase the allowed timeout. For example: `gpstop -t <timeout_in_seconds>`

11352	4.0.4.0	Fault Detection and Recovery	**"PANIC","XX000","Append-Only Mirror Resync EOFs intent count..." Error**
			
			In prior 4.0.x releases, if a system crash (such as a power failure) occurred while a primary and its mirror were in the process of resynchronizing append-only table data, it could cause both the primary and its mirror segment to become unavailable and unrecoverable through segment recovery utility (gprecoverseg). A symptom of this issue was Greenplum Database server and utility log messages such as:
			
			`PANIC","XX000","Append-Only Mirror Resync EOFs intent count...`
			`FATAL:  DTM initialization: failure during startup/recovery...`
			`... stderr:"failed: server did not respond with enough data"...`
			
			The condition that caused these errors has been resolved in this release.

11184, 11188	4.0.4.0	Standby Master	**"Master Mirroring Synchronization Lost" Error** In prior 4.0.x releases, the standby master host could possibly get into a state where it was unable to synchronize or catch up to the transaction state of the primary master. When this occurred, the user would see a "Master mirroring synchronization lost" error message, requiring a resynchronization of the standby master using the gpinitstandby utility. This condition that caused this error has been resolved in this release.

Greenplum Database 4.0.3.x Release Notes

This section lists the customer reported issues that were resolved in Greenplum Database 4.0.3.x:

Table A.2 Resolved Issues in 4.0.3.x

Issue Number	Category	Description
11406	Append-Only Tables	**Append-Only Tables: Concurrent Transaction Conflicts** In prior 4.0.x.x releases, multiple concurrent transactions that wrote to the same append-only (AO) table could potentially conflict with each other, especially on AO tables with indexes. When this occurred, the conflicting transaction would fail with an error such as: `ERROR "...tuple concurrently updated..."` This issue has been resolved in this release.

11399	Fault Detection	**Intermittent Segment Failover when Loading Append-Only Tables**
		
		In prior 4.0.x.x releases, loading an append-only table could intermittently trigger a segment failover. This would only happen in cases when the segment being updated was not running in its preferred role (for example, when the mirror segment was currently acting as the primary). When this type of segment failure occurred, administrators would see an error such as the following in the segment log file:
		
		`WARNING","58P02","mirror failure, could not link to 'pg_xlog...':`
		`File exists, failover requested...`
		This issue has been resolved in this release.

11378	Data Loading	**Greenplum Loaders Package Missing Files on AIX** The 4.0.1 release of the Greenplum Loaders client package was missing certain library files that prevented gpload from running. This issue has been resolved in this release.
10834	Management Utilities	**Shutdown in Progress Error After a gpstop** After stopping Greenplum Database using `gpstop`, you will not be able to restart Greenplum Database until shutdown completes on all segments. In some cases, if there are large transactions to roll back, a segment may not complete shutdown in the allowed timeout of 60 seconds. When this occurs, `gpstop` will now print an informational message if all segments do not complete shutdown in the allowed timeout. To increase the allowed `gpstop` timeout for future shutdown operations, use `gpstop -t <timeout_in_seconds>`.

11008	Data Loading	**Backslashes in gpload Configuration File Cause Error on Windows**
		
		In prior 4.0.x releases, backslash characters in the `gpload` configuration file (such as a backslash character in a Windows file path) would cause the following warning message, even though the use of the backslash character was legal in this context:
		
		`HINT:  Use the escape string syntax for backslashes, e.g., E'\\'`
		`WARNING:  nonstandard use of \\ in a string literal`
		This issue has been resolved in this release.

11181	Standby Master	**Filespace/Tablespace DDL Commands not Synchronized to Standby Master** In prior 4.0.x releases, `CREATE` and `DROP FILESPACE/TABLESPACE` commands did not always synchronize the change to the standby master. This issue has been resolved in this release.

Table A.2 Resolved Issues in 4.0.3.x

Issue Number	Category	Description
11198	PL/Java	**PL/Java Resources Cannot Be Retrieved from the CLASSPATH** In prior 4.0.x releases, resources (other than class files) could not be retrieved from the `CLASSPATH`. For example: `Example.class.getResource("`*`somefile`*`")` Would always return `null` for non-class files. This issue has been resolved in this release.
11216	System Expansion	**Incorrect gpexpand Warning: The current system appears to be non-standard...** In prior 4.0.x releases, using the `gpexpand` utility in interactive mode would always give the following warning, even when the system was using a standard configuration: `The current system appears to be non-standard...` This issue has been resolved in this release.
11257	Table Partitioning	**Splitting a Default Partition Does Not Preserve Table Distribution Policy** In prior 4.0.x releases, `ALTER TABLE...SPLIT DEFAULT PARTITION` did not propagate the table distribution key to the new partition that was created. This issue has been resolved in this release.
11371	gp_toolkit	**gp_toolkit Log Views: ERROR: data line too long** In prior 4.0.x releases, using the `gp_toolkit.gp_log_*` family of views would sometimes give the following error if the Greenplum Database log files contained large log messages: `ERROR: data line too long...` To reduce the likelihood of this error, the parameter `gp_max_csv_line_length` has an increased default of 1MB (1048576 bytes) and an increased allowed maximum of 4MB.
11389	Management Utilities	**Add --config Option to gpcheck** The `gpcheck` utility determines the platform on which you are running Greenplum Database and validates various platform-specific configuration settings. The checks are specified in the default file `$GPHOME/etc/gpcheck.cnf`. The `gpcheck` utility now accepts a `--config` option so you can override the default file with your own config file.

Greenplum Database 4.0.2.x Release Notes

This section lists the customer reported issues that were resolved in Greenplum Database 4.0.2. Greenplum Database 4.0.2 was a feature and service pack release intended for use on the EMC Greenplum Data Computing Appliance (DCA):

Table A.3 Resolved Issues in 4.0.2.x

Issue Number	Category	Description
10343	Performance Monitor	**Performance Monitor Log File Rotation** In prior releases of Greenplum Performance Monitor, log files were truncated when the configured size limit was reached. Logging has been changed so that log files are now rolled over. See the *Greenplum Performance Monitor Administrator Guide* for more information on log rotation and maintenance.
11079	Performance Monitor	**Performance Monitor Console Online Help** Prior releases of Greenplum Performance Monitor did not have online help available from within the Performance Monitor Console web application. Online help is now provided.
11103	DCA	**EMC Greenplum DCA - ConnectEMC Integration** For Greenplum Database instances running on the EMC Greenplum Data Computing Appliance (DCA), administrators now have the option of configuring ConnectEMC phone-home support. With ConnectEMC enabled, certain system events (such as a segment host failures) will be captured and automatically sent to EMC Customer Support.
10893	DCA	**EMC Greenplum DCA - Performance Monitor Health Monitoring Tab** For Greenplum Database instances running on the EMC Greenplum Data Computing Appliance (DCA), the Performance Monitor Console now has an additional Health Monitoring tab. This new tab shows the status of the various hardware components of the DCA.
11232	Management Utilities	**New gpcheck Utility to Verify System Hardware and OS Settings** This release introduces a new `gpcheck` utility that performs some pre-installation checks of the hardware and operating systems of hosts that comprise a Greenplum Database system. This utility replaces the `gpcheckos` utility.
10814	Management Utilities	**Reload all postgresql.conf Files with gpstop -u** In prior releases, a `gpstop -u` command would reload the `postgresql.conf` file of the master instance only. It now reloads all segment `postgresql.conf` files as well.

Greenplum Database 4.0.1.x Release Notes

Greenplum Database 4.0 is a major release which introduces a number of significant new features, performance and stability enhancements, and enhancements to the product architecture. Version 4.0.1 is the first general availability (GA) release of Greenplum Database 4.0, and contains all issue resolutions reported in the 4.0.0.0-4.0.0.8 controlled releases. Please refer to the following sections for more information about this release:

- New Features in Greenplum Database 4.0

- Changed Features in Greenplum Database 4.0
- Resolved Issues in Greenplum Database 4.0.1.x
- Known Issues in Greenplum Database 4.0.5.x
- Upgrading to Greenplum Database 4.0.5.0
- Greenplum Database Documentation
- Copyright Information

New Features in Greenplum Database 4.0

Greenplum Database 4.0 offers the following new features:

- Enhanced Workload Management with Dynamic Query Prioritization
- Self Healing Fault Tolerance Model with Differential Online Recovery
- Direct Dispatch Performance Optimization of Single Row Operation
- MPP Tablespace Support for Non-Uniform and SSD Segment Storage
- B-Tree and Bitmap Indexes on Column-Oriented and Append-Only Tables
- Health Monitoring Infrastructure with Email and SNMP Alerting
- Writable External Tables for Parallel Data Output
- Object-level 'Metadata Management' Tracking and Querying
- Enhanced Global Statistics Collection
- MapReduce Support for C Language Functions
- Support for User Defined Functions in PL/Java

Enhanced Workload Management with Dynamic Query Prioritization

Prior releases of Greenplum Database have included a range of workload management capabilities to allow database administrators (DBAs) to manage the resources allocated to query workloads. The primary mechanism has been role-based resource queues, which provide configurable query admission limits. By using resource queues to set limits on incoming queries, DBAs can control the number and complexity of active queries on the system at any given time, thereby protecting the system from over allocation of resources. Prior to 4.0, DBAs had to explicitly enable resource queues. In 4.0, resource queues are now always enabled.

> **Important:** Resource queues are required for all roles (users) in Greenplum Database 4.0.5. Any role not explicitly assigned to a resource queue will be assigned to the default resource queue, `pg_default`.

In addition to resource queues, Greenplum Database 4.0 adds a dynamic query prioritization infrastructure. Each query in the system has a priority value, which determines the relative share of system resources provided to it. The priority of a query is initially determined by the priority set on the resource queue through which it

enters. However, administrators also have the ability to adjust priority at runtime. This feature allows DBAs to control processing resources and ensure that important workloads can run with minimal interference from lower priority jobs.

Self Healing Fault Tolerance Model with Differential Online Recovery

In Greenplum Database 4.0, data redundancy (mirroring) is now performed using *physical block replication*. The primary and mirror segments are kept in sync at the physical disk block level, and changes to the primary are automatically applied to the mirror in a transactionally consistent manner. This new mirroring architecture offers a number of improvements over prior releases:

- **Automatic Failure Detection and Failover.** Should a segment server become unavailable, the system will automatically detect the failure and promote the necessary mirror segments to maintain full read/write operation. There is no longer a need to specify a fault action mode (*read-only* or *continue*).

- **Fast Differential Recovery.** Greenplum Database 4.0 keeps track of the changes that are made while a segment is down. When a failed segment becomes available again, only the modified disk blocks (as opposed to the entire contents) are copied over from the mirror. This ensures the fastest possible recovery time.

- **No Downtime for Segment Recovery.** Segment recovery takes place in the background while the system is fully online. The database is fully available and can support read/write operations while recovery is in progress.

- **Improved Write Performance for AO Tables.** Write transactions for compressed append-only tables are only processed once at the primary segments, and segment mirroring ensures that all modified disk blocks are synchronized to the mirrors.

In prior releases, Greenplum Database used *logical database replication* to maintain a mirror copy of a segment instance. This meant that a statement issued to Greenplum Database, such as an `INSERT`, was run on a primary segment first and then again on its corresponding mirror segment. While this was an effective technique for data redundancy, the new physical block replication infrastructure has a number of functional and performance advantages. This new infrastructure will also be the basis for future Greenplum Database high-availability and replication features.

Direct Dispatch Performance Optimization of Single Row Operation

Greenplum Database 4.0 introduces a performance enhancement to the query planning and dispatch process for small queries that only access data on a single segment (for example, a single-row `INSERT`, `UPDATE`, `DELETE` or `SELECT` statement). In queries such as these, the query plan is not dispatched to all segments, but is targeted to the segment that contains the affected row(s). This *direct dispatch* approach for this type of query dramatically reduces the response time and resource utilization of small queries.

MPP Tablespace Support for Non-Uniform and SSD Segment Storage

Greenplum Database 4.0 introduces support for tablespaces. Tablespaces allow database administrators to have multiple file systems per machine and decide how to best use their physical storage to store database objects. Tablespaces are useful for a

number of reasons, such as allowing different storage types for frequently versus infrequently used database objects, or controlling storage capacity and I/O performance on certain database objects. For example, highly utilized tables can be placed on file systems that use high performance solid-state drives (SSD), while the remaining tables utilize standard hard drives. This is an advanced feature for Greenplum system administrators who need greater control and flexibility over their database storage.

B-Tree and Bitmap Indexes on Column-Oriented and Append-Only Tables

In Greenplum Database 4.0, support for non-unique indexes has been added for append-only storage tables, including tables using compression and/or column-oriented storage. Indexes can greatly improve performance on compressed append-only tables for queries that return a targeted set of rows, as the optimizer now has the option to use an index access method rather than a full table scan when appropriate. For compressed data, an index access method means only the necessary rows are uncompressed.

Health Monitoring Infrastructure with Email and SNMP Alerting

Greenplum Database can now be configured to send email notifications to a system administrator whenever certain events occur, such as fatal server errors, segment failures, or system restarts.

Greenplum Database 4.0 also introduces support for SNMP. The Greenplum SNMP agent, `gpsnmpd`, can be configured to run on your Greenplum master host. This agent supports the standard relational database application management information base (`RDBMS-MIB.txt`) and can be polled by a network monitoring program, such as HP OpenView or Nagios. Greenplum Database can also be configured to send an SNMP notification to your network monitoring program when certain alert events occur (such as a segment failure). Greenplum Database supplies a custom management information base (`GPDB-MIB.txt`) to enable SNMP notifications for certain Greenplum Database events.

Writable External Tables for Parallel Data Output

Greenplum Database 4.0 now supports writable external tables, allowing users to perform high-speed parallel data output from a Greenplum Database instance to a file system, and ETL server, or other applications or databases. Writable external tables can be used in conjunction with Greenplum MapReduce to output job results to any external target. Writable external tables utilize the same *Scatter-Gather Streaming* infrastructure that is used when loading data.

Object-level 'Metadata Management' Tracking and Querying

Greenplum Database 4.0 now tracks metadata management information in its system catalogs about the objects stored in a database, such as tables, views, indexes and so on, as well as global objects such as roles and tablespaces. This allows administrators to examine information about an object, such as when it was created or what was the last operation performed. The system views *pg_stat_operations* and *pg_stat_partition_operations* can be used to look up actions performed on an object, such as a table. For example, you can use these views to see when a table was last vacuumed and analyzed.

Enhanced Global Statistics Collection

The `ANALYZE` command in Greenplum Database 4.0 now collects global database statistics from all active segments in the system, thereby providing the most accurate (and consistent) statistics for query planning and optimization. Previous versions of Greenplum Database would select a single segment to use as the basis for statistical data analysis, which did not always represent the true statistical variance of the data. Existing customers will be able to see the benefit of improved statistics collection the first time they run `ANALYZE` on a table after upgrading to 4.0. No other additional configuration is necessary.

MapReduce Support for C Language Functions

Greenplum MapReduce allows programmers who are familiar with the MapReduce programming paradigm to write map and reduce functions and submit them to the Greenplum Database parallel data flow engine for processing. Prior releases of Greenplum MapReduce provided language support for Perl and Python. In 4.0, developers can also use C functions. Both user-defined C functions and built-in database functions are supported.

Support for User Defined Functions in PL/Java

PL/Java is a PostgreSQL language extension that allows users to write custom database functions in the Java programming language. Implementation details can be found in `$GPHOME/share/postgresql/pljava/PLJAVA_README` of your 4.0 installation. Note that PL/Java support for Greenplum MapReduce is not included in this release.

Changed Features in Greenplum Database 4.0

This section describes the new and changed SQL commands, utilities and server configuration parameters for 4.0. Refer to the *Greenplum Database Administrator Guide* for detailed descriptions of items listed in this section.

- SQL Commands
- Client Utilities
- Management Utilities
- Server Configuration Parameters
- System Catalogs

SQL Commands

- Changed SQL Commands
- New SQL Commands

Changed SQL Commands

Table 2 Changed SQL Commands in 4.0

SQL Command	Description of Change
ALTER RESOURCE QUEUE	New syntax for setting resource queue attributes: WITH (*queue_attribute=value*) Deprecated (but still accepted) syntax: ACTIVE THRESHOLD, COST THRESHOLD OVERCOMMIT|NOOVERCOMMIT, IGNORE THRESHOLD
ALTER ROLE	New role attributes to grant permission to create an external table: CREATEEXTABLE | NOCREATEEXTABLE
ALTER TABLE	SET WITH(OIDS=TRUE) has been disabled for partitioned tables and append-only column-oriented tables.
COPY	New syntax to declare the newline format used in a load file: NEWLINE '[LF | CR | CRLF]'
CREATE DATABASE	SQL_ASCII disabled as a database ENCODING.
CREATE EXTERNAL TABLE	New syntax for WRITABLE external tables.
CREATE RESOURCE QUEUE	New resource queue attribute to set priority. New syntax for setting resource queue attributes: WITH (*queue_attribute=value*) Deprecated (but still accepted) syntax: ACTIVE THRESHOLD, COST THRESHOLD OVERCOMMIT|NOOVERCOMMIT, IGNORE THRESHOLD The CREATE RESOURCE QUEUE command cannot be run within a transaction in 4.0.
CREATE ROLE	New role attributes to grant permission to create an external table: CREATEEXTABLE | NOCREATEEXTABLE
CREATE TABLE	WITH OIDS (OIDS=TRUE) clause has been disabled for partitioned tables and append-only column-oriented tables.
CREATE INDEX	Added support for non-unique indexes on append-only tables. Dropped support for Hash indexes. Dropped support for GIN indexes. Dropped support for CONCURRENTLY keyword (building indexes concurrently).

New SQL Commands

Table 3 New SQL Commands in 4.0

SQL Command	Description of Change
DROP FILESPACE	New command for tablespace support.
CREATE FILESPACE	New command for tablespace support.
ALTER FILESPACE	New command for tablespace support.
CREATE TABLESPACE	Now supported in Greenplum Database. New syntax for declaring storage locations: FILESPACE *filespace_name*
ALTER EXTERNAL TABLE	New command for altering an external table definition (partial support).

Client Utilities

Greenplum Database has merged in client side changes from PostgreSQL 8.3, which has changed the following client utilities:

Table 4   PostgreSQL Client Utility Changes in 4.0

Utility Name	Description
`clusterdb`	**NEW OPTIONS:** `-v|--verbose, -w|--no-password`
`createdb`	**NEW OPTIONS:** `-w|--no-password` **DEPRECATED OPTIONS:** `-q|--quiet`
`createlang`	**NEW OPTIONS:** `-w|--no-password`
`createuser`	**NEW OPTIONS:** `-w|--no-password` **DEPRECATED OPTIONS:** `-q|--quiet`
`dropdb`	**NEW OPTIONS:** `-w|--no-password` **DEPRECATED OPTIONS:** `-q|--quiet`
`droplang`	**NEW OPTIONS:** `-w|--no-password`
`dropuser`	**NEW OPTIONS:** `-w|--no-password` **DEPRECATED OPTIONS:** `-q|--quiet`
`pg_dumpall`	**NEW OPTIONS:** `-f|--filespaces, -r|--resource-queues`
`reindexdb`	**NEW OPTIONS:** `-w|--no-password`
`vacuumdb`	**NEW OPTIONS:** `-w|--no-password, -F|--freeze`

Management Utilities

- Deprecated Utilities
- New Utilities
- Changed Utilities

Deprecated Utilities

Table 5   Deprecated Management Utilities in 4.0

Utility Name	Description
`gpchecknet`	The `gpchecknet` utility has been removed. The same functionality is provided in the `gpcheckperf` utility.
`gpcheckos`	The `gpcheckos` utility has been deprecated and replaced by the `gpcheck` utility.
`gpdemo`	The `gpdemo` program has been deprecated (located in `$GPHOME/demo/gpdemo.tar.gz` in prior releases). This program initialized a single-node Greenplum system, which is still possible using the sample single-node configuration file is supplied in: `$GPHOME/docs/cli_help/gp_init_singlenode_example`.
`gprebuildsystem`	Recommended process to rebuild a Greenplum system is to reinitialize the array using `gpinitsystem` and then restore your data from backup files.

Table 5 Deprecated Management Utilities in 4.0

Utility Name	Description
gpsizecalc	The gpsizecalc utility was used in prior releases to check the size of a database, schema, or table. Similar functionality is now supplied in the gp_toolkit administrative schema.
gpskew	The gpskew utility was used in prior releases to check the data distribution of a table. Similar functionality is now supplied in the gp_toolkit administrative schema.

New Utilities

Table 6 New Management Utilities in 4.0

Utility Name	Description
gpfilespace	New utility for tablespace support.
gpcheck	New utility for verifying system and OS settings.
gpconfig	New utility for editing postgresql.conf files.
gpperfmon_install	New utility to install the gpperfmon database and optionally enable the Performance Monitor data collection agents.
gpsnmpd	New utility for SNMP support.

Changed Utilities

The following utilities have deprecated, new, or changed options:

Table 7 Changed Management Utilities in 4.0

Utility Name	Description of Change
gp_dump	**CHANGED OPTIONS:** --gp-s=dbid (no more p(rimary), m(irror), i(ndividual) choices)
gpaddmirrors	**DEPRECATED OPTIONS:** -D (debug) **NEW OPTION:** -v (verbose) **CHANGED OPTIONS:** -p *port_offset* (calculation of mirror ports has changed), -i *mirror_config* (configuration file format has changed)
gpcheckperf	**NEW OPTIONS:** -r M (full matrix network test), --duration (duration for network test), --netperf (use netperf binary for network test instead of Greenplum network test) **CHANGED OPTIONS:** -r n|N (The network tests now use a Greenplum-provided network benchmark test. netperf is no longer distributed with Greenplum Database).
gpcrondump	**DEPRECATED OPTIONS:** -m (dump mirrors only), -p (dump primaries only)
gpdetective	**NEW OPTIONS:** --start_date, --end_date, --diagnostics, --logs, --cores, --pg_dumpall, --pg_dump_options, --tempdir, --connect
gpexpand	**CHANGED OPTIONS:** -i *expansion_config* (configuration file format has changed)
gpinitstandby	**DEPRECATED OPTIONS:** -i (do not start standby master synchronization process)

Table 7   Changed Management Utilities in 4.0

Utility Name	Description of Change
`gpinitsystem`	**NEW OPTIONS:** `--locale`, `--lc-collate`, `--lc-ctype`, `--lc-numeric`, `--lc-monetary`, `--lc-time`, `--lc-messages` **DEPRECATED OPTION:** `-r` (serial mode), `-i` (do not start standby master synchronization process)
`gpload`	**NEW OPTIONS:** `--gpfdist-timeout`
`gpmapreduce`	The YAML control file format for defining MapReduce jobs has been modified to allow support for built-in database functions and user-defined C functions.
`gprecoverseg`	**DEPRECATED OPTIONS:** `-S` *seg_dbid*, `-z` *seg_data_dir:seg_hostname*, `-f` (force restart), `-D` (debug) **NEW OPTION:** `-v` (verbose) **CHANGED OPTION:** `-F` (full recovery)
`gpstart`	**DEPRECATED OPTION:** `--recover` **NEW OPTION:** `-t` *timeout_in_seconds*
`gpstate`	**DEPRECATED OPTION:** `-t` (show default utility settings) **NEW OPTION:** `-e` (show segments with error conditions) **CHANGED OPTION:** `-v` (was 'version'; now 'verbose')
`gpstop`	**DEPRECATED OPTION:** `--recover` **NEW OPTION:** `-t` *timeout_in_seconds* **CHANGED OPTIONS:** `-u` (upload configuration files - now reloads `postgresql.conf` and `pg_hba.conf` files of master and all segments), `-r` (a restart now returns all segments to their preferred role)

Server Configuration Parameters

- Deprecated Parameters
- New Parameters

Deprecated Parameters

Table 8   Deprecated Server Configuration Parameters in 4.0

Parameter Names	Description
gp_external_grant_privileges	Marked for future deprecation. The `CREATE ROLE` and `ALTER ROLE` commands now have the ability to grant the permission to create an external table to an individual role. If this parameter is set to `on`, then pre-4.0 behavior will remain (any role can create an external table regardless of the individual permissions granted to the role).
gp_fault_action	Removed. With the new mirroring architecture, the system is always in *continue* (read/write) mode.
gp_hashagg_compress_spill_files	Replaced by `gp_workfile_compress_algorithm`
resource_scheduler	Removed. Resource queues are now always on.
stats_block_level stats_row_level	Combined and renamed to `track_counts`.
stats_command_string	Renamed to `track_activities`.

Table 8 Deprecated Server Configuration Parameters in 4.0

Parameter Names	Description
stats_reset_server_on_start	Removed. Similar functionality provided by the `pg_stat_reset()` function.
stats_start_collector	Removed. Statistics collection is always started in 4.0.

New Parameters

Table 9 New Server Configuration Parameters in 4.0

Parameter Names	Description
gp_contentid	New read-only parameter that shows the local segment content id.
gp_dbid	New read-only parameter that shows the local segment id.
gp_email_smtp_server gp_email_smtp_userid gp_email_smtp_password gp_email_from gp_email_to	New parameters to configure email alerts.
gp_enable_gpperfmon gpperfmon_port	Greenplum Performance Monitor parameters are now integrated into the default server `postgresql.conf` files.
gp_max_databases gp_max_filespaces gp_max_tablespaces	New parameters to control the number of databases, filespaces, and tablespaces allowed in a Greenplum Database system.
gp_num_contents_in_cluster	New read-only parameter that shows the number of primary segments in the Greenplum system.
gp_resqueue_priority gp_resqueue_priority_cpucores_per_segment gp_resqueue_priority_sweeper_interval	New parameters to enable and configure query prioritization.
gp_snmp_monitor_address gp_snmp_community gp_snmp_use_inform_or_trap	New parameters to enable SNMP notifications.
gp_vmem_idle_resource_timeout	New parameter to manage system resource utilization for idle database sessions.
gp_workfile_compress_algorithm	Replaces the `gp_hashagg_compress_spill_files` parameter.
max_work_mem	New parameter to control memory usage at the query processing level.
pljava_classpath pljava_statement_cache_size pljava_release_lingering_savepoints pljava_vmoptions	New parameters for PL/Java implementation.
ssl_ciphers	New parameter to specify the list of SSL ciphers that are allowed to be used on secure connections.
track_activities	Replaces `stats_command_string` parameter.
track_counts	Replaces `stats_block_level` and `stats_row_level` parameters.

Parameters with Changed Defaults

Table 10 Parameters with Changed Default Values in 4.0

Parameter	Old Value	New Value
join_collapse_limit	8	16
from_collapse_limit	8	16
gp_max_csv_line_length	65536	1048576

System Catalogs

Deprecated System Catalogs

Table A.1 Deprecated System Catalogs in 4.0

Catalog Name	Description
gp_configuration	Replaced by *gp_segment_configuration*
gp_jetpack (schema)	Replaced by the *gp_toolkit* schema, which is now installed in all databases by default.

New System Catalogs

Table 2 New System Catalogs in 4.0

Catalog Name	Description
gp_fastsequence gp_global_sequence gp_persistent_database_node gp_persistent_filespace_node gp_persistent_relation_node gp_persistent_tablespace_node gp_relation_node	New catalogs to support the new physical file block replication mirroring architecture.
gp_segment_configuration	New Greenplum system configuration table. Replaces *gp_configuration*.
pg_appendonly_alter_column	New catalog to support adding columns to append-only tables.
pg_filespace pg_filespace_entry	New catalogs for tablespace support.
pg_resourcetype pg_resqueuecapability	New catalogs to support query prioritization and future resource queue enhancements.
pg_stat_last_operation pg_stat_last_shoperation	New catalogs to support meta-data tracking.

New System Views

Table 3 New System Views in 4.0

View Names	Description
pg_stat_operations pg_stat_partition_operations	New views for 'last operation' tracking.
pg_resqueue_attributes	New view for examining the attributes of a resource queue.

New System Schemas

Table 4 New System Schemas in 4.0

Schema Name	Description
gp_toolkit	Replaces gp_jetpack (the optional 3.3.x Greenplum Administrative Schema). gp_toolkit is automatically installed in all databases in 4.0. Also gp_ is now a reserved prefix for schema and tablespace names.

Built-in Database Functions

Table 5 New Built-in Functions in 4.0

Function	Description
gp_adjust_priority(*session_id*, *statement_count*, *priority*)	New function to adjust the priority of a running query. This function replaces some customer-specific (and unsupported) utilities that are no longer available in Greenplum Database.
gp_elog('*message_text*', *boolean*)	New function that can be used to trigger an email or SNMP alert in Greenplum Database.

Resolved Issues in Greenplum Database 4.0.1.x

This section lists the 3.3.x customer reported issues that were resolved in Greenplum Database 4.0.1.x. For customers who participated in the 4.0 controlled release beta program, release 4.0.1 contains all of the controlled release fixes reported in 4.0.0.0-4.0.0.8, plus the additional fixes described in this section:

Table A.1 Resolved Issues in 4.0.1.x

Issue Number	Category	Description
10343	Performance Monitor	**Performance Monitor Log File Rotation** In prior releases of Greenplum Performance Monitor, log files were truncated when the configured size limit was reached. Logging has been changed so that log files are now rolled over. See the *Greenplum Performance Monitor Administrator Guide* for more information on log rotation and maintenance.
11079	Performance Monitor	**Performance Monitor Console Online Help** Prior releases of Greenplum Performance Monitor did not have online help available from within the Performance Monitor Console web application. Online help is now provided.
11103	DCA	**EMC Greenplum DCA - ConnectEMC Integration** For Greenplum Database instances running on the EMC Greenplum Data Computing Appliance (DCA), administrators now have the option of configuring ConnectEMC phone-home support. With ConnectEMC enabled, certain system events (such as a segment host failures) will be captured and automatically sent to EMC Customer Support.
10893	DCA	**EMC Greenplum DCA - Performance Monitor Health Monitoring Tab** For Greenplum Database instances running on the EMC Greenplum Data Computing Appliance (DCA), the Performance Monitor Console now has an additional Health Monitoring tab. This new tab shows the status of the various hardware components of the DCA.
11232	Management Utilities	**New gpcheck Utility to Verify System Hardware and OS Settings** This release introduces a new `gpcheck` utility that performs some pre-installation checks of the hardware and operating systems of hosts that comprise a Greenplum Database system. This utility replaces the `gpcheckos` utility.
10814	Management Utilities	**Reload all postgresql.conf Files with gpstop -u** In prior releases, a `gpstop -u` command would reload the `postgresql.conf` file of the master instance only. It now reloads all segment `postgresql.conf` files as well.
2976	Functions (Built-In)	**Cannot Insert pg_relation_size Results into a Table** The use of the `pg_relation_size` function has been disabled for `CREATE TABLE AS SELECT` and `INSERT INTO SELECT` queries. For example: `CREATE TABLE mytbl_size AS SELECT pg_relation_size(mytable) FROM pg_tables WHERE tablename = 'mytable';` In prior releases, the `pg_relation_size` function would not return results when used in these types of queries.
4835	Functions (User-Defined)	**Use of SECURITY DEFINER in User-Defined Functions** In prior releases, Greenplum Database did not properly handle execution of user-defined functions created with `SECURITY DEFINER` declared. `SECURITY DEFINER` specifies that the function is to be executed with the privileges of the user that created it, including superuser privileges. This issue has been resolved in this release.
4974	Performance Monitor	**Performance Monitor - ERROR: missing data for column "t0_val"** In prior releases of Greenplum Performance Monitor, the data collection agents would fail with the following error when loading certain query history data into the *iterators_history* table of the *gpperfmon* database: `ERROR: missing data for column "t0_val"` This issue has been resolved in this release.

Table A.1 Resolved Issues in 4.0.1.x

Issue Number	Category	Description
6379	Table Partitioning	**Added Partitions Do Not Inherit Primary Keys and Indexes of Parent Table** If a user creates a new partitioned table with a primary key or an index, the child table partitions will inherit the primary keys and indexes of the parent table. However, in prior releases, adding a new partition using `ALTER TABLE...ADD PARTITION` did not inherit primary keys or indexes as expected. This issue has been resolved in this release.
6400	Client Authentication	**Format of pg_hba.conf File Changed for LDAP and PAM Authentication** This release of Greenplum Database uses the PostgreSQL 8.4 client authentication libraries (not the 8.2 version used in Greenplum releases prior to 3.3.x). This means that the format of the `pg_hba.conf` client authentication file requires different syntax when declaring LDAP or PAM authentication methods. If the old format is used, Greenplum Database will not start. See the *Greenplum Database Administrator Guide* for more information on the `pg_hba.conf` file format.
6567	Data Loading	**Unencrypted Password Authentication Support Added to gpload** In prior releases, `gpload` did not support unencrypted `password` authentication. If the user running `gpload` had `password` authentication configured in `pg_hba.conf` configuration file of Greenplum Database, their load operations would fail. Password authentication for `gpload` is now supported in this release.
6940	Memory Control	**Improved Query-Level Memory Control** One challenge Greenplum system administrators face is preventing user workloads from causing out-of-memory errors on the segment hosts, while still allowing users to tune memory usage for a particular query. To address this challenge, the server configuration parameter `max_work_mem` has been added to control memory usage at the query processing level. As a complimentary parameter to `work_mem`, it sets the maximum amount of memory that can be used by a single segment worker process during query processing.
6957	Data Loading	**Function Expression Support Added to gpload** In prior releases, `gpload` did not support the use of function expressions in the `gpload` control file (for example, using a function to encrypt a column in the target table). This is now supported in this release.
6958	Concurrency	**Resource Utilization of Idle Database Sessions** In prior releases, idle sessions would maintain a hold on their allocated resources (such as shared memory). While this is often desirable behavior, it can result in a lower number of concurrent sessions allowed to the system. In 4.0.x, if a database session is idle for longer than 180 seconds, the session will release its resources but remain connected. This idle timeout can be configured using the server configuration parameter: `gp_vmem_idle_resource_timeout`.
8340	Standby Master	**Changes Made by gpactivatestandby Not Transactional** In prior releases, if the `gpactivatestandby` utility exited or was cancelled mid-operation, it did not properly roll back any changes it made to the Greenplum Database catalogs. When this occurred, any subsequent `gpactivatestandby` or `gpstop/start` operations would fail when trying to connect to the current master instance. This issue has been resolved in this release.

Table A.1 Resolved Issues in 4.0.1.x

Issue Number	Category	Description
8422	Standby Master	**Do Not Allow Activation of an Out-of-Date Standby Master** In prior releases, the gpactivatestandby utility did not safe-guard against activating an out-of-date standby master. Activating a standby master that is not in sync with the transactional state of the segments may introduce catalog and data inconsistencies that can render your Greenplum Database instance unusable. In this release, gpactivatestandby will not promote the standby master to active until first checking its synchronization status. If problems are detected, the utility will give the user an error.
8458	Table Partitioning	**Failed ALTER TABLE...SPLIT PARTITION on an Open-Ended Range Partition** In 3.3.x releases, users could not split a range partition if the partition was created with an open-ended range (did not have both a START and an END criteria defined). Running an ALTER TABLE...SPLIT PARTITION command on such a partition would result in the following error: ERROR: Unexpected internal error... This issue has been resolved in this release.
8471	Data Loading	**SQL_ASCII Disabled as a Database Encoding** In prior releases, the SQL_ASCII character set was allowed as a server-side encoding. This can potentially cause problems if users load any non-ASCII data into the database, as this encoding does not validate non-ASCII characters, and may allow invalid encoding sequences to be loaded. To prevent problems when converting between server and client-side encodings, Greenplum has disabled the use of SQL_ASCII as a database encoding.
8559	Backup and Restore	**gpcrondump Fails when Using Multiple Tables with the -T Option** The -T option for gpcrondump allows you to specify one or more table names to exclude from the database dump. Using -T with a comma-separated list of table names caused gpcrondump to fail. In this release, you can specify the -T option multiple times (for example, -T table1_name -T table2_name -T table3_name).
8623	Data Loading	**Add Role Permissions for CREATE EXTERNAL TABLE** In 3.3.x releases, only superusers had the ability to create an external table by default. The gp_external_grant_privileges server configuration parameter was added to allow all other roles the ability to create external tables, however this ability could not be granted on a per-role basis. In 4.0, the CREATE ROLE and ALTER ROLE commands now include the CREATEEXTABLE/NOCREATEEXTABLE attributes which can be granted to individual roles. To utilize this new feature, the gp_external_grant_privileges parameter must be off in the master postgresql.conf file. If set to on, then pre-4.0 behavior will remain (any role can create an external table regardless of the individual permissions granted to the role).
8658	Data Loading	**gpfdist Does Not Read Appended Data in gzip-Compressed Load Files** In prior releases, gpfdist did not read data that was appended to an existing gzip compressed file. This would result in appended data not being loaded. This issue has been resolved in this release.
8854	DDL and Utility Statements	**ALTER TABLE - ERROR: ZLIB Compress2 Failed (Detail: 'Insufficient Memory'...)** In 3.3.x releases, adding or altering a column in a compressed append-only table would potentially fail due to insufficient memory. This issue has been resolved in this release.

Table A.1   Resolved Issues in 4.0.1.x

Issue Number	Category	Description
9429	System Initialization	**Improved Error Message for gpinitsystem** In prior releases, `gpinitsystem` would fail with a confusing error message if it could not start a `postgres` database listener process. For example: `The program "postgres" is needed by initdb but was not found in the same directory as "$GPHOME/bin/initdb"` The error message has been changed to provide more information to help users troubleshoot the problem.
9823	Table Partitioning	**ALTER TABLE...ADD/DROP COLUMN on Partitioned Table not Propagated to Child Tables** In 3.3.0 and later releases, when adding or dropping columns on a partitioned parent table, the columns did not fully propagate to the child table partitions. This could cause some column data to not appear in the child tables when data was loaded through the parent table. This issue has been resolved in this release.
10361	Query Execution	**ERROR: unsupported call to mark position of Motion operator** In 3.3.x releases, certain queries involving a merge-join operation in their query plan could fail to execute with the following error: `ERROR: unsupported call to mark position of Motion operator...` This issue has been resolved in this release.
10532	User Defined Functions	**PL/Java Support for User Defined Functions** The PostgreSQL PL/Java language extension is now supported in Greenplum Database, which allows users to write custom database functions in the Java programming language. Implementation details can be found in `$GPHOME/share/postgresql/pljava/PLJAVA_README` of your 4.0 installation.
10597	Data Loading	**New gpload Option to Set Allowed Timeout** In prior releases, `gpload` operations could fail if they did not establish a connection to the `gpfdist` file distribution process before the timeout threshold. On systems with significant network traffic, the default timeout threshold of 10 seconds could be insufficient. `gpload` now has a `--gpfdist-timeout` option to allow users to increase the timeout threshold.
10603	Query Execution	**Query Planner Performance on Queries Involving Partitioned Tables** In prior releases, certain queries involving large partitioned tables could take longer than expected in the query planning phase, especially when most of the partitions did not contain any data. This issue has been resolved in this release.

10753	Table Partitioning	**ALTER TABLE...SPLIT PARTITION Improvements**
		
		In prior 3.3.x releases, the `ALTER TABLE...SPLIT PARTITION` command had a number of issues. First, users would encounter the following error when reusing an existing partition name in an `ALTER TABLE...SPLIT PARTITION` command:
		
		`ERROR:  invalid use of boundary specification for DEFAULT partition "<name>"`
		
		This restriction has been removed in this release. The existing partition name can now be reused in a `SPLIT PARTITION` command.
		
		Second, a split on a partitioned table that had a primary key constraint would sometimes fail with the following error:
		
		`ERROR:  new partition overlaps existing partition...`
		
		This issue has been resolved in this release.

Table A.1   Resolved Issues in 4.0.1.x

Issue Number	Category	Description
10911	Data Loading	**ERROR missing chunk number 0 for toast value...** In prior 3.3.x releases, users sometimes encountered the following error when accessing certain large rows from data load error tables: `ERROR missing chunk number 0 for toast value...` This issue has been resolved in this release.
10927	Query Execution	**Certain Date Values Not Recognized in WHERE Clause** In prior 3.3.x releases, when a `WHERE` clause contained an array of date values, certain dates with years prior to1910 and later than 2089 would not always be recognized by the query planner. This issue has been resolved in this release.
10419	Performance Monitor	**Performance Monitor: 'ERROR: data line too long...'** When the Greenplum Performance Monitor agents attempted to save query information to the Performance Monitor database, certain long queries would fail to load with the following error: `ERROR:  data line too long. likely due to invalid csv data` This issue has been resolved in this release.
8948	DDL and Utility Statements	**reindexdb: 'ERROR: Invalid distributed snapshot received...'** In prior 3.3.x releases, users occasionally encountered the following error when running the `reindexdb` client utility (rebuilding all indexes in a database) when the database contained TOAST tables (oversized attribute storage tables): `ERROR: Invalid distributed snapshot received...` This issue has been resolved in this release.
11074	Client Tools - gpmapreduce	**gpmapreduce: C Function Fatal Error** In prior 4.0.x controlled releases, running a Greenplum MapReduce program written in C could result in the following fatal error if the C function encountered complier warnings: `Program terminated with signal 11, Segmentation fault.` This issue has been resolved in this release.

Copyright Information