Greenplum
August 23, 2010

Welcome to Greenplum Database 3.3.7.0

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 3.3.7 is the seventh service pack release for version 3.3. This service pack release includes the contents of 3.3.0.0 - 3.3.6.7 plus additional new issue resolutions. Please refer to the following sections for more information about this release:

- Resolved Issues in Greenplum Database 3.3.7.x
- Known Issues in Greenplum Database 3.3.7.x
- Upgrading to Greenplum Database 3.3.7.x
- Installing Greenplum Database 3.3.7 (New Users)
- Greenplum Database 3.3 Documentation

For the contents of previous 3.3.x releases, please see the release notes section of each corresponding release:

- Greenplum Database 3.3.6.x Release Notes
- Greenplum Database 3.3.5.x Release Notes
- Greenplum Database 3.3.4.x Release Notes
- Greenplum Database 3.3.3.x Release Notes
- Greenplum Database 3.3.2.x Release Notes
- Greenplum Database 3.3.1.x Release Notes
- Greenplum Database 3.3.0.x Release Notes

Resolved Issues in Greenplum Database 3.3.7.x

This section lists the customer reported issues that are now resolved in Greenplum Database 3.3.7.x:

Table 1 Resolved Issues in 3.3.7.x

Issue	Category	Description
8561	Performance Monitor	**Improved Stability of Performance Monitor Agents** In prior releases of Greenplum Performance Monitor, the monitor agent on the master (`gpmmon`) would occasionally fail due to out-of-memory errors. This release introduces a number of stability and performance enhancements for Performance Monitor. Memory usage of `gpmmon` has been reduced by up to 80 percent.
8902	Performance Monitor	**Performance Monitor: Active Queries Show Submit Time of Jan-1-1970** In prior releases of Greenplum Performance Monitor, occasionally an active query would be logged with a submit time of 10:00 AM Jan-1-1970 and a blank user and database name. This issue has been resolved in this release.

8906
10451
	Performance Monitor	**Performance Monitor Error: Data Missing from History Tables**
		In prior releases of Greenplum Performance Monitor, not all history data for completed queries would load successfully into the *gpperfmon* database. Users may have seen errors such as the following in their log files:
		`ERROR: missing data for column...`
		`ERROR: extra data after last expected column...`
		This issue has been resolved in this release.

9045	Management Utilities	**psql Does Not Recognize Unicode Byte Order Marks** Unicode (or UTF-8) formatted files often have a byte-order mark (BOM) at the beginning of the file. This special character allows programs to automatically recognize the file as Unicode. In prior releases, the `psql` command-line client utility did not recognize the BOM character in files passed to `psql`, and would fail with the following error: `ERROR: syntax error...` `psql` now recognizes the BOM character in this release.
9514	Transaction Management	**Transaction Errors - Unable to Connect With psql** In prior releases, running several million small, concurrent transactions could cause issues in Greenplum's global transaction manager. When these issues occurred, users may have noticed failed transactions in their log files or were unable to initiate new sessions with `psql`. This issue has been resolved in this release.
9762	Management Utilities	**Segment Recovery (gprecoverseg) Not Reporting Status When Run In Background** In prior releases, running the `gprecoverseg` utility with `nohup` or in the background would cause the output to be buffered and reported in stages. When recovering a large amount of data, the time lapse between status updates could appear to users like the utility was not working. Output from `gprecoverseg` is no longer buffered in this release.
10321	Table Partitioning	**Error When Splitting a Partition Date Range Partition** In prior 3.3.x releases, users sometimes encountered the following error when trying to split a date range partitioned table: `ERROR: new partition overlaps existing partition "<name>"` This issue has been resolved in this release.

Table 1 Resolved Issues in 3.3.7.x

Issue	Category	Description
10362	Query Execution	**ERROR: unsupported call to mark position of Motion operator...** In prior 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.
10599	Data Loading	**New gpload Option to Set Allowed Timeout** In prior releases, `gpload` operations could fail if they could not establish a connection to the `gpfdist` file distribution process within 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.

Known Issues in Greenplum Database 3.3.7.x

This section lists the customer reported issues that remain unresolved in Greenplum Database 3.3.7.0. Work arounds are provided where applicable.

Table 2 Known Issues in 3.3.7.x

Issue	Description
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.

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`

4427	**Cannot Downgrade with gpmigrator** In this release, users cannot use the `gpmigrator` upgrade utility to downgrade (return an upgraded system to a Greenplum version prior to 3.2.1). Users wishing to downgrade must either use ZFS snapshots (on Solaris only) or rebuild and restore their database from backup files.
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.

Table 2 Known Issues in 3.3.7.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.
6126	**Jetpack Function gp_skew_details Requires Superuser Privileges** Though the documentation commenting in *jetpack.sql* does not specify it, the function `gp_skew_details` can be run only by Greenplum Database superusers. Non-superusers who try to run this function will see an error similar to the following: `ERROR: permission denied for schema pg_aoseg`
6407	**Upgrade Utility Requires Disabling of Pluggable Authentication Modules (PAM)** The 3.3 upgrade utility, `gpmigrator`, does not support pluggable authentication modules. Work-around: if you specify any PAM modules in `pg_hba.conf`, temporarily disable those entries during the upgrade process. You may safely enable them after upgrading.
6379	**Adding or dropping a primary key constraint on a partitioned table does not cascade** Commands to add or drop the primary key constraint on a partitioned parent table do not cascade to child table partitions as expected. Workaround: explicitly drop or add primary key constraints for child tables.
7024	**Child tables do not inherit storage attributes of parent table** When a child table is created in a partitioned table, the child table does not inherit the storage attributes (append-only, column orientation) of the parent table as expected. Workaround: explicitly declare the storage options for child tables using the WITH clause.
9573	**gpload Performance Degradation** The `gpload` utility creates and drops external tables as part of its normal load processing. After running many load operations without vacuuming the system catalog tables in between, the startup performance of `gpload` can degrade due to bloat in the system catalogs. Workaround: Run `VACUUM` on the system catalog tables after load operations using gpload.
10224	**System Catalog Indexes not Synchronized on the Standby Master** In certain situations, updates to the system catalog indexes do not get synchronized to the standby master. When this occurs, users may see the following error after activating their standby master: `FATAL: index "<name>" contains unexpected zero page at block 0` Workaround: Run `REINDEX` on the named index to update it.

Table 2 Known Issues in 3.3.7.x

Issue	Description
10421	**ALTER TABLE...SPLIT PARTITION Error When Using Existing Partition Name** When you split a partitioned table (other than the default partition) using `ALTER TABLE...SPLIT PARTITION`, you must specify two new table names for the split. You cannot reuse the current partition name for one of the split tables. Reusing the current partition name during a split will result in the following error: `ERROR: invalid use of boundary specification for DEFAULT partition "<name>"`
10741	**Performance Monitor queries_now Entries Not Cleared on Error** If a query fails before it has completed processing, its status is not always updated correctly in the *queries_now* data files of Performance Monitor. These queries can appear as active queries in Performance Monitor even though they are no longer running. Workaround: Find all files in `$MASTER_DATA_DIRECTORY/gpperfmon/data` which match the pattern `q*.txt` and are older than 2 days. Update these files to change the last byte in the file from `1` or `2` to `4` (1 = submitted, 2 = running, 4 = error).

Upgrading to Greenplum Database 3.3.7.x

Version 3.3.7.0 provides an upgrade utility to facilitate upgrades from version 3.2.x.x. Reference documentation for the `gpmigrator` utility is provided in the *Greenplum Database Administrator Guide*, and detailed step-by-step procedures for usage are provided in this section.

To upgrade from 3.1.1.5 - 3.1.3.x versions, you must first upgrade to version 3.2 using the instructions in the Greenplum Database 3.2 release notes. This documentation is available for download from http://gpn.greenplum.com.

If you are running a Greenplum Database release prior to 3.1.1.5, contact Greenplum customer support for instructions.

For detailed upgrade procedures, see the following sections:

- Upgrading From 3.3.x.x to 3.3.7.0
- Upgrading from 3.2.x.x to 3.3.7.0
- Upgrading From 3.1.1.5-3.1.3.x to 3.3.7.0
- Upgrading from Releases Prior to 3.1.1.5

Upgrading From 3.3.x.x to 3.3.7.0

These instructions are for users currently running on 3.3. An upgrade from 3.3.x.x to 3.3.7.0 involves updating the Greenplum Database software binaries only.

1. Stop your current Greenplum Database 3.3.x.x system:

   ```
   gpstop
   ```

2. Run the installer for 3.3.7.0 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-3.3.7.0`

3. 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-3.3.5.0/greenplum_path.sh
```

to:

```
source /usr/local/greenplum-db-3.3.7.0/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-3.3.7.0
/usr/local/greenplum-db
```

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

```
source ~/.bashrc
```

5. Log in as root user and install the 3.3.7.0 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:

```
gtar -cvf gp.tar /usr/local/greenplum-db-3.3.7.0
gpscp -f seghostname_file gp.tar =:/
gpssh -f seghostname_file
   => gtar --directory / -xvf /gp.tar
   => chown -R gpadmin /usr/local/greenplum-db-3.3.7.0
   => rm /gp.tar
   => rm /usr/local/greenplum-db
   => ln -s /usr/local/greenplum-db-3.3.7.0
   /usr/local/greenplum-db
```

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

```
gpstart
```

Upgrading from 3.2.x.x to 3.3.7.0

You must be on release 3.2.x.x in order to use the 3.3.7.0 upgrade utility. The Greenplum upgrade utility (`gpmigrator`) makes the required changes to the system catalogs without requiring users to dump and restore their databases.

Greenplum 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
  - Pre-Upgrade Preparation
  - Upgrade Execution
  - Post-Upgrade
  - Troubleshooting a Failed Upgrade

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 3.2.x.x to 3.3.7.0. Detailed upgrade instructions are also provided in the Upgrade Procedure section.

Pre-Upgrade Preparation (on your current system)	
* 3.2.x.x system is up and available	
☐	Log in to your master host as the `gpadmin` user (your Greenplum superuser).
☐	Install the Greenplum Database 3.3.7.0 binaries on all Greenplum hosts.
☐	Copy any custom modules from your current installation to your 3.3.7 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`.
☐	Check for and recover any failed segments (`gpstate`, `gprecoverseg`).
☐	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 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 3.3.7.0 installation.
☐	Run `gpmigrator`
☐	After the upgrade process finishes successfully, your 3.3.7.0 system will be up and running.
Post-Upgrade (on your 3.3.7 system)	
* The 3.3.7 system is up	
☐	Reinitialize your standby master host (`gpinitstandby`).
☐	Upgrade `gpfdist` on all of your ETL hosts by installing the version 3.3 Load Tools package.

☐	See "Server Configuration Parameter Changes" on page 31 for parameter changes made in 3.3. Edit your `postgresql.conf` files as needed to adjust the appropriate parameters.
☐	Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 3.3.7.0 installation (if necessary).

Upgrade Procedure

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

Note: Carefully evaluate each section, determine which conditional steps apply to your system (for example, "If you have failed segments") and perform all required 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

Perform these steps on your current 3.2.x.x Greenplum Database system. This procedure is performed from your Greenplum master host and should be executed by the Greenplum superuser (`gpadmin`).

1. Install the Greenplum Database 3.3.7.0 binaries on all Greenplum hosts. See the following sections of *Chapter 6* of the *Greenplum Database 3.3.7 Administrator Guide* for detailed instructions:
   - "Running the Installer"
   - "Copying the Greenplum Software to the Segment Hosts"
2. 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 3.3.7 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-3.2.1.5/lib/postgresql/plr.so \
     =:/usr/local/greenplum-db-3.3.7.0/lib/postgresql/plr.so
   ```
3. Run `gpstate` to check for failed segments.

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

   ```
   $ gprecoverseg
   ```

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.

5. (*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 configuration files. For example, using `gpcrondump`:

   ```
   $ /usr/local/greenplum-db-3.2.1.5/bin/gpcrondump -x database
   -c -g -G -p -u /backup_target_directory
   ```

   (**Sun DW Appliance users**) If using ZFS snapshots as your backup method, you must shutdown Greenplum Database before taking your snapshots. Make sure you take a recent 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

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

   ```
   $ gpinitstandby -r
   ```

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

   ```
   $ gpstop -C -f
   ```

8. Update your environment so that it is sourcing your new 3.3.7 installation.

   a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 3.3.7 installation directory:

      ```
      # rm /usr/local/greenplum-db
      # ln -s /usr/local/greenplum-db-3.3.7.0
      /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:

      ```
      # gpssh -f segment_hosts_file
      => rm /usr/local/greenplum-db
      => ln -s /usr/local/greenplum-db-3.3.7.0
      /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 3.3.7.0 installation. For example, update the following line in your profile file:

      ```
      source /usr/local/greenplum-db-3.3.7.0/greenplum_path.sh
      ```

And then source the updated file. For example:

```
source ~/.bashrc
```

9. Run `gpmigrator` specifying your old and new `GPHOME` locations. For example:

```
$ /usr/local/greenplum-db-3.3.7.0/bin/gpmigrator \
/usr/local/greenplum-db-3.2.1.5 \
/usr/local/greenplum-db-3.3.7.0
```

10. After the `gpmigrator` utility has completed successfully, your Greenplum Database 3.3.7 system will be up and running.

Post-Upgrade

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

```
$ gpinitstandby -s standby_hostname
```

12. (optional) See "Server Configuration Parameter Changes" on page 31 for a summary parameters that have changed from 3.2 to 3.3. Edit your `postgresql.conf` files as needed to adjust the appropriate parameters.

13. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL server and reinstall `gpfdist` using the Greenplum Database 3.3 Load Tools package. Packages are available on http://gpn.greenplum.com.

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 Greenplum Subject Matter Experts and Central Point of Contact Administrators can log a support incident on the support portal. If you are a Greenplum Subject Matter Expert or Central Point of Contact 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 3.1.1.5-3.1.3.x to 3.3.7.0

Users on a release prior to 3.2.x.x cannot upgrade directly to 3.3.7.0. You must first upgrade from your current release to 3.2 (follow the upgrade instructions in the Greenplum Database 3.2 release notes available on http://gpn.greenplum.com). After you have upgraded to 3.2, follow the upgrade instructions for Upgrading from 3.2.x.x to 3.3.7.0.

Upgrading from Releases Prior to 3.1.1.5

If you are running a Greenplum Database release prior to 3.1.1.5, contact Greenplum customer support at 1-866-410-6060 or open a support incident. Authorized Customer Greenplum Subject Matter Experts and Central Point of Contact Administrators can log a support incident on the support portal. If you are a Greenplum Subject Matter Expert or Central Point of Contact at your company, and do not have access, please contact entitlement@greenplum.com.

Installing Greenplum Database 3.3.7 (New Users)

These are the high-level steps to install and initialize a new Greenplum Database system. For detailed instructions, please see the installation chapter of the *Greenplum Database 3.3 Administrator Guide*.

1. Run the installer on the Greenplum Database master host.
2. As root, set the OS tuning parameters for your platform on all Greenplum hosts.
3. Allocate a `gpadmin` user to own and run your installation. This user must exist on all Greenplum hosts.
4. Source the `greenplum_path.sh` file in your `gpadmin` user profile (`.bashrc`). This sets the environment variables needed by Greenplum Database.
5. Create your data directory locations on all Greenplum hosts.
6. Use the `gpssh-exkeys` utility to exchange SSH keys between all hosts in your Greenplum array. Note that for a single host demo configuration you still must exchange ssh keys between the current host and itself.
7. (multi-host configuration only) Use the `gpscp` and `gpssh` utilities to copy and install the Greenplum Database software on all segment hosts.
8. Use the `gpinitsystem` utility to initialize and start your Greenplum Database system. This utility requires a configuration file. For example:

   ```
   gpinitsystem -c gp_init_config
   ```

   A sample `gp_init_config` configuration file can be found in `$GPHOME/docs/cli_help/gp_init_config_example`. Edit this file to reflect your desired Greenplum Database array configuration.

Greenplum Database 3.3 Documentation

The following Greenplum Database documentation is available in the `$GPHOME/docs` directory of your Greenplum installation, or you can go to http://gpn.greenplum.com to download the latest documentation:

GPAdminGuide.pdf - *Greenplum Database 3.3 Administrator Guide*

Greenplum Database 3.3.6.x Release Notes

This section lists the customer reported issues that were resolved in Greenplum Database 3.3.6.x:

Table 3 Resolved Issues in 3.3.6.x

Fixed In	Issue	Category	Description

3.3.6.7	10595	Data Loading	**Load Times Out on Busy Network**
			In prior releases, customers loading data via Greenplum Loader (`gpload`) or the Greenplum Connector for Informatica PWX would sometimes see load job failures when the network was busy. When this occurred, log files might show errors such as:
			`ERROR | http response code 400 (time out) from gpfdist`
			`[ERROR ] The Integration Service failed to write to a pipe`
			To address this problem, the default `gpfdist` timeout value has been changed to `0` (no timeout).

3.3.6.7	10600	Data Loading	**Changed Default Timeout for Parallel File Server (gpfidist)** To address load timeouts in Greenplum Loader (`gpload`) and the Greenplum Connector for Informatica PWX, the default timeout for `gpfdist` has changed to `0` (no timeout). This timeout can be changed by using the `-t` option when starting `gpfdist`.
3.3.6.6	8200	DDL	**ERROR: Too Many Updates/Deletes Within a Transaction** In prior releases, certain combinations of data definition language (DDL) commands within the same transaction could potentially cause the following error: `Too Many Updates/Deletes Within a Transaction; Reader gang not able to provide correct visibility 0/0` This issue has been resolved in this release.
3.3.6.6	10104	Query Execution	**Query of Column-Oriented Table Fails when SELECT List Contains CASE Statement** In prior releases, certain queries of an append-only, column-oriented (CO) table would fail when the `SELECT` list contained a `CASE` statement with an `ELSE` clause that included a non-constant value. This issue has been resolved in this release.
3.3.6.6	10372	Management Utilities	**psql ODBC Driver Not Included in Windows Connectivity Package** In prior 3.3.x releases, the psql ODBC driver (08.04.0200) was not included in the Windows 32-bit version of the Greenplum Database connectivity package. This driver is now included in the Windows 32-bit connectivity package.
3.3.6.4	8407	Data Loading, Table Partitioning	**Failed INSERT on Table with 10,000 or More Partitions** In prior 3.3.x releases, using `INSERT` to load data into a parent table with 10,000 or more child table partitions would sometimes result in the following error: `ERROR: Error on receive from seg# ... lost synchronization with server: got message type "o"...` This issue has been resolved in this release.

Table 3 Resolved Issues in 3.3.6.x

Fixed In	Issue	Category	Description
3.3.6.4	9082	Backup and Restore	**gpsnap.py Utility Removed from Greenplum Database 3.3.7 Distribution** `gpsnap.py` was an undocumented utility introduced in 3.2.x for Greenplum Database systems running on Solaris. It was a prototype utility that suspended database activity and took a ZFS snapshot of the segment data directories. It has been removed from the release because it was not a complete solution, and was never intended for production distribution. This utility will be replaced by the production utility, `gpsuspend`, in a future 4.x release.
3.3.6.4	9228	Data Loading	**Failed gpload Processes not Releasing Memory** In 3.3.5.x and later, in some situations where `gpload` failed or was cancelled mid-operation, the failed `gpload` processes would not release memory. This issue has been resolved in this release.
3.3.6.4	9573	Data Loading	**Degrading gpload Performance with Trickle Loads** In 3.3.5.x and later, the performance of `gpload` would degrade after doing a large number (thousands) of small trickle loads. Before doing a load operation, `gpload` would query the system catalog to perform some pre-load verification. This pre-load query response time would continually increase after many small load operations. This performance issue has been addressed in this release.
3.3.6.4	9938	DDL	**Failed CREATE INDEX Operation on a Partitioned Table** In 3.3.5.x and later, running a `CREATE INDEX` operation on a partitioned table with a large number (hundreds or thousands) of child table partitions would intermittently fail with the following error: `ERROR: relation "public." does not exist` This issue has been resolved in this release.
3.3.6.4	9957	Query Execution	**Failed Queries When Using CASE Expressions in the SELECT List** In 3.3.5.x and later, certain queries that used `CASE` expressions in the `SELECT` list would fail with the following error: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.
3.3.6.3	9793	Data Loading	**COPY Memory Allocation Issue** In prior releases, certain usages of the COPY command cause Greenplum Database to encounter insufficient memory errors. COPY was not properly freeing all allocated memory when several COPY load operations were run concurrently. This issue has been resolved in this release.
3.3.6.2	8657	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.
3.3.6.2	8840	Bitmap Index	**Bitmap Index Not Updated After VACUUM FULL** In prior releases, a bitmap index would not reflect updates that occurred after a VACUUM FULL operation was run on the table containing the bitmap index. A REINDEX was required to update the bitmap index. This issue has been resolved in this patch release.
3.3.6.2	8855	DDL	**Insufficient Memory Error when Altering an Append-Only Table** In prior releases, certain ALTER TABLE operations of append-only tables would use more resources than expected and occasionally cause 'Insufficient memory' errors on high concurrency systems. This issue has been resolved in this release.

Table 3 Resolved Issues in 3.3.6.x

Fixed In	Issue	Category	Description
3.3.6.1	8645	Cursors	**Cursor Visibility Within Subtransactions** Patch 3.3.6.1 was a follow-up fix to issue 8259. In certain cases involving the use of cursors within a subtransaction, a cursor that was in the process of scanning would not pick up `UPDATE` or `DELETE` operations that occurred within its parent transaction. This issue has been resolved in this patch release.
3.3.6.0	3492	Workload Management	**Resource Queue Threshold does not Reset when Queue is Empty** In prior releases, cost-based resource queues did not fully release resources after a query was terminated. This issue has been resolved in this release.
3.3.6.0	5657	Management Utilities	**gpstop Reports that Database is Stopped Before Shutdown is Complete** In releases 3.2.x and earlier, `gpstop` would display a message that the database had successfully shut down when shutdown of the backend processes was still in progress. If users tried to restart Greenplum Database before shutdown was fully complete, they would see the following error: `FATAL: the database system is shutting down...` `gpstop` now waits until all shutdown processes are complete before displaying the success message.
3.3.6.0	6298	Backup and Restore	**Partitioned Table Schema is Modified During Backup** In prior releases, the `gp_dump` utility in some cases created DDL statements for partitioned tables with conflicting range boundaries, resulting in errors when the partitioned table was restored. This issue has been resolved in this release.
3.3.6.0	7377	Bitmap Index	**ERROR: oid ## already in use...** In releases 3.2.x and earlier, running multiple concurrent `VACUUM` and/or `TRUNCATE` statements on a table with a bitmap index would sometimes cause the statement to fail with the following error: `ERROR: oid ## already in use...` This issue has been resolved in this release.
3.3.6.0	7529	Workload Management	**Syntax Error when Setting IGNORE THRESHOLD on a Resource Queue** In prior releases, users would receive a syntax error when trying to reset the `IGNORE THRESHOLD` attribute of a resource queue to no limit (`-1.0`). This issue has been resolved in this release - setting `IGNORE THRESHOLD` to `-1.0` is now accepted.
3.3.6.0	7631	Query Processing	**ERROR: Invalid Sequence Access...** In certain queries involving functions, the use of the `nextval` function would return the following error: `ERROR: Invalid Sequence Access. Slice table did not specify a seqserver` This issue has been resolved in this release.
3.3.6.0	7633	Backup and Restore	**Child Table Partitions not Dumped with gp_dump/gpcrondump -t** In prior releases, when using the `-t` option with `gp_dump` or `gpcrondump`, only the specified table was dumped. In the case when the table was a top level partitioned table, the dump utilities were ignoring the child table partitions. This issue has been resolved in this release - the dump utilities now dump the data from child tables as well.

Table 3 Resolved Issues in 3.3.6.x

Fixed In	Issue	Category	Description
3.3.6.0	7652	Query Execution	**Query Fails on a Column-Oriented Table with COUNT(*) and INNER JOIN** In prior releases, certain queries of a column-oriented table containing `COUNT(*)` and `INNER JOIN` clauses would fail with the following error: `ERROR: Unexpected internal error: Segment process received signal SIGSEGV` This issue has been resolved in this release.
3.3.6.0	7725	Management Utilities	**Allow Non-Standard Hostnames in gpexpand Utility** In prior releases, the `gpexpand` utility would not allow you to add a host to your array if the hostname did not conform to the current segment host naming convention. This restriction has been removed in this release.
3.3.6.0	7735	Query Execution	**Query with GROUPING SETS Clause Fails** In prior releases, certain queries containing a `GROUPING SETS` clause would fail with the following error: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.
3.3.6.0	7741	Table Partitioning	**Cannot SPLIT a List Partition With a Multi-Column Partition Key** In prior releases, users could not split a list partition if the partition was created using a multi-column partition key. This issue has been resolved in this release.
3.3.6.0	7779	Query Execution	**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.
3.3.6.0	7822	Fault Tolerance	**Fault Prober Process Failure** In prior releases, when the master host was under heavy load and all memory was being utilized, the Greenplum fault detection prober process (`ftsprobe`) could fail and cause a temporary system outage. This issue has been resolved in this release.
3.3.6.0	7823	Data Loading	**Load Fails if First Line is Bigger Than 64K** In prior releases, a load would fail if the first line of data was larger than 64K even if `gp_max_csv_line_length` was set to 1MB. This issue has been resolved in this release.
3.3.6.0	7897	Table Partitions	**Splitting a Default Partition with NULL Partition Key Values** In prior releases, if a partitioned table contained NULL values in the partition key column, those rows would be moved to the new partition boundary when a split of the default partition was performed. This issue has been resolved in this release. NULL values now remain in the default partition after a split.
3.3.6.0	7997	Query Execution	**Query of Partitioned Table Fails with ROLLUP or CUBE Grouping Functions** In prior releases, queries of partitioned tables containing ROLLUP or CUBE functions in the GROUP BY clause would fail with the following error: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.

Table 3 Resolved Issues in 3.3.6.x

Fixed In	Issue	Category	Description
3.3.6.0	8112	Query Execution	**Resource Utilization on Queries that Scan Large Numbers of Partitions** In prior releases, a query plan that involved a scan of a deeply partitioned table (hundreds of child partitions) would use more resources than needed. The resource utilization for this type of query has been improved in this release.
3.3.6.0	8130	Management Utilities	**gpexpand Unicode Decode Error** In prior releases, the `gpexpand` utility would fail with the following error if table column names contained non-ASCII characters: `Unicode decode error ASCII code cannot decode byte...` This issue has been resolved in this release.
3.3.6.0	8174	DDL	**Unique Index Creation on Data with Duplicate Keys** In prior releases, the creation of a primary key or unique index would in rare cases succeed when the data contained duplicates. This issue has been resolved in this release.
3.3.6.0	8259	Transaction Management	**ERROR: qExec Reader: writer gang raced ahead** In certain cases involving the use of cursors within a subtransaction, the subtransaction would receive the following error and roll back: `ERROR: qExec Reader: writer gang raced ahead...` This issue has been resolved in this release.
3.3.6.0	8284	Performance Monitor	**Performance Monitor Displays Out of Date Resource Queue Information** In prior releases, the performance monitor did not correctly account for cancelled queries when displaying resource queue status information in the Performance Monitor user interface. This issue has been resolved in this release.
3.3.6.0	8318	Backup and Restore	**Cannot Use pg_dump to Dump a Single Child Table Partition** In prior releases, the following command was allowed but would not dump any data if *`table_name`* was a child table partition: `pg_dump -t table_name`. In 3.3.6, an error message is now given if users try to dump a single child table partition with *`pg_dump`*. Users who want to dump a single partition should use `COPY` instead.
3.3.6.0	8428	DDL	**CREATE TABLE...WITH OIDS Disabled on Column-Oriented Tables** Creating tables that use per-row system generated object ids (OIDs) is now disabled for column-oriented tables.
3.3.6.0	8454	Workload Management	**Do Not Allow Negative Resource Queue Cost Values** As a result of the fix for 3492, in certain cases the resource queue cost limit would reset to a negative number instead of zero. This issue has been resolved in this release.

Greenplum Database 3.3.5.x Release Notes

Changed Features in Greenplum Database 3.3.5.0

Changed Default for gp_statistics_use_fkeys

When the parameter `gp_statistics_use_fkeys` is enabled, the query optimizer uses foreign key information stored in the system catalog to optimize joins between foreign keys and primary keys. In previous releases this parameter defaulted to "off." After intensive testing, Greenplum has changed that default value to "on" so that newly installed systems can benefit from the optimization without the need to edit `postgresql.conf`.

Resolved Issues in Greenplum Database 3.3.5.0

This section lists the customer reported issues that were resolved in Greenplum Database 3.3.5.x:

Table 4 Resolved Issues in 3.3.5.0

Issue	Category	Description
7153	Backup and Restore	**Failed gpcrondump operation causes Greenplum Database to hang** Under certain conditions, a unsuccessful `gpcrondump` operation could cause some of the segments to hang and output error messages in a continuous loop. This issues has been resolved in this release.
7199, 7520	Loading	**gpload on 32-bit Linux Platforms Fails with "File too large" Error** Loading files larger than 2GB on 32-bit Linux systems could fail with error messages containing the warning, "`File too large`." This issue has been resolved in this release.
7241	Documentation	**Documentation refers to a file not included in the installation** The Administrator Guide for release 3.3.4 referred to a Jetpack README file that was not included in the Greenplum Database installation. In this release, `README_JETPACK` is located as expected in `$GPHOME/lib/jetpack`.
7286	Query Processing	**Errors with count(distinct) queries using sort** Some `count(distinct)` queries that ran successfully in in earlier versions failed in version 3.3.4 with SIGSEGV and other errors. A viable workaround could be achieved by setting the parameter `gp_enable_mk_sort` to "off" (default in 3.3.4 is "on"). This issue has been resoved in this release, and these queries run successfully with `gp_enable_mk_sort` set to "on".
7327	Transaction Management	**SIGSEGV errors encountered when creating a table** Infrequent issues with memory allocation could cause a `CREATE TABLE` command to fail with SIGSEGV errors. Improvements in memory allocation in this release have resolved this issue.
7359	Loading	**Slower performance when loading data to a column-oriented partitioned table** In version 3.3.4, loading data to a column-oriented partitioned table could proceed slower than loading to row-oriented equivalents. This issue has been resolved in this release.
7381	Query Processing	**Out of memory errors for queries involving TOAST columns** In some queries involving large source tables with TOAST columns, a memory leak could develop and cause out of memory errors. Low settings for `gp_vmem_protect_limit` could make this issue more likely to occur. This issue has been resolved in this release.

Table 4 Resolved Issues in 3.3.5.0

Issue	Category	Description
7399	Loading	**gpload fails with password authentication** In previous versions, the `gpload` command-line client could incorrectly interpret carriage return characters as part of the password for authentication. This issue has been resolved in this release.
7403	Loading	**Load failures with Informatica Pre-SQL functions returning void** If a function entered in Informatica as “Pre-SQL” returned a value of void, the loading operation could fail. This issue has been resolved in this release.
7448	Transaction Management	**Queries with user-defined queries in a cursor fail with shared snapshot errors** Certain queries that use user-defined functions, which execute normally in a non-cursor context, result in “shared snapshot” errors when used from a cursor. This issue has been resolved in this release.
7477	Transaction Management	**Unable to drop bitmap index** An issue in earlier versions prevented the dropping of bitmap indexes under certain conditions. This issue has been resolved in this release.
7493	Loading	**Data loaded with incorrect client encoding yields incorrect results** Under certain conditions, improperly encoded data is loaded with uneven distribution among segments, resulting in failed queries with joins. The conditions are: • An error table is used for loading • The client encoding differs from the server encoding • The input row is invalid in the client encoding, but actually is valid in the server encoding In this release, the incorrectly encoded data is rejected at load time as expected.

Greenplum Database 3.3.4.x Release Notes

This section lists the customer reported issues that were resolved in Greenplum Database 3.3.4.x:

Table 5 Resolved Issues in 3.3.4.0

Issue	Category	Description
3492	workload management	**Resource Queue Threshold does not Reset When Queue is Empty** In prior releases, cost-based resource queues did not fully release resources after a query was terminated. This issue has been resolved in this release.
6278	client tools	**Tab Auto-Completion in PSQL Client Program Exits Session** In prior releases, tab auto-completion in the PSQL command-line client program would sometimes cause the session to exit. This issue has been resolved in this release.
6483	query planner	**Improve Query Planner Row Width Estimates for Partitioned Tables** In prior releases, the query planner was miscalculating the row width of partitioned tables, which resulted in a less than optimal plan. This release introduces a performance enhancement to the query planner’s row width estimate algorithm for partitioned tables, resulting in a 30-50% reduction in query execution time for certain queries involving partitioned tables.

Table 5 Resolved Issues in 3.3.4.0

Issue	Category	Description
6524	loading	**Increase Timeout for gpload to gpfdist Connections** In prior releases, the gpfdist parallel file distribution program had a timeout of 2 seconds. This timeout has been increased to 30 seconds to account for systems running high numbers of concurrent workloads.
6533	query execution	**ERROR: qExec Reader: writer gang raced ahead...** In prior 3.3.x.x releases, users would sometimes encounter the following error when running transactions on systems with high numbers of concurrent workloads: `ERROR: qExec Reader: writer gang raced ahead, invalid snapshot` This issue has been resolved in this release.
6565	query execution	**Query Failure for COUNT(DISTINCT x) Involving Large Aggregations** In prior 3.3.x.x releases, certain queries involving a `COUNT(DISTINCT x)` operation would intermittently fail with the following error when the number of rows to be aggregated was several million or more: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.
6694	vacuum	**VACUUM FULL Operation Fails on Table with a Bitmap Index** In prior 3.3.x.x releases, a `VACUUM FULL` operation on a table that had a bitmap index would intermittently fail with the following error: `ERROR: oid` *xxx* `already in use...` This issue has been resolved in this release.
6699	loading	**External Tables Not Recognizing Single-Quoted Values in CSV Files** In prior 3.3.x.x releases, single-quoted values in a comma-separated values (CSV) file were not recognized as a valid formatting option by Greenplum external tables. This issue has been resolved in this release.
6740	loading	**COPY TO Operation Involving a Sort is Slow** In prior releases, performing a `COPY TO` operation to unload data from Greenplum was significantly slower when the command contained an `ORDER BY` clause (sort) as compared to `COPY TO` without a sort. This release introduces a performance enhancement that makes this operation approximately three times faster than prior releases.
6758	loading	**gpload Fails on UPDATE Operation** In 3.3.2 and later, the temporary tables created by `gpload` to process an `UPDATE` operation did not always use the same distribution policy as the target table, thereby causing some `UPDATE` operations to fail. This issue has been resolved in this release.
6785	query execution	**Query Executor Process Failures** In 3.3.2 and later, certain queries with HashAgg operations would fail during query processing and cause the segments to reset. The error encountered as a result was: `FATAL: the database system is starting up...` This issue has been resolved in this release.

6866	query execution	**ERROR: table row type and query-specified row type do not match...**
		In prior releases, users would intermittently encounter the following error on tables with recently dropped columns:
		`ERROR: table row type and query-specified row type do not match...`
		`DETAIL: Query provides a value for a dropped column at ordinal position...`
		This issue has been resolved in this release.

Table 5 Resolved Issues in 3.3.4.0

Issue	Category	Description
6894	query execution	**Query Failure when Joining on "char" Datatype** In 3.2.x.x and later, certain queries involving a join on a `"char"` datatype (not `char` or `char(n)`) would fail with the following error: `FATAL: Unexpected internal error: Master process received signal SIGBUS` Note that this only occurred for the `"char"` datatype, not `char` or `char(n)`. This issue has been resolved in this release.
6899	query planner	**Inconsistent Results for HashAgg Operation on Two Node Systems** In prior releases, the query planner would produce inconsistent results for plans involving Hash Aggregations that did not fit into memory and spilled to disk. This problem was only evident on small Greenplum systems (2 segment hosts). This issue has been resolved in this release.
6908	query execution	**CREATE INDEX Out of Memory Error on Tables with INET Distribution Key** In prior releases, certain operations such as CREATE INDEX would encounter out of memory errors when run on very large fact tables that were distributed on an `inet` data type column. This issue has been resolved in this release.
6948	query execution	**Cannot Cancel or Kill Process or Shutdown Database** In prior 3.3 releases, certain large queries that encountered certain errors during processing would reach a state where they could not be cancelled. In some cases, this also prevented a shutdown of Greenplum Database using `gpstop`. This issue has been resolved in this release.
7007	backup/restore	**ERROR: invalid byte sequence for encoding...** In prior releases, loading data via `COPY` or external tables would (in certain rare cases) accept invalidly encoded data values into the database. This error would only occur with multi-byte database encodings when the load data included backslash escape sequences. Some users encountered this problem as a result of trying to restore an invalid string from a dump file. When this occurred, users would see the following error: `ERROR: invalid byte sequence for encoding...` This issue has been resolved in this release.

7011	management utilities	**gpskew and gpsizecalc 'Table Does Not Exist' Error on Solaris Platform**
		In prior releases, the `gpskew` and `gpsizecalc` utilities would return the following error when run against Greenplum databases using a multi-byte locale. This error was only encountered on the Solaris platform:
		`Bad string`
		`[FATAL]:-Table <name> does not exist in database <name>. Script Exiting!`
		This issue has been resolved in this release.

7042	table partitioning	**DELETE FROM ... USING Statement Scans All Partitions** In prior 3.3.x.x releases, a `DELETE FROM ... USING` statement scanned all partitions of the `USING` table even when a `WHERE` predicate was applied to filter out unselected partitions. This issue has been resolved in this release.
7053	system expansion	**Primary Keys not Enforced While System Expansion is in Progress** While a system expansion is in progress, tables are marked with a random distribution policy. During this time, primary key constraints cannot be enforced until the expansion utility (`gpexpand`) successfully completes the redistribution of the table. In this release, a warning has been added to `gpexpand` to inform administrators, and `gpexpand` will now redistribute tables with primary keys first to limit the possibility of duplicate keys being introduced.

Table 5 Resolved Issues in 3.3.4.0

Issue	Category	Description
7078	query execution	**Out of Memory Error on DELETE Query Involving Joins of Partitioned Tables** In prior releases, certain large DELETE operations that involved joining multiple partitioned tables would reach the memory protection limit and fail with the following error: `ERROR:  Out of memory...` This issue has been resolved in this release.
7089	backup/restore	**gp_dump Error: 'relation does not exist'** In prior releases, users would get the following error when using `gp_dump` to backup tables whose names began with **`gp_`** (the naming convention used for Greenplum catalog tables): `ERROR: relation "table_name" does not exist` This issue has been resolved in this release. Users can now backup tables whose names begin with **`gp_`**.
7120	query planner	**Query Failure When Accessing Oversized Attributes in pg_catalog Tables** In prior releases, certain complex queries of the system catalog tables would fail with the following error when the oversized attribute storage technique (TOAST) was involved: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.
7132	query planner	**Change Logging Level for 'Extreme skew in the innerside of Hashjoin' Message** In prior releases, certain query plans involving a hash join operator with skewed input would output a warning: `WARNING: Extreme skew in the innerside of Hashjoin...` This event has been changed from a WARNING logging level to a LOG logging level so it no longer displays to the client. This event does not prevent queries from executing successfully.
7144	upgrade	**Catalog Tables Larger Than 1GB Cause Upgrade to Fail** Using prior versions of the 3.3.x upgrade utility (`gpmigrator`), some large Greenplum systems were not able to upgrade if they had catalog tables larger than 1GB in size. The error encountered as a result of this issue was: `[FATAL] ... catalog is missing 4 attribute(s) for relid...` This issue has been resolved in this release.
7172	query monitor	**Monitor does not Display CPU Usage and Query Text for Prepared Statements** In prior releases, the performance monitor user interface did not display CPU usage and query text metrics for prepared statements. This issue has been resolved in this release.

Greenplum Database 3.3.3.x Release Notes

This section lists the issues that were resolved in Greenplum Database 3.3.3.x.

Table 6 Resolved Issues in 3.3.3.x

Issue	Fixed In	Description
6637	3.3.3.0	**ALTER TABLE ... RENAME Does Not Cascade to DEFAULT Partitions** In prior 3.3.x.x releases, changing the name of a partitioned parent table did not cascade to DEFAULT child table partitions as expected. This issue has been resolved in this release.
6691	3.3.3.0	**ALTER TABLE ... ALTER COLUMN Prevented on Tables with DEFAULT Partitions** In prior 3.3.x.x releases, you could not alter the column type of a partitioned parent table if there was a DEFAULT partition in the partition hierarchy. This restriction has been removed in this release.
6745	3.3.3.0	**Users Allowed to Create a Table in Read-Only Mode** In prior 3.3.x releases, certain create table operations that should have been blocked were permitted when running in read-only fault operation mode. This issue has been resolved in this release. Note that read-only mode, by design, does not restrict the creation of temporary tables.

6774	3.3.3.0	**Query Failure Causes Segment to Go Into Recovery Mode**
		In prior 3.3.x releases, certain complex queries containing multiple aggregations could possibly fail during execution, and thereby cause one or more segments to go into recovery mode. The error messages encountered as a result of this issue were:
		`ERROR: Error on receive from segx...server closed the connection unexpectedly`
		`ERROR: GPDB performed segment reconfiguration`
		This issue has been resolved in this release.

6781, 6878	3.3.3.1	**Upgrade Utility Removes $MASTER_DATA_DIRECTORY/gpperfmon Directory** In prior releases, the upgrade utility (`gpmigrator`) did not preserve the `$MASTER_DATA_DIRECTORY/gpperfmon` directory requiring users to rerun Greenplum Performance Monitor setup after upgrading. This issue has been resolved in this release.
6788	3.3.3.0	**gprecoverseg Utility Does not Complete on Solaris** In prior 3.3.x releases, a full `/tmp` directory on Solaris systems would prevent the `gprecoverseg` utility from completing. This issue has been resolved in this release.
6790	3.3.3.0	**gpaddmirrors Utility Does not Complete on Solaris** In prior 3.3.x releases, a full `/tmp` directory on Solaris systems would prevent the `gpaddmirrors` utility from completing. This issue has been resolved in this release.
6801	3.3.3.0	**VACUUM FULL Operation Cannot be Cancelled with CTRL+C** In prior 3.3.x releases, users would encounter the following error when attempting to cancel an in progress `VACUUM FULL` operation using CTRL+C: `ERROR: Statement Error, writer gang busy: possible attempt to execute volatile function in unsupported context` This issue has been resolved in this release.
6802, 6682	3.3.3.0	**COPY FROM STDIN Failure** In prior releases, executing a `COPY FROM STDIN` command followed by a large amount of data would sometimes terminate the database session process with the following error: `server process (PID xxxx) was terminated by signal 11` This issue was caused by the database logger running out of memory attempting to log the statement and the associated data. This issue has been resolved in this release by truncating large messages written to the log.

Table 6 Resolved Issues in 3.3.3.x

Issue	Fixed In	Description
6632, 6867	3.3.3.1	**ERROR: Statement Error, writer gang busy…** In prior 3.3.x releases, cancelling a running query would sometimes cause the following error: `ERROR: Statement Error, writer gang busy: possible attempt to execute volatile function in unsupported context` This issue has been resolved in this release.
6877	3.3.3.1	**Upgrade Utility Removes $MASTER_DATA_DIRECTORY/db_dumps Directory** In prior releases, the upgrade utility (`gpmigrator`) did not preserve the `$MASTER_DATA_DIRECTORY/db_dumps` directory created by the `gpcrondump` utility. This issue has been resolved in this release.
6882	3.3.3.1	**Upgrade Utility Slower Than Prior Releases** Release 3.3.2.x introduced additional file checking to the upgrade utility (`gpmigrator`) which significantly increased the time required to complete the upgrade as compared to prior releases. This performance issue has been addressed in this release.
6908	3.3.3.1	**CREATE INDEX on Inet Datatype Column Fails** In prior releases, creating an index on a column of the inet datatype on tables with a large number of rows would sometimes fail with the following error: `ERROR:  Out of memory. Failed on request of size xxx bytes.` This issue has been resolved in this release.

Greenplum Database 3.3.2.x Release Notes

This section lists the issues that were resolved in Greenplum Database 3.3.2.x.

Table 7 Resolved Issues in 3.3.2.x

Issue	Fixed In	Description
6268	3.3.2.2	**Support for Greenplum Client Tools on AIX** Certain customers required the 3.3 Greenplum Client Tools packages on AIX. 3.3.2.2 is a release of the client tools packages only, the Greenplum Database server remains unchanged.
6477	3.3.2.0	**ERROR: could not open tablespace directory "pg_tblspc": No such file or directory** After upgrading to 3.3, certain queries of the system catalog would cause the above error. This issue has been resolved in this release of the upgrade utility (`gpmigrator`).
6497	3.3.2.0	**Query Failure When Using a NOT IN(SELECT ...) Clause** In prior 3.3 releases, certain queries containing a `NOT IN(SELECT ...)` clause would fail with the following error: `FATAL: Unexpected internal error: Master process received signal SIGSEGV` This issue has been resolved in this release.
6499	3.3.2.0	**ERROR: invalid string enlargement request size...** In prior releases, attempting to load DOS-formatted files via COPY or external tables would sometimes fail with the following error: `ERROR: invalid string enlargement request size` This issue has been resolved in this release.
6523	3.3.2.1	**Modify gpload to Allow Pre and Post SQL Commands** The `gpload` utility now allows users to specify SQL commands in the load configuration file to run before and after a load operation.

Table 7 Resolved Issues in 3.3.2.x

Issue	Fixed In	Description
6526	3.3.2.0	**Problem Upgrading Append-Only Tables** In prior 3.3 releases, the upgrade utility (`gpmigrator`) not able to upgrade append-only tables if there was a gap in the numbering of the database files on disk. This issue has been resolved in this release.

6774	3.3.2.3	**Query Failure Causes Segment to Go Into Recovery Mode**
		In prior 3.3.x releases, certain complex queries containing multiple aggregations could possibly fail during execution, and thereby cause one or more segments to go into recovery mode. The error messages encountered as a result of this issue were:
		`ERROR: Error on receive from segx...server closed the connection unexpectedly`
		`ERROR: GPDB performed segment reconfiguration`
		This issue has been resolved in this release.

6253 6335	3.3.2.0	**pg_relation_size Function in Subqueries Can Cause System to be Unresponsive** In prior 3.3 releases, running the function `pg_relation_size` in a subquery would sometimes (in rare cases) cause the system to become unresponsive to attempts to cancel the query. This issue has been resolved in this release.
6253 6335	3.3.2.0	**pg_relation_size Function in Subqueries Can Cause System to be Unresponsive** In prior 3.3 releases, running the function `pg_relation_size` in a subquery would sometimes (in rare cases) cause the system to become unresponsive to attempts to cancel the query. This issue has been resolved in this release.
6394 6418	3.3.2.0	**Error: "Could not connect to 'WAL Send Server': Operation already in progress** In prior 3.2 and 3.3 releases, some connections were interrupted when a Greenplum Database system was under heavy load. In rare cases, the system was unable to re-establish a connection, resulting in this error message, and causing issues with the log synchronization process for the standby master (if configured). This issue has been resolved in this release.
6405 6484	3.3.2.0	**Upgrade Utility Does Not Preserve SSL Certificates** In prior 3.3 releases, the upgrade utility (`gpmigrator`) did not preserve all files in a segment's data directory, such as SSL certificates. The upgrade utility will now preserve all files found in the data directory, however non-standard directories (such as backup directories) will not be migrated. Administrators must take action to migrate any non-standard directories.

Greenplum Database 3.3.1.x Release Notes

This section lists the issues that were resolved in Greenplum Database 3.3.1.x.

Table 8 Resolved Issues in 3.3.1.x

Issue	Description
6417	**Login Error when Restarting Greenplum Database with gpstart** In previous version, if Greenplum Database superusers lacked login privileges, a system restart could fail with errors like the following: `FATAL: role ""Engineering"" is not permitted to log in` This issue has been resolved in this release.
6212	**gpload Utility Fails When Loading Not Null Columns** In previous versions, a load operation with `gpload` might fail when attempting to load NOT NULL columns from a temporary table (created by `gpload` for faster update/merge) to an external table. This issue has been resolved in this release.

Table 8 Resolved Issues in 3.3.1.x

Issue	Description
6323, 6249	**Command to Start in Restricted Mode Hangs Intermittently** In previous versions, starting Greenplum Database in restricted mode (`gpstart- R`) could sometimes cause the command to hang for an extended period. This issue has been resolved in this release.
6343	**Internal Error with Commands to Set Random Distribution** In previous releases, if a user submitted incorrectly formed commands to set random distribution, a SIGSEGV error could occur. This issue has been resolved in this release.
6186	**Failure of Concurrent Inserts with generate_series on Append-Only Tables** When concurrently inserting records into append-only tables using the Postgres function `generate_series`, some of the insert operations may fail with errors like the following: `ERROR: attempted to lock invisible tuple` This issue has been resolved in this release.
5483	**Dropped Columns of User-defined Types not Supported by gpexpand or ALTER TABLE** With tables that have dropped columns of user-defined data types, you could not successfully perform redistribution with `ALTER TABLE` or with the expansion utility (which runs `ALTER TABLE` commands). This issue has been resolved in this release.
6316	**Expansion Utility Allows Unsupported Mirror Configuration in Interview** In the interactive interview process to create an expansion input file in previous releases, the utility allowed the user to specify “none” for the desired mirroring strategy, but then prompted the user for mirror data directories. This occurred when existing hosts were configured with mirrors (the same configuration is required on new hosts). In this release, “none” is not a valid interview option in any system that has mirroring configured on existing hosts.
6217	**System Expansion Performance with Extremely Large Numbers of Schema Objects** In previous releases, extremely large numbers of schema objects could prolong the offline portion of the expansion process. This issue has been resolved in this release.
6243	**System Expansion Copies Populated Partition Catalogs to New Segments** In the previous release, the system expansion utility copied populated versions of the system catalogs *pg_partition* and *pg_partition_rule* to new segments. Some aspects of Greenplum Database table partitioning expects these tables to be empty on new segments. This issue has been resolved in this release.
6183	**Database Logs Contain Numerous Lock Type Warnings** In previous versions, database log files might include large numbers of warning messages similar to the following: `WARNING","01000","you don't own a lock of type AccessShareLock` These warnings were safe to ignore, and have been removed in this release.
6443	**Query Planner not Using Constraints Implied by Join Predicates** In previous versions, the query planner did not always use constraints implied by join predicates. This applied specifically to BETWEEN and IN clauses and resulted in the need to scan potentially very large data sets before processing joins. This issue has been resolved in this release.
6423	**Executing Immutable Functions as a Non-Superuser Causes SIGSEGV** In previous versions, executing an immutable function as a non-superuser could cause a SIGSEGV error and a system core dump. This issue has been resolved in this release.

Table 8 Resolved Issues in 3.3.1.x

Issue	Description
6341	**Performance Issue with Count Distinct on a Distribution Key Column** Running a SELECT command with count distinct on a distribution key column of a table could result in an inefficiently optimized query plan. This issue has been resolved in this release.
6340	**Group Aggregation Control Parameter Ignored** In previous versions, although the parameter `enable_groupagg` was set to `off` (default is `on`), the query planner still used group aggregation plan types in certain types of queries. This issue has been resolved in this release.

Greenplum Database 3.3.0.x Release Notes

Greenplum Database 3.3 is a major release which introduces several new features, performance and stability enhancements, and internal changes to the system catalogs. Please refer to the following sections for more information about this release:

New Features in Greenplum Database 3.3

System Expansion and Scaling

This release introduces the administrative utility `gpexpand` for adding additional hardware resources to an existing Greenplum Database system in order to scale performance and storage capacity. The procedure to expand a system with `gpexpand` has been designed to minimize offline time during system expansion and allow flexible scheduling of expansion operations.

Supporting Documentation for System Expansion and Scaling

For more information on system expansion and scaling, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 22, Expanding a Greenplum System
- Appendix B, Management Utility Reference, “gpexpand”

Automatic Log Rotation and Enhanced Log Search Capability

This release introduces two enhancements to Greenplum Database logging. The first is automatic log rotation. In prior releases, the master and each segment would log to a the same directory location as the instance’s data directory and automatic log rotation was not enabled. This release now uses the default PostgreSQL log location (the `pg_log` directory inside the instance’s data directory) and automatically rolls over log files every 24 hours.

In addition to automatic log rotation, the log format is now CSV, making it easier to parse and search the log files. A new log search utility, `gplogfilter`, has also been added to help administrators locate specific information in the logs.

Important: These features affect some log-related parameters. For more information, see the Parameters with Changed Defaults and Defunct Parameters in this document, and refer to the supporting documentation.

Supporting Documentation for Automatic Log Rotation and Logging Enhancements

For more information on automatic log rotation and log search capability, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 24, Routine System Maintenance Tasks, “Managing Greenplum Database Log Files”

pgAdmin III for Greenplum Database

With this release, Greenplum introduces an enhanced, Greenplum-compatible version of the popular management tool pgAdmin III. This tool supports PostgreSQL databases with the same features pgAdmin users are familiar with, while adding support for Greenplum-specific features such as resource queues, table partitioning, external tables, and append-only tables. Installation packages are available for download from Greenplum Network and from the pgAdmin download site.

Supporting Documentation for pgAdmin III

For more information on pgAdmin III for Greenplum Database, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 12, Accessing the Database, “Supported Client Applications”

Upgrade Utility Enhancements

The upgrade utility (`gpmigrator`) has been improved in this release to provide better performance and scalability for upgrading large Greenplum Database systems to the 3.3 release. See “Upgrading to Greenplum Database 3.3.7.x” on page 5 for more information about upgrading.

Supporting Documentation for the Upgrade Utility

For more information on the upgrade utility, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Appendix B, Management Utility Reference, “gpmigrator”

Jetpack Administrative Schema

Version 3.3 introduces a set of administrative tools for common tasks such as viewing logs, checking parameter values, and diagnosing skew and table bloat. These views, queries and functions comprising the schema *gp_jetpack* are packaged in the `jetpack.sql` file provided in the `$GPHOME/lib/jetpack` directory of the 3.3 installation. Install *gp_jetpack* explicitly for each database where you want to use Jetpack tools.

Supporting Documentation for the Jetpack

For more information on the Jetpack schema, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

Chapter 23, Monitoring a Greenplum System, “Using the JetPack Administrative Interface”

Enhanced Subquery Support

This release introduces support for subqueries with `NOT IN` and `NOT EXISTS`. The version 3.3 query planner now evaluates various strategies for optimizing these types of subqueries. In query plans with `NOT IN` and `NOT EXISTS` you may observe the following types of join operations:

- Hash Left Anti Semi Join
- Merge Left Anti Semi Join
- Nested Loop Left Anti Semi Join

Supporting Documentation for Enhanced Subquery Support

For more information on new support for subqueries, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Appendix A, SQL Command Reference, “SELECT”

Enhanced Fault Detection

Fault detection in version 3.3 is improved by the introduction of the new backend process *fts_prober*. This process monitors the Greenplum array, scanning all segments and database processes at configurable intervals. When an issue is detected, *fts_prober* logs information about the event in a new system catalog table named *gp_configuration_history*. These improvements greatly reduce the time to detect and correct segment failures.

Supporting Documentation for Fault Detection

For more information on fault detection, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 20, Enabling High Availability Features, “Knowing When a Segment is Down”
- Appendix I, System Catalog Reference, “gp_configuration_history”

Distinct-Qualified Aggregate Enhancements for Grouping

This release introduces query tuning options to produce more efficient query plans when computing distinct-qualified aggregates for OLAP grouping queries. With these new parameters enabled, the query planner creates alternate plans and dynamically chooses the more efficient plan.

Supporting Documentation for Multiple Distinct-Qualified Aggregate Enhancements

For more information on multiple distinct-qualified aggregate enhancements, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 19, Configuring your Greenplum System, "Query Tuning Parameters"
- Appendix D, Server Configuration Parameters

Improved Sorting Performance

Version 3.3 introduces enhancements to the way Greenplum Database compares strings to perform sort operations. Depending on the data type and distribution of the data to sort, these enhancements can result in performance improvements over previous releases. Particularly significant performance improvements should be attained when sorting large numbers of columns, especially when the leading columns contain duplicate values.

Functions to Check Compression and Distribution for AO Tables

With version 3.3, Greenplum provides built-in functions to check the compression ratio and the distribution of append-only tables. Superuser privileges are required to run these function.

Table 9 Functions for compressed append-only table metadata

Return Type	Full Syntax	Description
`Set of (dbid, tuplecount) rows`	`get_ao_distribution(oid,name)`	Shows the distribution of rows of an append-only table across the array. Returns a set of rows, each of which includes a segment *dbid* and the number of tuples stored on the segment.
`float8`	`get_ao_compression_ratio(oid,name)`	Calculates the compression ratio for a compressed append-only table. If information is not available, this function returns a value of -1.

Supporting Documentation for AO Table Functions:

For more information on built-in functions to check the compression ratio and the distribution of append-only tables, see the following sections of the *Greenplum Database 3.3 Administrator Guide* (`GPAdminGuide.pdf`):

- Chapter 14, Defining Database Objects, "Checking the Compression and Distribution of an Append-Only Table."

Changed Features in Greenplum Database 3.3

- SQL Command Changes
- Server Configuration Parameter Changes
- System Catalog Changes

Management Utility Changes

The following management utilities have new or changed command-line options in 3.3:

Table 10 Management Utility Changes in 3.3

Utility	New / Changed Options	Deprecated Options
`gpexpand`	New utility for system expansion.	
`gplogfilter`	New utility for searching logs.	

`gpssh-exkeys`	`-e existing_hosts_file`
	`-x expansion_hosts_file`
	New options for use in key exchange for new expansion hosts.	

`gpmigrator`	`--debug` `-R (revert)`	

`gpstop`	`-v (verbose output)`	`-z`
		`-t`
		`-f , -i ,-s` -- these are replaced by `[-M smart] | fast] | immediate]`

`gpstart`	`--recover` (attempt to restart failed segments)	`-e` `-s`
`gpfdist`	`-t (timeout)` `-m (max_length)`	
`gprecoverseg`		`-T`
`gpstate`		`-n -e -r -g`
`pg_dump`	`--no-gp-syntax` (do not display Greenplum syntax when creating a dump file)	
`gpdbrestore`	`--noanalyze` (suppresses automatic analyze on restored tables)	
`gpdetective`		`-f output_file`

SQL Command Changes

The following SQL commands have new or changed syntax in 3.3.

Table 11 Changed SQL Commands in 3.3

Command	Description of Change
`SELECT`	`AS` keyword now optional for column aliases.
`ALTER TABLE`	New syntax to change a table's distribution policy: `SET DISTRIBUTED BY/DISTRIBUTED RANDOMLY`. New syntax to redistribute a table: `ALTER TABLE SET WITH (REORGANIZE=TRUE)` New support for subpartition templates: `SET SUBPARTITION TEMPLATE` **NOTE:** These commands are not compatible with pre-3.2 partitions created with the deprecated scripts `gpaddpart` and `gpcreatepart`.
`COPY, CREATE EXTERNAL TABLE`	Enables `FORCE NOT NULL` in external tables. New syntax added for `FILL MISSING FIELDS` `DELIMITER` now has `OFF` option for loading unstructured data into a single column (for special use cases, especially with MapReduce)
`CREATE EXTERNAL TABLE`	`LIKE` clause accepted to copy column/data type definitions from another table. Column constraints are ignored by external tables.

Server Configuration Parameter Changes

- New Parameters
- Parameters with Changed Defaults
- Defunct Parameters

New Parameters

Table 12 New Server Configuration Parameters in 3.3

Parameter	Description
cursor_tuple_fraction	Tells the query planner how many rows are expected to be fetched in a cursor query, thereby allowing the planner to use this information to optimize the query plan. The default of 1 means all rows will be fetched.
gp_statistics_pullup_from_child_partition	Enables the query planner to utilize statistics from child tables when planning queries on the parent table. Enabled by default.
gp_enable_groupext_distinct_pruning	Enables or disables three-phase aggregation and join to compute distinct-qualified aggregates on grouping extension queries. Usually, enabling this parameter generates a cheaper query plan that the planner will use in preference to existing plan. Enabled by default.
gp_enable_groupext_distinct_gather	Enables or disables gathering data to a single node to compute distinct-qualified aggregates on grouping extension queries. When this parameter and `gp_enable_groupext_distinct_pruning` are both enabled, the planner uses the cheaper plan. Enabled by default.
gp_enable_predicate_propagation	When enabled, the query planner applies query predicates to both table expressions in cases where the tables are joined on their distribution key column(s). Filtering both tables prior to doing the join (when possible) is more efficient.

Table 12 New Server Configuration Parameters in 3.3

Parameter	Description
gp_fts_probe_interval	Specifies the cycle time for the `fts_prober` process. Another complete probe of all segments starts each time this period expires. The `fts_prober` process will take approximately this amount of time to detect segment failures. Typically, this parameter is set to the same value as *gp_segment_connect_timeout*. Default is `60` (seconds).
gp_fts_probe_threadcount	Specifies the number of `fts_prober` threads to create. This parameter should be set to a value equal to or greater than the number of segments per host. Default is `5`.
gp_gpperfmon_send_interval	Sets the frequency at which backend server processes send iterator data to segment monitor agents. Performance data for query plan iterator nodes executed during this interval are sent through UDP to the segment monitor agents. If you find that an excessive number of UDP packets are dropped during long-running, complex queries, you may consider increasing this value.
gp_log_format	Specifies the format of the server log files. Valid values are `text` and `csv`. If you want to use the administrative jetpack tools, the log files must be in CSV format.
gp_max_csv_line_length	The maximum length of a line in a CSV formatted file that will be imported into the system. May need to be increased if using the jetpack administrative views. Maximum allowed is 1048575 (1MB).
gp_statistics_pullup_from_child_partition	Enables the query planner to utilize statistics from child tables when planning queries on the parent table. Enabled by default.
gp_statistics_use_fkeys	When enabled, allows the optimizer to use foreign key information stored in the system catalog to optimize joins between foreign keys and primary keys.
gp_vmem_protect_gang_cache_limit	If a query executor process consumes more than this configured amount, then the process will not be cached for use in subsequent queries after the process completes. Systems with lots of connections or idle processes may want to reduce this number to free more memory on the segments.
IntervalStyle	Sets the display format for interval values. The value *sql_standard* produces output matching SQL standard interval literals. The value *postgres* produces output matching PostgreSQL releases prior to 8.4 when the DateStyle parameter was set to ISO. The value *postgres_verbose* produces output matching Greenplum releases prior to 3.3 when the DateStyle parameter was set to non-ISO output. The value *iso_8601* will produce output matching the time interval *format with designators* defined in section 4.4.3.2 of ISO 8601. See the PostgreSQL 8.4 documentation for more information.
log_autostats	Logs information about automatic `ANALYZE` operations related to gp_autostats_mode and gp_autostats_on_change_threshold.
log_timezone	Sets the time zone used for timestamps written in the log. Unlike TimeZone, this value is system-wide, so that all sessions will report timestamps consistently. The default is `unknown`, which means to use whatever the system environment specifies as the time zone.

Parameters with Changed Defaults

The following parameters have changed default values from 3.2.x.x:

- `escape_string_warning=on` (was off in 3.2.4)
- `gp_adjust_selectivity_for_outerjoins=on` (was off in 3.2.4)
- `gp_autostats_mode=on_no_stats` (was `none` in 3.2.4)

- log_filename=gpdb-%Y-%m-%d_%H%M%S.csv
- log_rotation_size=0 (was 10MB in 3.2.4)
- max_appendonly_tables=2048 (was 256 in 3.2.4)
- tcp_keepalives_count=0 (was 9 in 3.2.4)
- tcp_keepalives_idle=0 (was 7200 in 3.2.4)
- tcp_keepalives_interval=0 (was 60 in 3.2.4)

Defunct Parameters

The following parameters are disabled in 3.3, and will be removed in the next major release of Greenplum Database. If these are referenced in your postgresql.conf files, Greenplum recommends that you disable them and plan for their future removal.

- redirect_stderr
- log_line_prefix
- log_directory
- log_destination
- syslog_facility
- syslog_ident

Note: These defunct parameters are disabled. If you specify values for them, Greenplum Database version 3.3 will output warnings.

System Catalog Changes

The following new system catalog tables and views have been added or modified in 3.3.

Table 13 New System Catalogs in 3.3

Catalog	Description
gp_configuration_history	New table to store history of fault detection related configuration changes.
gp_db_interfaces	New table to store information about the relationship of segments to network interfaces.
gp_interfaces	New table to store information about network interfaces on segment hosts.
gpexpand.status	New table to store status information for expansion operations.
gpexpand.status_detail	New table to store the detailed status of tables in expansion operations.
gp_appendononly	New column for future feature development: *compresstype* column added to support multiple compression types for append-only tables.

Resolved Issues in Greenplum Database 3.3.0

This section lists the customer reported issues that were resolved in Greenplum Database 3.3.0.x:

Table 14 Resolved Issues in 3.3.0.x

Issue	Category	Description
2553	fault detection	**First Query of a Session Always Requires a Reconnect in readonly Mode** When running in readonly fault tolerance mode and a segment is down, the first query of a session will always attempt to reconnect when a segment is down. This is because failed segments are not marked as invalid in the *gp_configuration* table when running in read-only mode. This issue is resolved in version 3.3, in which invalid segments are updated in the new system catalog, *gp_configuration_history*.
3530	transaction management	**Concurrent Vacuum Operations Cause Bitmap Index Errors** In 3.1.1.x releases, multiple concurrent VACUUM operations on tables with a bitmap index can sometimes cause the following error: `ERROR: could not open relation...`
4707	DDL/DML	**Error when Writing Data to More Than 256 Append-Only Tables** In previous releases, the default value for the server configuration parameter max_appendonly_tables was 256, and operations exceeding that threshold failed. The new default value for this parameter is 2048.
4725	DDL/DML	**ALTER TABLE ADD PARTITION does not Create Default Subpartitions** In previous releases, additional commands were required to create default subpartitions when using `ALTER TABLE ADD PARTITION`. This issue has been resolved in this release.
4957	query execution	**SEGV Error with ANALYZE on a Dropped Temporary Table** In previous releases, a SEGV error could be encountered when running `ANALYZE` on temporary tables. This occurred when a temporary table or namespace had been dropped in a different session, and the system could not resolve the name of temporary objects. In version 3.3, this situation is handled by returning the message, "`Unable to collect rel stats from segment databases. Schema/Relation probably does not exist.`"
5085	DDL/DML	**Backup and Restore does not Preserve Subpartition Template Definition** In previous releases, a backup created with the `gp_dump` utility might not correctly preserve a single-level subpartition template defintion. In such cases, when the system was restored from the backup file, subpartitions could not be created using the subpartition template. This issue is resolved for single-level partitions.
5204	management utilities	**Migration Fails Importing Schema with Escape Characters in External Table Definition** In previous releases, the migration utility could not import schemas that used apostrophes for the escape character in external table definitions. This issue has been resolved in this release.
5255	management utilities	**gprestore Utility Fails when -d Option is Omitted** In previous releases, the gprestore utility did not fail gracefully in cases where `-d` (database) was omitted. This issue has been resolved in this release.
5294	query execution	**Session Failure when Repeatedly Running a Stored Procedure** When running a stored procedure a second time in the same session (having run the procedure successfully once), the connection is lost and the session terminated. This issue has been resolved in this release.

Table 14 Resolved Issues in 3.3.0.x

Issue	Category	Description
5319	query execution	**Queries fail with Error "Planner committed to impossible hash aggregate"** When the server parameter `gp_statistics_pullup_from_child_partition` (introduced in version 3.2.3) is disabled, queries on partitioned tables may fail with the error, `Planner committed to impossible hash aggregate`. This issue has been resolved in this release.
5338	transaction management	**Calling a Stored Procedure from within Another Procedure Subverts Resource Queues** When a stored procedure is called from within another procedure, transaction management rules set by resource queues will not be applied to the second procedure. This issue has been resolved in this release.
5492	fault detection	**Segment Connections Refused by Standby Master** In some scenarios when the system fails over from the primary master to the standby master, segment connections are refused due to insufficient permissions as defined by `pg_hba.conf`. This is resolved in version 3.3, in which the master uses a different mechanism to authenticate segment connections.
5635, 4476	fault detection	**"DTM transaction is not active" Error with Segment Failure in continue Mode** In a system configured to continue database operations in case of failures, a segment failure could cause running SQL statements to return errors similar to the following: `ERROR: DTM transaction is not active (CdbDoCommand, detail = 'set gp_write_shared_snapshot=true')` This issue has been resolved in this release.
5699	DDL/DML	**Ownership not Propagated to Leaf Partitions** In previous releases, leaf partitions did not inherit ownership attributes from the parent table. This issue has been resolved in version 3.3.
5831	DDL/DML	**Splitting Partitions by Date Range with Timestamp without Time Zone Fails** When splitting a partition by date range defined by the data type timestamp without time zone, the command could fail with errors like the following: `ERROR: partition being split has 1 column but parameter has x columns` This issue has been resolved in this release.
5858	DDL/DML	**Split Partitions do not Inherit Storage Attributes** In previous releases, leaf partitions created by splitting a partition did not inherit the root partition's storage attributes. This issue has been resolved in version 3.3.
5870	query execution	**Queries Fail with "ERROR: invalid attribute number 7 (heaptuple.c:1202)"** In rare cases in previous releases, a query could fail with the following error: `ERROR: invalid attribute number 7 (heaptuple.c:1202)` This issue has been resolved in this release.
6005	management utilities	**gpfdist does not Load Text Columns Longer than 32K** In previous releases, `gpfdist` set a limit of 32K characters for text columns. Release 3.3 introduces a new `gpfdist` option, `-m`, to allow adjusting that limit as high as 1MB. Default is 32768 bytes. Higher values incur increased resource consumption.

Table 14 Resolved Issues in 3.3.0.x

Issue	Category	Description
6028	query execution	**Issues with OLAP Queries with Window Functions and Group By Clauses** Some OLAP queries that contained both window functions and group-by clauses generated unexpected results. This issue has been resolved in this release.
6045	management utilities	**gpload does not Always Terminate on Completion of Load Operation** In previous releases, the `gpload` utility may not terminate after a load operation had completed and the transaction had been committed. This issue has been resolved in this release. Note: the improved version of `gpload` no longer prints status messages such as `XXXX bytes loaded out of YYYY bytes.`