

## Welcome to Greenplum Database 4.1

---

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

---

### About Greenplum Database 4.1

Greenplum Database 4.1 introduces a number of significant new features, as well as performance and stability enhancements. Please refer to the following sections for more information about this release:

- [New Features in Greenplum Database 4.1](#)
- [Changed Features in Greenplum Database 4.1](#)
- [Resolved Issues in Greenplum Database 4.1](#)
- [Known Issues in Greenplum Database 4.1.0.0](#)
- [Upgrading to Greenplum Database 4.1](#)
- [Greenplum Database 4.1 Client Tool Packages](#)
- [Greenplum Database 4.1 Documentation](#)

---

### New Features in Greenplum Database 4.1

Greenplum Database 4.1 offers the following new features:

- [Enhanced Workload Management: Statement-Level Memory Control](#)
- [Tight Hadoop Integration](#)
- [Simpler, Faster Installation: Parallel Installation Utility](#)
- [Better In Database Analytics](#)
- [Enhanced Support of Joins for UPDATE and DELETE](#)
- [Ordered Aggregates and Ordered Reduce Specifications in Greenplum MapReduce](#)
- [Transaction Performance Enhancement](#)

- Enhanced gpload Performance to Avoid Catalog Bloat
- Online Segment Rebalancing Following a Recovery
- Performance Monitor Enhancements
- PostgreSQL 9.0 Client Compatibility
- Oracle Unbreakable Linux 5.5 Support

### Enhanced Workload Management: Statement-Level Memory Control

This release enhances Greenplum Database Workload Management by providing greater control of memory utilization for concurrent workloads. Resource queues have a new `MEMORY_LIMIT` attribute that allows administrators to control memory usage for all queries submitted through a resource queue. This limit determines the total amount of memory that all worker processes of a query can consume on a segment host during query execution. The default amount of memory allotted per query on statement-based queues is: `MEMORY_LIMIT / ACTIVE_STATEMENTS`.

Memory can also be allotted on a per-query basis using the new `statement_mem` server configuration parameter (database superusers can set an upper limit using `max_statement_mem`). When using the new memory features of 4.1, these parameters replace the `work_mem`, `max_work_mem` and `maintenance_work_mem` parameters.

### Tight Hadoop Integration

Greenplum Database external tables now support reading files from and writing files to a Hadoop File System (HDFS). This allows administrators to load data in parallel from HDFS as well as use Greenplum MapReduce jobs to access data on HDFS using the power of SQL and Greenplum advanced analytic functions.

### Simpler, Faster Installation: Parallel Installation Utility

The `gpseginstall` parallel installation utility allows system administrators to install the Greenplum Database software on multiple hosts at once. When run as `root`, it also automates other 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. See the *Greenplum Database Installation Guide* for more information on using this new utility.

### Better In Database Analytics

This release contains three new functions for advanced in-database analytics. There are two new linear regression aggregate functions, `t-statistics` and `p-values`, that complement the existing functions that calculate the regression coefficients and r-squared values for an ordinary-least-squared linear regression.

There is a new Naïve Bayes function, `nb_probabilities`, that calculates the probability for each classification in a Naïve Bayes analysis. See the *Greenplum Database Administrator Guide* for new detailed use cases and examples for these advanced analytic functions.

## Enhanced Support of Joins for UPDATE and DELETE

Prior releases of the Greenplum Database query planner did not support runtime redistribution of rows for UPDATE and DELETE commands. This meant that to execute an UPDATE or DELETE by joining two or more tables, the tables had to be hash distributed and have the same distribution key. This limitation has been removed in this release except when updating the distribution key or updating the partition key of a partitioned table which requires moving tuples to a different partition. Joins for UPDATE and DELETE are now supported the same as for SELECT queries.

## Ordered Aggregates and Ordered Reduce Specifications in Greenplum MapReduce

Support has been added for ordered aggregate functions, providing a method for controlling the order in which values are fed to an aggregate function. In Greenplum Database, only aggregate functions defined as ORDERED can be called with an ORDER BY clause after their ordinary arguments to specify a system-defined ordering.

The three built-in ordered aggregates and optional ORDER BY clauses that have been implemented in 4.1, are shown in the following table.

**Table 1** Built-in Ordered Aggregate Functions in Greenplum Database 4.1

Aggregate Function	Description
array_agg(anyelement)	Concatenates anyelement into an array. Example: SELECT array_agg(anyelement ORDER BY anyelement) FROM table;
string_agg(text)	Concatenates text into a string. Example: SELECT string_agg(text ORDER BY text) FROM table;
string_agg(text, delimiter)	Concatenates text into a string delimited by delimiter. Example: SELECT string_agg(text, ',' ORDER BY text) FROM table;

The columns in an ORDER BY clause are not necessary the same as the aggregated column, as shown in the following statement that references a table named product with columns store\_id, product\_name, and quantity.

```
SELECT store_id, array_agg(product_name ORDER BY
quantity desc) FROM product GROUP BY store_id;
```

**Note:** There can only be one aggregated column. Multiple columns can be specified in the ORDER BY clause.

This support for ordered aggregates extends to Greenplum MapReduce (gpmapreduce). Input to a reduce specification can now be ordered on some combination of key/value columns allowing for efficient handling of the data for algorithms that require ordered input.

## Transaction Performance Enhancement

This release streamlines the way Greenplum Database handles two-phased transaction transactions. Customer environments with high-levels of concurrent transactions (20 or more concurrent sessions running transactions) and/or high numbers of small transactions (such as single-row INSERTS) will notice up to a 30 percent performance improvement.

### Enhanced gpload Performance to Avoid Catalog Bloat

The `gpload` utility has been enhanced to better support trickle load workflows. Administrators can now set `REUSE_TABLES` in the `gpload` control file. If set to `true`, `gpload` will not drop the external table objects and staging table objects it creates. These objects will be reused for future load operations that use the same load specifications. This reduces churn in the system catalog tables caused by trickle loads (ongoing small loads to the same target table).

### Online Segment Rebalancing Following a Recovery

After running the recovery utility (`gprecoverseg`), segment instances are not automatically 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 than is optimal for top system performance. In prior releases, a system restart was required in order to rebalance your Greenplum Database array. In 4.1, segments can now be rebalanced while the system is online by running a segment rebalance operation (`gprecoverseg -r`) following segment recovery. Any active queries will be rolled back, however client sessions will remain connected.

### Performance Monitor Enhancements

In this release, the Performance Monitor data collection agents gather additional metrics to help administrators analyze network patterns of Greenplum Database. Metrics are now collected for Greenplum interconnect and primary-to-mirror replication processes. Collecting these metrics allows system administrators to pinpoint the cause of network issues, and separate hardware issues from software issues. See the *Greenplum Performance Monitor Administrator Guide* for more information about the new metrics being collected.

### PostgreSQL 9.0 Client Compatibility

In this release, the client side applications and libraries have been upgraded to PostgreSQL 9.0. This allows users to take advantage of the latest features and bug fixes to PostgreSQL clients and client-side authentication methods. Database users can use the latest PostgreSQL clients (such as `psql`) to connect to Greenplum Database. Greenplum Database is also backwards compatible with 8.x PostgreSQL client applications.

### Oracle Unbreakable Linux 5.5 Support

Greenplum Database 4.1 is now certified on Oracle Unbreakable Linux kernel version 5.5.

---

## Changed Features in Greenplum Database 4.1

- [SQL Commands](#)
- [Management Utilities](#)
- [Client Utilities](#)
- [Server Configuration Parameters](#)
- [System Catalogs](#)

## SQL Commands

**Table 2** SQL Command Changes in 4.1

SQL Command	Description of Change
ALTER RESOURCE QUEUE	New <code>MEMORY_LIMIT</code> resource queue attribute to set memory quota.
ALTER ROLE	New <code>protocol='gpdfs'</code> value added to the <code>CREATEEXTABLE   NOCREATEEXTABLE</code> role attributes (allows granting of privileges for Hadoop-based external tables).
CREATE AGGREGATE	New <code>ORDERED</code> keyword added for defining an aggregate as an ordered aggregate. For example: <code>CREATE [ORDERED] AGGREGATE name (input_data_type [ , ... ])</code> For ordered aggregate functions, you can control the order of input values at runtime using an <code>ORDER BY</code> clause following the ordinary input arguments. For example: <code>aggregate_name (expression [ , ... ] [ORDER BY orderspec [, ...]] )</code> <code>[FILTER (WHERE condition)]</code>
CREATE EXTERNAL TABLE	New <code>gpdfs</code> protocol added for Hadoop-based external tables.
CREATE RESOURCE QUEUE	New <code>MEMORY_LIMIT</code> resource queue attribute to set memory quota.
CREATE ROLE	New <code>protocol='gpdfs'</code> value added to the <code>CREATEEXTABLE   NOCREATEEXTABLE</code> role attributes (allows granting of privileges for Hadoop-based external tables).

## Management Utilities

**Table 3** Management Utility Changes in 4.1

Utility Name	Description of Change
<code>gprecoverseg</code>	New <code>-r</code> (rebalance) option added. This allows administrators to return all segments to their preferred role after a segment host recovery without having to shut down Greenplum Database. To use this option, all segments must be up and synchronized. Rebalancing will cause any in progress queries to be rolled back.
<code>gpmapreduce</code>	In the YAML control file, an optional <code>ORDERING</code> attribute has been added to the <code>REDUCER</code> specification. For example: <code>REDUCER:</code> <code>  NAME reducer_name</code> <code>  ORDERING [column [description] [,...]]</code> The <code>ORDERING</code> attribute is only supported in documents that have a <code>VERSION</code> of 1.0.0.3 or higher.
<code>gpload</code>	New <code>REUSE_TABLES</code> option added to the <code>PRELOAD</code> section of the YAML load control file. If set to true, <code>gpload</code> will not drop the external table objects and staging table objects it creates. These objects will be reused for future load operations that use the same load specifications. This improves performance of trickle loads (ongoing small loads to the same target table).
<code>gpmigrator</code> <code>gpmigrator_mirror</code>	When upgrading to a new version of the Greenplum Databases, use <code>gpmigrator</code> , if your system does not have mirrors. <code>gpmigrator_mirror</code> , if your system has mirrors.

## Client Utilities

**Table 4** Client Utility Changes in 4.1

Client Utility	Description of Changes
psql	<p><b>Command-Line and Environment</b></p> <p>New command line option: <code>-w   --no-password</code></p> <p>Support for new <code>application_name</code> parameter and <code>PGAPPNAME</code> environment variable</p> <p><b>Meta Commands</b></p> <p>Consistent use of capital <code>s</code> (include system objects) in <code>\d</code> meta commands</p> <p>More <code>\d</code> meta commands now support <code>+</code> (show detailed info)</p> <p>New <code>\conninfo</code> meta command to see how you are connected</p> <p>New <code>\prompt</code> meta command to prompt the user for substitution variables</p> <p>New <code>\pset format</code> option: <code>wrapped</code></p> <p>New <code>\pset columns</code> meta command to set width for <code>format wrapped</code></p> <p>New <code>\pset linestyle</code> meta command (values are <code>unicode</code>, <code>ascii</code>, or <code>old-ascii</code>)</p> <p>New <code>\pset pager</code> option: <code>always</code></p> <p><code>\timing</code> can now take arguments: <code>on   off</code></p> <p><code>\pset tuples-only</code> can now take arguments: <code>on   off</code></p>

## Server Configuration Parameters

- [Deprecated Parameters](#)
- [New Parameters](#)
- [Parameters with Changed Defaults](#)

### Deprecated Parameters

**Table 5** Deprecated Server Configuration Parameters

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

## New Parameters

**Table 6** New Server Configuration Parameters in 4.1

Parameter Names	Description
<code>application_name</code>	Sets the application name for a client session. For example, if connecting via <code>psql</code> , this parameter will be set to <code>psql</code> . Setting an application name allows it to be reported in log messages and statistics views.
<code>gp_connectemc_mode</code>	Controls the ConnectEMC event logging and dial-home capabilities of Greenplum Performance Monitor on the EMC Greenplum Data Computing Appliance (DCA). ConnectEMC must be installed in order to generate events. Allowed values are: <ul style="list-style-type: none"> <li>• <code>on</code> (the default) - log events to the <code>gpperfmon</code> database and send dial-home notifications to EMC Support</li> <li>• <code>off</code> - turns off ConnectEMC event logging and dial-home capabilities</li> <li>• <code>local</code> - log events to the <code>gpperfmon</code> database only</li> </ul>
<code>gp_fts_probe_timeout</code>	Sets the allowed timeout for the fault detection process ( <code>ftsprobe</code> ) to establish a connection to a segment instance. The default is 3 minutes. In prior releases, the interconnect timeout and the fault detection timeout were controlled by the same parameter ( <code>gp_segment_connect_timeout</code> ).
<code>gp_resqueue_memory_policy</code>	Enables the new memory management features of 4.1. When set to <code>none</code> , memory management is the same as in 4.0 (using <code>work_mem</code> ). When set to <code>auto</code> , query memory usage is controlled by <code>statement_mem</code> and/or resource queue memory limits. The <code>work_mem</code> parameter becomes obsolete when this is enabled.
<code>gp_workfile_checksumming</code>	Adds a checksum value to each block of a work file (or spill file) used by <code>HashAgg</code> and <code>HashJoin</code> query operators. This adds an additional safeguard from faulty OS disk drivers writing corrupted blocks to disk. When a checksum operation fails, the query will cancel and rollback rather than potentially writing bad data to disk. Default is <code>on</code> .
<code>max_statement_mem</code>	Sets the maximum memory limit for a query. This parameter can only be set by database superusers. It prevents regular users from oversubscribing <code>statement_mem</code> . Default is 2000MB.
<code>statement_mem</code>	Sets a per-query memory limit. When <code>gp_resqueue_memory_policy</code> is enabled, this parameter replaces <code>work_mem</code> . Default is 125MB.
<code>smdw_aliases</code>	This is a new parameter added to the Greenplum Performance Monitor configuration file <code>gpperfmon.conf</code> (not <code>postgresql.conf</code> ) that allows the monitor agents to use multiple interfaces to the standby master host ( <code>smdw</code> ). For example: <code>smdw_aliases=smdw-1,smdw-2</code>

## Parameters with Changed Defaults

**Table 7** Changed Server Configuration Parameter in 4.0.5.0

Parameter Name	Old Value	New Value	Description of Change
gp_resqueue_priority	OFF	ON	Enables query prioritization for resource queues. Setting this parameter to ON by default is a convenience for users who decide to implement PRIORITY settings on their workload management resource queues.
gp_resqueue_priority_cpucores_per_segment	1	24 on master 4 on segments	These are the appropriate default settings for the Greenplum Data Computing Appliance.

## System Catalogs

### Deprecated Catalog Objects

Deprecated catalog objects will be removed in the next major release of Greenplum Database. Any user-defined objects (views, functions, and so on) that reference these objects should be updated accordingly.

**Table 8** Deprecated Catalog Objects

Object Type	Schema	Catalog Name	Description
table	pg_catalog	pg_resqueue	Functionality moved to the <code>pg_resqueuecapability</code> table
view	pg_catalog	pg_resqueue_status	Use the <code>gp_toolkit.pg_resqueue_status</code> view instead

### Changed Catalog Objects

**Table 9** Changed Catalog Objects

Object Type	Schema	Catalog Name	Description
view	pg_catalog	pg_locks	New columns added to show locks held on a segment instance: <code>mppsessionid</code> - The client session id. <code>mppiswriter</code> - Is the lock held by a writer process? <code>gp_segment_id</code> - The Greenplum segment (dbid) where the lock is held. Prior releases only showed locks held on the master.
view	pg_catalog	pg_stat_activity	New columns added: <code>application_name</code> - Name of client application. <code>xact_start</code> - Transaction start time.



## Resolved Issues in Greenplum Database 4.1

This section lists the customer reported issues that are now resolved in Greenplum Database 4.1.0.0:

**Table A.1** Resolved Issues in 4.1.0.0

Issue Number	Category	Description
6567	Data Loading	<p><b>gpload Error when Using password Authentication</b></p> <p>In prior releases, the <code>gpload</code> utility did not support <code>password</code> authentication (only <code>md5</code>). When password authentication was used, <code>gpload</code> would fail with the following error:</p> <pre>ERROR unexpected error - backtrace written to log file</pre> <p>This issue has been resolved in this release - password authentication is now supported.</p>
6571	Performance Monitor	<p><b>Missing or Incorrect Performance Monitor System Metrics</b></p> <p>In prior releases of Greenplum Performance Monitor, some rows in the <code>system_history</code> table would have incorrect numbers or missing data for the <code>net_rb_rate</code>, <code>net_wb_rate</code>, <code>net_rp_rate</code> and <code>net_wp_rate</code> columns. This issue has been resolved in this release.</p>
10848	Management Utilities	<p><b>Management Utilities Should Place Mirror on Different Subnet than its Primary</b></p> <p>In prior releases, the Greenplum Database management utilities such as <code>gpinitssystem</code>, <code>gpaddmirrors</code> and <code>gpexpand</code> 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.</p>
11696	Fault Detection and Recovery	<p><b>gprecoverseg: Multi-Segment Recovery Process Unresponsive</b></p> <p>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 <code>gprecoverseg</code> utility became unresponsive and segments were not recovered as expected. This issue has been resolved in this release.</p>
11707, 11006	Management Utilities	<p><b>Cannot Start Greenplum Database when a NIC is Down</b></p> <p>In prior releases, the Greenplum Database startup utility, <code>gpstart</code>, failed to start Greenplum Database if a segment host had a failed network interface (NIC). When this occurred, the <code>gpstart</code> utility became unresponsive. This issue has been resolved in this release.</p>
11713	Fault Detection and Recovery	<p><b>Failed Segment Unable to Failover to Mirror</b></p> <p>In prior 4.0 releases, certain heavy load conditions could cause primary-to-mirror segment communications to become unresponsive. When this occurred, Greenplum Database was unable to transition operations over to the mirror when a primary segment failed, leaving Greenplum Database in an unresponsive state. This release introduces a fix to the primary-to-mirror segment communication processes to prevent this issue in the future.</p> <p>In addition to this fix, customers experiencing this issue should also set the <code>tcp_keepalives</code> server configuration parameters to the recommended values by running the following series of commands on the master host:</p> <pre>\$ gpconfig -c tcp_keepalives_idle -v 180 \$ gpconfig -c tcp_keepalives_count -v 2 \$ gpconfig -c tcp_keepalives_interval -v 75 \$ gpstop -r</pre>

**Table A.1** Resolved Issues in 4.1.0.0

Issue Number	Category	Description
11727	Fault Detection and Recovery	<p><b>NIC Failure on Master Host Causes Greenplum Database to Become Unresponsive</b></p> <p>In prior releases, if the primary network interface (NIC) used for interconnect traffic was disabled on the master host, the fault detection process (<code>ftsprobe</code>) 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:</p> <pre>"FATAL", "XX000", "FTS: setting segDB state failed, error connecting to entry db..."</pre> <p>This issue has been resolved in this release.</p>
11728	DML/DDL	<p><b>Slow TRUNCATE Performance on Large Objects</b></p> <p>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.</p>
11732	Upgrade	<p><b>gpupgrademirror Errors when Upgrading Greenplum Database from 3.3.x to 4.0.x</b></p> <p>When running the Greenplum Database upgrade utility (<code>gpmigrator</code>), the <code>gpupgrademirror</code> 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:</p> <pre>Exception: There was a problem with one of the gpupgrademirror sub processes.</pre> <p>This issue has been resolved in this release.</p>
11743	Fault Detection and Recovery	<p><b>Unrecoverable Segment After a System Crash</b></p> <p>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 <code>gprecoverseg</code> utility, and the <code>gpstop</code>/<code>gpstart</code> 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:</p> <pre>"PANIC", "XX000", "Append-Only Mirror Resync EOFs intent count would go negative..."</pre>
11752, 11886	Query Execution	<p><b>Information in Stack Dump and Core Dump Files Could Become Unreadable</b></p> <p>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.</p>
11773	Data Loading	<p><b>gpload Error on Tables with Added or Dropped Columns</b></p> <p>In prior releases, data loads would fail with the following error if the target table had ever been modified using <code>ALTER TABLE...ADD COLUMN</code> or <code>ALTER TABLE...DROP COLUMN</code>:</p> <pre>ERROR: syntax error at or near "--"</pre> <p>This issue has been resolved in this release.</p>

**Table A.1** Resolved Issues in 4.1.0.0

Issue Number	Category	Description
11800	Query Execution	<p><b>EXPLAIN ANALYZE Not Printing Detailed Statistics on Large Queries</b></p> <p>In prior releases, the <code>EXPLAIN ANALYZE</code> 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.</p>
11841	Transaction Management	<p><b>FATAL: the limit of xx distributed transactions has been reached</b></p> <p>In prior 4.0.x releases, if a segment failure occurred while many concurrent <code>COPY</code> or load operations were running, additional connections would sometimes fail with the following error:</p> <pre>FATAL: the limit of xx distributed transactions has been reached...</pre> <p>This issue has been resolved in this release.</p>
11843	Fault Detection and Recovery	<p><b>Primary Segment Failure Does Not Activate Mirror</b></p> <p>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:</p> <pre>PANIC", "58P01", "could not open control file ""global/pg_control"": No such file or directory...</pre> <p>When this error occurred, Greenplum Database was unable to restart. This issue has been resolved in this release.</p>
11891	Data Loading	<p><b>Unhelpful gpfdist Error Message Upon Load Failure</b></p> <p>In prior releases, when loading data using the <code>gpfdist</code> parallel file distribution server, the following error message would display upon load failure:</p> <pre>[INTERNAL ERROR gpfdist.c:1575] cannot listen all the time...</pre> <p>The <code>gpfdist</code> 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.</p>
11916	Backup and Restore	<p><b>Error Restoring Tables with Triggers</b></p> <p>In prior releases, when restoring a table from backup that had triggers associated with it, the restore would fail during <code>COPY</code> with the following error (even when the table was not an append-only or column-oriented table):</p> <pre>ERROR: AOCs Does not do trigger yet (copy.c:4007)</pre> <p>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.</p> <p><b>Note:</b> Although Greenplum Database does not fully support user-defined triggers, certain uses of triggers on heap storage tables have been verified for specific customers by Greenplum Technical Support. Triggers are disabled in Greenplum Database for append-only (AO) storage tables.</p>
11985	Backup and Restore	<p><b>Cannot Restore a Single Table Using gpdbrestore Parallel Restore Utility</b></p> <p>In prior releases, when using the parallel restore utility <code>gpdbrestore</code> with the <code>-T</code> 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.</p> <p><b>Note:</b> When using the <code>-T</code> option, <code>gpdbrestore</code> 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 <code>gpdbrestore</code>.</p>

**Table A.1** Resolved Issues in 4.1.0.0

Issue Number	Category	Description
11986	Backup and Restore	<p><b>Cannot Restore a Single Table Using gpdbrestore Parallel Restore Utility</b></p> <p>In prior releases, when using the parallel restore utility <code>gpdbrestore</code> with the <code>-T</code> 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.</p> <p><b>Note:</b> When using the <code>-T</code> option, <code>gpdbrestore</code> 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 <code>gpdbrestore</code>.</p>
11999	Workload Management	<p><b>Functions Not Evaluated Against Resource Queue Limits</b></p> <p>In prior releases, resource queues did not evaluate queries that executed functions against the limits of the resource queue. Functions containing <code>SELECT</code> statements were allowed to run, even if they exceeded the resource queue limits. This issue has been resolved in this release.</p>
12028	Workload Management	<p><b>Function <code>gp_adjust_priority()</code> Does Not Impact CPU Utilization</b></p> <p>In prior 4.0.x releases, using the <code>gp_adjust_priority()</code> 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.</p>
12103	Query Planning and Dispatch (JDBC)	<p><b>Error When Using JDBC: "Statement Error, writer gang busy..."</b></p> <p>In prior 4.0.x releases, certain <code>SELECT</code> and <code>DELETE</code> statements would fail with the following error when using the JDBC driver:</p> <pre>Statement Error, writer gang busy: possible attempt to execute volatile function in unsupported context...</pre> <p>This issue has been resolved in this release.</p>
12105	Fault Detection and Recovery	<p><b>Intermittent Network Outages Causing Segment Failures</b></p> <p>In prior 4.0.x releases, intermittent network outages could prevent connections between the master and segments, thereby causing segment failures. In this release, the master will now retry a connection if the first attempt fails, thereby decreasing the likelihood of segment failures.</p>
12124	Fault Detection and Recovery	<p><b>New Server Configuration Parameter for Fault Detection Timeout</b></p> <p>In prior releases, the interconnect timeout and the fault detection timeout were controlled by the same parameter (<code>gp_segment_connect_timeout</code>). Having too small of a fault detection timeout increased segment failures on busy systems. A new parameter has been added (<code>gp_fts_probe_timeout</code>) that allows users to control the fault detection timeout separately. The default is 3 minutes.</p>
12203	Query Execution / Append-Only Tables	<p><b>ERROR: Bad append-only storage header...</b></p> <p>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:</p> <pre>ERROR: Bad append-only storage header. Header check error 1, detail 'Append-only storage header is invalid...</pre> <p>This issue has been resolved in this release.</p>
12287	Management Utilities	<p><b>Cannot Start Greenplum Database when a Segment is Down</b></p> <p>In prior releases, the Greenplum Database startup utility, <code>gpstart</code>, 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.</p>

**Table A.1** Resolved Issues in 4.1.0.0

Issue Number	Category	Description
12339	Backup and Restore	<p><b>Partitioned Table Names Different Following a Restore</b></p> <p>In prior 4.0.x releases, the backup utilities (<code>gp_dump</code> and <code>gpcrondump</code>) 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.</p>
12346	Data Loading	<p><b>COPY Error: No partition for partitioning key...</b></p> <p>In prior releases, using the <code>COPY</code> command to load certain columns from standard input into a partitioned table would sometimes fail with the following error:</p> <pre>ERROR: no partition for partitioning key.</pre> <p>This issue has been resolved in this release.</p>

## Known Issues in Greenplum Database 4.1.0.0

This section lists the known issues in Greenplum Database 4.1.0.0. A work-around is provided where applicable.

**Table 2** Known Issues in 4.1.x

Issue	Category	Description
1589	System Catalogs and Metadata	<p><b>PostgreSQL Usage Statistics Views and Functions do not Work as Expected</b></p> <p>PostgreSQL has a number of views (<code>pg_stat_*</code>, <code>pg_statio_*</code>) 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, <code>pg_stat_get_tuples_inserted()</code> shows only those inserts into the master (usually 0), not the number inserted into the table in question.</p>
5517	DDL	<p><b>Deadlock Detected when Concurrently Altering Tables with Bitmap Indexes</b></p> <p>Concurrent <code>ALTER TABLE</code> commands on tables that have bitmap indexes may encounter a deadlock and be unable to proceed. This issue occurs with concurrent <code>ALTER TABLE</code> commands that set distribution policies on tables that have bitmap indexes.</p> <p><b>Work-around:</b> if you need to run concurrent <code>ALTER TABLE</code> commands on tables with bitmap indexes (as when running <code>gpexpand</code> with <code>-n</code> for multiple parallel processes), first drop the bitmap indexes and reinstate them when the concurrent <code>ALTER TABLE</code> commands are completed.</p>
5517, 3213	Query Execution	<p><b>Transaction Within a Function Not Recognized as a Sub-Transaction</b></p> <p>When a function containing multiple transaction blocks is run and an error occurs in one transaction block, the entire function exits with the errors:</p> <pre>ERROR: The distributed transaction 'Prepare' broadcast failed to one or more segments ERROR: current transaction is aborted, commands ignored until end of transaction block</pre>

**Table 2** Known Issues in 4.1.x

Issue	Category	Description
5647	Upgrade	<p><b>Upgrade Utility Requires Standard Prefixes for Segment Data Directories</b></p> <p>The <code>gpmigrator</code> 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 <code>gpmigrator</code>, and cannot be upgraded using this utility.</p> <p>The naming convention for data directories in a Greenplum Database system is <code>SEG_PREFIXnumber</code> where number starts with 0 for segment instances (the master is always -1). So for example, if you choose the prefix <code>gp</code>, your master instance data directory would be named <code>gp-1</code>, and the segment instances would be named <code>gp0</code>, <code>gp1</code>, <code>gp2</code>, and so on.</p>
6279	DDL	<p><b>Append-Only Table Error After ALTER TABLE...DROP COLUMN</b></p> <p>After altering an append-only table to drop a column, users encounter the following error when trying to access the table:</p> <pre>ERROR: cache lookup failed for type 0</pre> <p><b>Work-around:</b> If you need to drop a column from an append-only table, do not use <code>ALTER TABLE</code>. Instead recreate the append-only table with the desired columns using <code>CREATE TABLE ...AS SELECT</code>.</p>
8445	DDL	<p><b>CREATE DATABASE ERROR: "template1" is being accessed by other users</b></p> <p>When the system has segments in <code>resynchronizing</code> mode, administrators will not be able to create a new database. Running a <code>CREATE DATABASE</code> command when segments are in the process of resynchronizing will result in the following error:</p> <pre>ERROR: source database "template1" is being accessed by other users...</pre> <p><b>Work-around:</b> Use <code>gpstate -e</code> to check the status. When all segments are synchronized, you may retry the <code>CREATE DATABASE</code> command.</p>
9968	Backup and Restore	<p><b>Slow gp_dump / gpccrondump Performance</b></p> <p>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.</p> <p><b>Work-around:</b> Run <code>VACUUM ANALYZE</code> 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:</p> <pre>ALTER ROLE gpadmin SET enable_nestloop = on; ALTER ROLE gpadmin SET random_page_cost = 10;</pre> <p>After the dump is complete, return to the default settings:</p> <pre>ALTER ROLE gpadmin RESET enable_nestloop; ALTER ROLE gpadmin RESET random_page_cost;</pre>

**Table 2** Known Issues in 4.1.x

Issue	Category	Description
10028	Fault Detection and Recovery	<p><b>After Running gprecoverseg, Segments May Not Be Running in Their Preferred Role</b></p> <p>In 4.x releases, when a primary segment goes down, the mirror is activated and becomes the primary segment. After running <code>gprecoverseg</code>, 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.</p> <p>After recovering a failed segment, the roles may be reversed (the original mirror is now the primary and vice-versa). The <code>gpstate -e</code> command will show you if you have segments in this condition.</p> <p><b>Work-around:</b> Run a segment rebalance operation (<code>gprecoverseg -r</code>) following segment recovery. Any active queries will be rolled back, however client sessions will remain connected.</p>
10278	System Catalogs and Metadata	<p><b>pg_relation_size and pg_database_size do not Account for Custom Filespaces</b></p> <p>The <code>pg_relation_size</code> and <code>pg_database_size</code> 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 tablespaces besides the default <code>pg_system</code> tablespace.</p>
11668	Query Execution / Performance	<p><b>Append Only Table Performance Issues on Linux Platforms</b></p> <p>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.</p> <p><b>Work-around:</b> 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:</p> <pre>rw,noatime,inode64,allocsize=16m</pre>
11690	Fault Detection and Recovery	<p><b>Greenplum Database Self-Healing Process Can Create Zero Byte Files</b></p> <p>When Greenplum Database automatically recovers from errors and other system problems (self-healing), the following error message might display when accessing the database:</p> <pre>"Could not read from file ""pg_clog/0004"</pre> <p>This occurs because, in rare cases, zero byte files are created during the self-healing process.</p> <p><b>Work-around:</b> Contact EMC Technical Support.</p>
11734	Memory Quota	<p>When scanning a highly partitioned table with low reserved memory that was set by memory quota, an insufficient memory reserved error can appear.</p> <p><b>Work-around:</b> Increase the reserved memory increasing <code>statement_mem</code> or increasing the limits of a resource queue. Use <code>statement_mem</code> to set a memory limit for a query. When <code>gp_resqueue_memory_policy</code> is enabled, <code>statement_mem</code> replaces <code>work_mem</code>. The default is 125MB.</p>

**Table 2** Known Issues in 4.1.x

Issue	Category	Description
12050	DDL	<p><b>gp_toolkit Schema Has No Objects After Creating a New Database</b></p> <p>When a new database instance is created, the <code>gp_toolkit</code> schema is also created in that database. Within this schema are objects for Greenplum system administration. On rare occasions, creating a new database creates a <code>gp_toolkit</code> schema with no objects.</p> <p><b>Work-around:</b> Execute the following command:</p> <pre>psql -f \$GPHOME/share/postgresql/gp_toolkit.sql database_name</pre> <p>This populates the <code>gp_toolkit</code> schema with the required objects.</p>
12082	Query Execution	<p><b>Error When GROUP BY and a SELECT Clauses Refer to a Column in Different Ways</b></p> <p>Greenplum Database interprets queries that mix windowing (<code>OVER</code> clause) and aggregation (<code>GROUP BY</code> 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 <code>FROM</code> 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 <code>GROUP BY</code> clause refers to a column in one way and a <code>SELECT</code> clause refers to it in a different way.</p> <p><b>Work-around:</b> Supply an alias for the join result (either in ANSI JOIN syntax in the <code>FROM</code> clause or by expressing the JOIN as a subquery in the <code>FROM</code> clause), and refer to columns by this alias for clauses outside the <code>FROM</code> and <code>WHERE</code> clauses of the query.</p>
12099	DDL / Partitioned Tables	<p><b>Using ALTER TABLE...DROP COLUMN to Alter the Schema of a Partitioned Table Causes Errors</b></p> <p>Currently, adding additional partitions to the hierarchy after altering the table design by using <code>ALTER TABLE...DROP COLUMN</code> can cause those new partitions to become corrupted. In this scenario, the following error message might appear:</p> <pre>ERROR: distribution policy for "new_partition" must be the same as that for "parent_partition"</pre> <p><b>Work-around:</b> If you need to drop columns on a partitioned table and want to avoid this issue, run <code>ALTER TABLE...DROP COLUMN, CTAS the partitioned table, DROP the old table, and rename the new table to the old table name.</code></p>
12120	Installation	<p><b>Greenplum Database directories show the incorrect group for the owner user if the owner name and the group name are different.</b></p> <p>The installation package assumes the user name and group are the same; that is, <code>gpadmin</code> is part of the group named <code>gpadmin</code>. If this is not the case, the group is incorrect.</p> <p><b>Work-around:</b> Change the group ownership manually by running <code>chgrp</code>.</p>
12165, 12173	Query Execution	<p><b>Queries with Multi-row Subqueries Are Not Supported</b></p> <p>Currently, Greenplum Database does not support queries that contain multi-row subqueries.</p> <p><b>Work-around:</b> Rewrite to query statement to avoid using multi-row subqueries.</p>
12190	Query Execution	<p><b>ERROR: Type 19258 is not hashable</b></p> <p>Certain data types such as <code>ip4r</code> are not fully supported in Greenplum Database because the query planner does not know how to hash them for distributed query operations. Joining on columns of such data types will give an error such as:</p> <pre>ERROR: Type 19258 is not hashable</pre> <p><b>Work-around:</b> Convert columns using unsupported data types to data types supported by Greenplum Database.</p>



**Table 2** Known Issues in 4.1.x

Issue	Category	Description
12192	Query Execution	<p><b>Sub-optimal Performance when Joining on Columns of Different Data Types</b></p> <p>When you join on columns of different datatypes, the query planner implicitly casts the columns to a common datatype before it can execute the join. In some cases, this implicit casting causes the query planner to choose a sub-optimal query plan, thereby increasing query execution time.</p> <p><b>Work-around:</b> Columns that are joined frequently should be of the same datatype. If this is not possible, explicitly cast joined columns to the same datatype as part of your SQL statement.</p>
12233	Query Execution / Partitioning	<p><b>DEFAULT Partition Always Scanned Regardless of Query Predicate</b></p> <p>The default partition of a partitioned table is always scanned during query execution, even when the query predicate has explicit conditions that match other partitioned tables. This may impact query performance when the default partition contains significant amounts of data.</p> <p><b>Work-around:</b> After loading data into a partitioned table, make sure that any data landing in the default partition is split out into a new partition instead of remaining in the default partition.</p>
12250	Data Loading	<p><b>Connections to Greenplum Database Remain Open After an ETL Operation</b></p> <p>On rare occasions while performing an ETL operation using a third-party ETL tool, the following error might display:</p> <pre>"53300","sorry, too many clients already.,""There are no more available slots in the sharedSnapshotArray..."</pre> <p>This indicates that there are no more available client connection slots to the database. This issue is caused by previous ETL operations not properly closing their connections to Greenplum Database.</p> <p><b>Work-around:</b> Manually close the open connections.</p>
12288	System Catalogs and Metadata	<p><b>gpcheckcat Error: gp_persistent_* state check found</b></p> <p>The <code>gpcheckcat</code> utility is used to detect potential inconsistencies in the system catalog tables of the master and segments. In certain rare circumstances, an inconsistency to the <code>gp_persistent_*</code> catalog tables may be introduced if a segment experiences a system outage (such as a power failure) during certain <code>CREATE</code> or <code>DROP</code> operations. When this occurs, the <code>gpcheckcat</code> utility will detect and report errors.</p> <pre>[ERROR]:-gp_persistent_tablespace_node state check found</pre> <p><b>Work-around:</b> Contact EMC Technical Support.</p>
12334	Fault Detection and Recovery	<p><b>Shared Memory Error Can Cause Some Segment Mirrors to Become Unresponsive</b></p> <p>In very rare circumstances, an issue with shared memory in the Greenplum Database can result in the following error:</p> <pre>"FATAL","XX000","semop(id=21495876,num=13) failed: Numerical result out of range (pg_sema.c:499)"</pre> <p>When this error occurs, mirror segments can become unresponsive, and the system goes into change-tracking mode.</p> <p><b>Work-around:</b> To resolve this issue, run the <code>gprecoverseg</code> utility.</p>
12472	Memory	<p><b>No Available Memory</b></p> <p>In very rare circumstances, the Greenplum Performance Monitor process <code>gpmonws</code> can use all available memory on the master node.</p> <p><b>Work-around:</b> Restart the Greenplum Performance Monitor.</p>

**Table 2** Known Issues in 4.1.x

Issue	Category	Description
12547	Unsuccessful create index, reindex, or vacuum	<b>Inconsistent Results after unsuccessful create index, reindex, or vacuum with a bitmap index</b> If the system crashes when running an index, reindex, or vacuum command with a bitmap index you might see inconsistent results in the data. <b>Work-around:</b> Run the index, reindex, or vacuum command again.
12577	Expanding Partitioned Tables in Multiple Schemas	In systems with partitioned tables with the same name in multiple schemas, <code>gpexpand</code> issues multiple commands to redistribute the data for each partitioned table. This causes the expansion process to become very slow, as each table is expanded multiple times. For example, in the following setup <pre> schema_one.part_table_1 schema_two.part_table_1 schema_three.part_table_1 </pre> running <code>gpexpand</code> issues a total of nine <code>ALTER TABLE SET DISTRIBUTED BY</code> statements. <b>Work-around:</b> Contact EMC Technical Support if your schema meets these criteria and you are planning an expansion.
12590	Email Alerts	<b>Invalid SMTP Server for Email Alerts Causes Degraded Performance</b> When you configure the Greenplum Database to send email alerts, you must supply a value for the parameter <code>gp_email_smtp_server</code> . If the specified email server is invalid or unreachable from the Greenplum master host, then Greenplum Database can experience degraded performance or become unresponsive. <b>Work-around:</b> If you notice degraded performance after configuring email alerts, either disable the email alerting feature, or make sure that the specified email server is valid and accessible from the Greenplum master host.

## Upgrading to Greenplum Database 4.1

For detailed upgrade procedures, see the following sections:

- [Upgrade Checklist](#)
- [Upgrading From 3.3.x.x to 4.1.0.0](#)

### Upgrading from 4.0.x.x to 4.1.0.0

For detailed upgrade procedures, see the following sections:

- [Upgrade Procedure](#)

#### Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.0.x.x to 4.1.0.0. Detailed upgrade instructions are also provided in the [Upgrade Procedure](#) section.

Pre-Upgrade Preparation (on your current system)	
* 4.0.x.x system is up and available	
<input type="checkbox"/>	Log in to your master host as the <code>gpadmin</code> user (your Greenplum superuser).
<input type="checkbox"/>	Check for and recover any failed segments ( <code>gpstate</code> , <code>gprecoverseg</code> ).

<input type="checkbox"/>	Install the Greenplum Database 4.1.0.0 binaries on all Greenplum hosts.
<input type="checkbox"/>	Copy any custom modules from your existing installation to your 4.1 installation on all Greenplum hosts. For example, shared library files for user-defined functions in <code>\$GPHOME/lib</code> or PostgreSQL add-on modules (such as <code>plr.so</code> or <code>pgcrypto.so</code> ) in <code>\$GPHOME/lib/postgresql</code> .
<input type="checkbox"/>	Copy or preserve any additional folders or files (such as backup folders) that you have added in the Greenplum data directories or <code>\$GPHOME</code> directory. Only files or folders strictly related to Greenplum Database operations are preserved by the migration utility.
<input type="checkbox"/>	(Optional) Run <code>VACUUM</code> on all databases, and remove old server log files from <code>pg_log</code> in your master and segment data directories. This is not required, but will reduce the size of Greenplum Database files to be backed up and migrated.
<input type="checkbox"/>	Inform all database users of the upgrade and lockout time frame. From this point onward, users should not be allowed on the system until the upgrade is complete.
<b>Upgrade Execution</b>	
<i>* The system will be locked down to all user activity during the upgrade process</i>	
<input type="checkbox"/>	Backup your current databases ( <code>gpcrondump</code> or ZFS snapshots) and secure backup files in a location outside of your Greenplum data directories.
<input type="checkbox"/>	Remove the standby master from your system configuration ( <code>gpinitstandby -r</code> ).
<input type="checkbox"/>	Do a clean shutdown of your current system ( <code>gpstop</code> ).
<input type="checkbox"/>	Update your environment to source your Greenplum Database 4.1 installation.
<input type="checkbox"/>	Run the upgrade utility ( <code>gpmigrator_mirror</code> if you have mirrors, <code>gpmigrator</code> if you do not).
<input type="checkbox"/>	After the upgrade process finishes successfully, your 4.1 system will be up and running.
<b>Post-Upgrade (on your 4.1 system)</b>	
<i>* The 4.1 system is up</i>	
<input type="checkbox"/>	Reinitialize your standby master host ( <code>gpinitstandby</code> ).
<input type="checkbox"/>	Upgrade <code>gpfdist</code> on all of your ETL hosts by installing the version 4.1 Load Tools package.

- |                          |  |
|--------------------------|--|
| <input type="checkbox"/> | Install the new 4.1 Performance Monitor Web Application and update your environment to point to the 4.1 Performance Monitor binaries.                        |
| <input type="checkbox"/> | Inform all database users of the completed upgrade. Tell users to update their environment to source the Greenplum Database 4.1 installation (if necessary). |

### Upgrade Procedure

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



**Important:** Carefully evaluate each section and perform all required and conditional steps. Failing to perform any of these steps can result in an aborted upgrade, placing your system in an unusable or even unrecoverable state.

#### Pre-Upgrade Preparation (on your 4.0.x system)

Perform these steps on your current 4.0.x Greenplum Database system. This procedure is performed from your Greenplum master host and should be executed by the Greenplum superuser (`gpadmin`).

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

```
$ su - gpadmin
```
2. (optional) Vacuum all databases prior to upgrade. For example:
 

```
$ vacuumdb database_name
```
3. (optional) Clean out old server log files from your master and segment data directories. For example, to remove all existing log files from your segment hosts:
 

```
$ gpssh -f seg_host_file -e 'rm /gpdata/*/gp*/pg_log/*.csv'
```
4. (optional) Run the `gpcheckcat` utility to validate your system catalogs. For example:
 

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

If any errors are reported, contact Greenplum Customer Support.
5. Run `gpstate` to check for failed segments.
 

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

```
$ gprecoverseg
```

**Note:** It might be necessary to restart the database if the preferred role does not match the current role; for example, if a primary segment is acting as a mirror segment or a mirror segment is acting as a primary segment.

#### Install the Greenplum software binaries

1. Download or copy the installer file to the Greenplum Database master host.
2. Unzip the installer file. For example:

```
# unzip greenplum-db-4.1.x.x-PLATFORM.zip
```

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

```
# /bin/bash greenplum-db-4.1.x.x-PLATFORM.bin
```
4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press `ENTER` to accept the default install path (`/usr/local/greenplum-db-4.1.0.0`), or enter an absolute path to an install location. You must have write permissions to the location you specify.
6. The installer will install the Greenplum software and create a `greenplum-db` symbolic link one directory level above your version-specific Greenplum installation directory. The symbolic link is used to facilitate patch maintenance and upgrades between versions. The installed location is referred to as `$GPHOME`.
7. Create a `hostfile` file that has the machine configured host names and host addresses (interface names) for each host in your Greenplum system (master, standby master and segments). Make sure there are no blank lines or extra spaces. For example, if you have a master, standby master and three segments with two network interfaces per host, your file would look something like this:

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

8. Source the path file from your new 4.1.0.0 installation. For example:
 

```
$ source /usr/local/greenplum-db-4.1.0.0/greenplum_path.sh
```
9. Run the `gpsegininstall` utility referencing the `hostfile` file you just created. Use the `-u` and `-p` options to specify the name and password of your Greenplum administrative user (`gpadmin`). For example:
 

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

- 10.** Copy any custom modules from your existing 4.0 installation to your 4.1 installation on all Greenplum hosts. For example, shared library files for user-defined functions in `$GPHOME/lib` or PostgreSQL add-on modules (such as `plr.so` or `pgcrypto.so`) in `$GPHOME/lib/postgresql`.

#### Upgrade Execution

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

- 11.** Source the path file from your old 4.0.x.x installation. For example:
- ```
$ source /usr/local/greenplum-db-4.0.4.0/greenplum_path.sh
```
- 12.** (*optional but strongly recommended*) Back up all databases in your Greenplum Database system using `gpcrondump` (or `zfs` snapshots on Solaris systems). See the *Greenplum Database Administrator Guide* for more information on how to do backups using `gpcrondump`. Make sure to secure your backup files in a location outside of your Greenplum data directories.
- 13.** If your system has a standby master host configured, remove the standby master from your system configuration. For example:
- ```
$ gpinitstandby -r
```
- 14.** Perform a clean shutdown of your current Greenplum system. For example:
- ```
$ gpstop
```
- 15.** Source the path file from your new 4.1.0.0 installation. For example:
- ```
$ source /usr/home/greenplum-db-4.1.0.0/greenplum_path.sh
```
- 16.** Update your environment so that it is sourcing your new 4.1 installation.
- For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.1 installation directory. For example (as root):
 

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.1.0.0
  /usr/local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```
  - Using `gpssh`, also update the `greenplum-db` symbolic link on all of your segment hosts. For example (as root):
 

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.1.0.0
  /usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
=> exit
```
- 17.** As `gpadmin`, run the 4.1 version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:
- ```
$ su - gpadmin
```

```
$ gp migrator_mirror /usr/local/greenplum-db-4.0.4.0
/usr/local/greenplum-db-4.1.0.0
```

- 18.** The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.1 system will be running and accepting connections.

Note: After the migration utility has completed, the resynchronization of the mirror segments with the primary segments continues. Even though the system is running, the mirrors are not active until the resynchronization is complete.

#### Post-Upgrade (on your 4.1.0.0 system)

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

```
$ gpinitstandby -s standby_hostname
```

- 20.** If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the Greenplum Database 4.1 Load Tools package. Packages are available on <http://gpn.greenplum.com>.

- 21.** If you are using the Greenplum Performance Monitor, install the new 4.1 Performance Monitor Web Application and update your environment to point to the new 4.1 Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.1 installation). Greenplum Performance Monitor 4.1 Web Application Packages are available on <http://gpn.greenplum.com>.

- 22.** See “[Server Configuration Parameters](#)” on page 6 for parameter changes introduced in 4.1. Check your `postgresql.conf` server configuration files for any settings that should be removed or changed. You can use the `gpconfig` utility to check and set parameters.

#### Troubleshooting a Failed Upgrade

If you experience issues during the migration process, 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](mailto:entitlement@greenplum.com).

Be prepared to provide the following information:

- A completed [Upgrade Checklist](#).
- Log output from `gp migrator` and `gpcheckcat` (located in `~/gpAdminLogs`)

#### Upgrading From 3.3.x.x to 4.1.0.0

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.1.0.0. You must first upgrade from your current release to 4.0.x.x (follow the upgrade instructions in the latest Greenplum Database 4.0.x.x release notes available on <http://gpn.greenplum.com>). After you have successfully upgraded to 4.0, follow the upgrade instructions for [Upgrade Checklist](#).

## Greenplum Database 4.1 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 4.1 client tool packages available, and their version compatibility with Greenplum Database 4.1 server releases. Client tool packages are available for download from [Greenplum Network](#).

**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)<br>Greenplum MapReduce (gpmapreduce) <sup>1</sup> | 4.1.0.0         | 4.1.0.0         |
| Greenplum Connectivity | Standard PostgreSQL Database Drivers (ODBC, JDBC)<br>PostgreSQL Client C API (libpq)               | 4.1.0.0         | 4.1.0.0         |
| Greenplum Loaders      | Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)                                   | 4.1.0.0         | 4.1.0.0         |
| Greenplum Loaders      | Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)                                   | 4.1.0.0         | 4.1.0.0         |

1. gpmapreduce is not available on Windows.

## Greenplum Database 4.1 Documentation

**Table 2** Greenplum Database Documentation

| Title                                                          | Part Number | Revision |
|----------------------------------------------------------------|-------------|----------|
| Greenplum Database 4.1 Installation Guide                      | 300-012-429 | A01      |
| Greenplum Database 4.1 Administrator Guide                     | 300-012-428 | A01      |
| Greenplum Database Performance Monitor 4.1 Administrator Guide | 300-012-430 | A01      |
| Greenplum Database 4.1 Client Tools for UNIX                   | 300-012-432 | A01      |
| Greenplum Database 4.1 Client Tools for Windows                | 300-012-433 | A01      |
| Greenplum Database 4.1 Connectivity Tools for UNIX             | 300-012-434 | A01      |
| Greenplum Database 4.1 Connectivity Tools for Windows          | 300-012-435 | A01      |
| Greenplum Database 4.1 Load Tools for UNIX                     | 300-012-436 | A01      |
| Greenplum Database 4.1 Load Tools for Windows                  | 300-012-437 | A01      |

**Copyright © 2011 EMC Corporation. All rights reserved.**

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.



THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on [EMC.com](http://EMC.com)

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