
Welcome to Greenplum Database 4.1.2.0

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.

About Greenplum Database 4.1.2

Greenplum Database 4.1.2 is the second service pack release for version 4.1.2. This service pack release includes new issue resolutions and minor feature enhancements. Please refer to the following sections for more information about this release.

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

- [New Functionality in Greenplum Database 4.1.2](#)
- [Resolved Issues in Greenplum Database 4.1.2](#)
- [Known Issues in Greenplum Database 4.1.2](#)
- [Upgrading to Greenplum Database 4.1.2](#)
- [Greenplum Database Client Tool Packages and Performance Monitor](#)
- [Greenplum Database 4.1.2 Documentation](#)

Release Notes for previous 4.1.1.x and earlier releases are available on [Powerlink](#).

New Functionality in Greenplum Database 4.1.2

In previous versions of the Greenplum Loaders, `gpfdist` could not be run as a Windows service. Starting with Greenplum Loaders version 4.1.1.3, `gpfdist` can be run as a Windows service.

See “[Greenplum Database Client Tool Packages and Performance Monitor](#)” on page 23 for detailed compatibility information for all the Greenplum client tool packages.

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

1. Update your Greenplum Loader package to version 4.1.1.3 or higher. Greenplum Loader packages are available from the [EMC Download Center](#).
2. Register `gpfdist` as a Windows service:
 - a. Open a Windows command window

b. Run the following command:

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

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

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

3. Activate the `gpfdist` service:**a.** Open the Windows Control Panel and select **Administrative Tools>Services**.**b.** Highlight then right-click on the `gpfdist` service in the list of services.**c.** Select **Properties** from the right-click menu, the **Service Properties** window opens.

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

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

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

Resolved Issues in Greenplum Database 4.1.2

The following table lists the customer reported issues that are now resolved in Greenplum Database 4.1.2.

For issues resolved in Greenplum Database 4.1.1 and earlier, see the corresponding release notes available on n [Powerlink](#).

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
8635/6859	4.1.2	Access Methods	Bitmap index returned incorrect results If an update or an insert operation was performed following a vacuum full operation on a table with a bitmap index, the bitmap index returned incorrect results. This issue has been resolved in this release.
15620	4.1.2	emconnect	Header file incorrectly referenced On platforms where emconnect is enabled (redhat) there was an issue compiling certain contrib modules due to an inappropriate inclusion of emconnect/api.h: a header file we do not ship. This issue has been resolved in this release.
14697	4.1.2	Upgrade/Down grade	gp migrator failed in non-mirrored upgrades gp migrator failed if a catalog index had been reindexed. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
13963	4.1.2	Access Methods	Error: int64 out of range Insert into an append only table from an external table failed with the above error. This issue has been resolved in this release.
13818	4.1.2	Backup Restore	gp_dump failed during backup gpcrondump (and the underlying gp_dump) failed when excluding (using the <code>-T</code> option) tables that contain sequences. This issue has been resolved in this release.
13750	4.1.2	DDL/DML: Partitioning	Adding a partition failed when partial index exists Adding a partition to a table failed when there was already a partial index on the partitioned table. This issue has been resolved in this release.
13590	4.1.2	Replication: Crash recovery	Concurrent DDL caused panic errors In rare circumstances concurrent DDLs caused "PANIC", "XX000", "Waiting on lock already held!" errors. This issue has been resolved in this release.
13576	4.1.2	Query Execution	Running a query with concurrency caused database lockdown In rare circumstances, concurrently run queries encountered an internal deadlock. This issue has been resolved in this release.
13414	4.1.2	Query Execution	Error: Out of memory A memory leak occurred in the executor when the optimizer chose to use an index scan satisfying predicates on append only tables in a query. This issue has been resolved in this release.
13288	4.1.2	Interconnect	Excessive logging Excessive logging on segments filled up memory, affecting performance. This issue has been resolved in this release.
13237	4.1.2	Upgrade / Downgrade	Mirror segments were down after upgrade Following an upgrade to 4.0.5.3 the mirror segments were down. This issue has been resolved in this release.
13214	4.1.2	SQL Parser	Difference in behavior of to_date WITH to_char built-in functions In some releases of Greenplum Database the <code>to_date</code> and <code>to_char</code> functions didn't process leading spaces the same way. This issue has been resolved in this release.
12849	4.1.2	gpperfmon	Performance Monitor's python process used large amounts of memory gpperfmon's python process was taking up to 9GB of memory causing errors with other python-based management utilities. This issue has been resolved in this release.
12847	4.1.2	gmapreduce	Problems were occurring when running gmapreduce with functions pre-defined in the database Problems were encountered when gmapreduce attempted to call a pre-defined function in the database. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
12839	4.1.2	Loaders: gpfdist	gpfdist error: Segment reject limit reached Missing new line character in data file caused segment reject limit reached error. This issue has been resolved in this release.
12791	4.1.2	Fault Detection and Recovery	Erroneous segment failure message was being issued. In some circumstances the following failure message was erroneously being issued: "Greenplum Database detected segment failure(s), system is reconnected". This issue has been resolved in this release.
12773	4.1.2	Query Execution	Windows function caused error: unexpected internal error Executing more than one windows function was resulting in the above error. This issue has been resolved in this release.
12767	4.1.2	gmapreduce	gmapreduce returned incorrect error code When gmapreduce encountered an error, it did not always return a non-zero error code, as expected. This issue has been resolved in this release.
12755	4.1.2	Fault Detection and Recovery	4.0.4.0 upgrade to 4.1 failed Upgrades from Greenplum Database 4.0.4.0 to version 4.1, using gp migrator utility, without mirrors, failed. This issue has been resolved in this release.
12599	4.1.2	Upgrade	Upgrade failed due to pg_amop inconsistency issue When attempting to Greenplum version 4.1, a pg_amop inconsistency issue from gpcheckcat and system caused the upgrade to fail. This issue has been resolved in this release.
12540	4.1.2	Replication: Segment Mirroring	Prepare transaction caused panic errors In some circumstances, a prepare transaction operation on a segment resulted in segment panic errors, causing the database to crash. This issue has been resolved in this release.
12518	4.1.2	Query Optimizer	Query errors After running Analyze, some queries were returning the following internal error: "variable not found in subplan target lists" This issue has been resolved in this release.
12502	4.1.2	Loaders: gpload	gpload 'UPDATE_CONDITION' restriction In the gpload configuration file, the MATCH_COLUMNS could formerly only contain Greenplum distribution key columns for the target table. This has been improved so that non-distribution key columns can be specified in MATCH_COLUMNS. This issue has been resolved in this release.
12048	4.1.2	Query Optimizer	EXPLAIN ANALYZE failed causing internal error Explain Analyze statement was failing on specific tables with the following message ERROR: Unexpected internal error This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
11865	4.1.2	Backup and Restore	<p>Tables with indexes slow to restore</p> <p>The restore operation on a table with an index was taking place after the index was created, which slowed the restore process. The index is now created after restoring the data. The <code>CREATE INDEX</code> DDL is saved in a separate <code>post_data</code> backup file.</p> <p>This issue has been resolved in this release.</p>
11827	4.1.2	Loaders: gpload	<p>data loading using gpload impacted performance</p> <p>The constant create and drop of external tables that happens when using <code>gpload</code> caused catalog bloat on systems that perform micro batch loads, since these operations happen very frequently (in some cases 10,000's of times a day). In such cases, the result was slowing of the system and performance issues.</p> <p>This issue has been resolved in this release.</p>
9495	4.1.2	Loaders: gpfdist	<p>gpfdist on windows with wildcard in external table failed</p> <p>When running <code>gpfdist</code> on windows, combined with an external table that contained a wildcard, returned a 404 error.</p> <p>This issue has been resolved in this release.</p>
14219	4.1.2	Documentation	<p>Incorrect default batch size for gpexpand in documentation</p> <p>The default batch size (-B option) for the <code>gpexpand</code> utility is incorrectly documented in the help file and the Greenplum Administrator Guide as 8. The actual default batch size is 16.</p> <p>This issue has been resolved in this release.</p>
13596	4.1.2	Catalog and Metadata	<p>pg_type corruption</p> <p>Some queries were causing corruption of the <code>pg_type</code> system catalog table that resulted in the database being unusable.</p> <p>This issue has been resolved in this release.</p>
13025	4.1.2	Upgrade	<p>Upgrade to version 4.1 failed due to gp_toolkit dependencies</p> <p>In cases where there were external tables created that depend on <code>gp_toolkit</code>, upgrades to Greenplum Database failed.</p> <p>This issue has been resolved in this release.</p>
12907	4.1.2	Interconnect	<p>Intermittent glibc double free corruption detected</p> <p>In some circumstances a loss of connectivity to GPDIST while writing data to it was resulting in intermittent glibc double free corruption.</p> <p>This issue has been resolved in this release.</p>
12843	4.1.2	Upgrade / Downgrade	<p>Upgrade from 3.3 to 4.0.4 failed</p> <p>If, for any reason, migration failed during the phase of migration where the 4.0 version of the catalog is copied from the primary to the mirror, a restart of the migration (restart using the <code>-resume</code> option) sometimes failed.</p> <p>This issue has been resolved in this release.</p>
12775	4.1.2	Partitioning	<p>Partition column error</p> <p>Relations that did not have same number of columns as each other resulted in an error being issued.</p> <p>This issue has been resolved in this release.</p>

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
12738	4.1.2	Upgrade	Upgrade failure Under rare circumstances, upgrades from 4.0.4 to 4.1 failed with unexpected internal errors. This issue has been resolved in this release.
12523	4.1.2	Replication / Crash Recovery	Crash recovery inconsistencies Dropped objects may not be removed from gp_persistent_relation_node. This issue has been resolved in this release.
12472	4.1.2	Memory	No Available Memory In very rare circumstances, the Greenplum Performance Monitor process gpmonws can use all available memory on the master node. This issue has been resolved in this release.
12341	4.1.2	Documentation	Documentation of gprecoverseg was unclear The instructions for running gprecoverseg did not explicitly state that it should be run on the master. This issue has been resolved in this release.
12334, 12604	4.1.2	Fault Detection and Recovery	Shared Memory Error Can Cause Some Segment Mirrors to Become Unresponsive In very rare circumstances, an issue with shared memory in the Greenplum Database can result in the following error: "FATAL", "XX000", "semop(id=21495876,num=13) failed: Numerical result out of range (pg_sema.c:499)" When this error occurs, mirror segments can become unresponsive, and the system goes into change-tracking mode. This issue has been resolved in this release.
12051	4.1.2	Partitioning	COPY statement failed A COPY statement with certain positioning of attributes in a "COPY (attributes...) FROM STDIN" failed with the following error message: "No partition for partitioning key" This issue has been resolved in this release.
7097	4.1.2	Upgrade / Downgrade	ERROR: catalog is missing 4 attribute(s) received during gpmigration During migration from 3.2.x or 3.3.x to 4.0.x, in cases of systems with very large numbers of tables, the extension on the catalog table was not being picked up, resulting in the above error being issued. This issue has been resolved in this release.
6830	4.1.2	DDL and Utility Statements	Ctrl-C command following a vacuum full terminates the database This issue has been resolved in this release.
15439	4.1.1.8	Catalog and Metadata	gpcheckcat foreign key errors Dropping the grantor role caused gpcheckcat to report foreign key errors. This issue has been resolved in this release.
15292	4.1.1.8	Catalog and Metadata	gpcheckcat slow performance In a large scale environment gpcheckcat in some cases was taking a long time to complete. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
14396	4.1.1.8	Query Execution	Error: DELETE fails with "PANIC", "XX000", "Waiting on lock already held!" Updates and deletes resulted in a segment restart and transactions being aborted. This issue has been resolved in this release.
13624	4.1.1.8	Replication/ Crash Recovery	Error: Shutdown fails with "PANIC", "XX000", "Waiting on lock already held!" error A forced shutdown during concurrent activity on the system resulted in the above error and the master was unable to restart. This issue has been resolved in this release.
13420	4.1.1.8	Monitoring: Performance Monitor	Performance Monitor performance issues Performance Monitor's python process was using large amount of memory which was causing other python-based management utilities to throw various errors. This issue has been resolved in this release.
12614	4.1.1.7	Replication/ Crash Recovery	Error: 'Cannot read status of transaction from pg_clog file' In some circumstances, the database could not restart, because the transaction status was not available. This issue has been resolved in this release.
14283	4.1.1.6	Loaders	Error: 'Address already in use' In some circumstances when multiple gpload sessions were started at the same time using the same <code>PORT_RANGE</code> , some sessions might fail with "Address already in use" error. This issue has been resolved in this release.
14667	4.1.1.6	Loaders	gpload: not-null constraint violations gpload was throwing not-null constraint violations when using reusable temporary tables. This issue has been resolved in this release.
14519	4.1.1.6	Functions and Languages	Error: 'relation with OID xxxx does not exist' The above error was generated in cases where tables were being created and dropped frequently. This was a result of an issue with earlier versions of Postgres. This issue has been resolved in this release.
14125	4.1.1.5	Query Execution	Query with AGGREGATE functions failed Some queries that called <code>AGGREGATE</code> functions failed with a 'Segment process received signal SIGSEGV (postgres.c:3360)' error. This issue has been resolved in this release.
14037	4.1.1.5	Backup/ Restore	Slow multiple-table restore from backup When specifying multiple tables to be restored from a preexisting dump file, a portion of the restore operation was being repeated unnecessarily, generally resulting in longer completion times. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
14011	4.1.1.5	Resource Management	Resource queue initialization error A rare condition in resource management initialization may have lead to the Greenplum Database being unable to start. When this occurred, the system failed with the message "insufficient resource queues available". This issue has been resolved in this release.
13999/ 13973	4.1.1.5	DDL/DML: Heap	Insufficient memory error Under rare circumstances, updates involving very large numbers of records can result in an 'insufficient memory' error. This issue has been resolved in this release.
13969	4.1.1.5	DML	Window function returning incorrect results A window function that specifies <code>RANGE BETWEEN interval '0 seconds' FOLLOWING</code> returns incorrect results. This issue is resolved with the following workaround: Use <code>'0 seconds' PRECEDING</code> to specify the interval.
13845	4.1.1.5	Query Optimizer	SQLPrepare did not throw exceptions for unsupported query constructs The ODBC function, SQLPrepare, used to test the validity of SQL statements prior to sending the statement to GPDB for execution, incorrect indicated that unsupported query constructs were supported. This issue has been resolved in this release.
11911	4.1.1.5	Monitoring: Performance Monitor	Performance Monitor historical metrics not displaying correctly In the Performance Monitor user interface, system metrics graphs covering durations of 1hr (the default) or 6hrs did not display properly. This issue has been resolved in this release.
14145	4.1.1.5	Management Tools	gpexpand failed due to a preexisting, inconsistent gpexpand schema The existence of an inconsistent 'gpexpand' schema caused a failure in the setup phase of gpexpand, thereby requiring a rollback of several preceding steps. This issue has been resolved in this release.
14144	4.1.1.5	Management Tools	Scalability issue during subprocessing in Management Tools File descriptors used during the creation of subprocesses in the Management Tools were not being properly recycled. At large enough scale, this was causing exhaustion of available file descriptors, preventing any further subprocesses from being spawned. In rare cases, this could cause failures of in-progress operations like starting, stopping, expanding, etc. This issue has been resolved in this release.
14010	4.1.1.4	Upgrade/ Downgrade	Log files removed after upgrade Running gp migrator to upgrade removed the master log files. This issue has been resolved in this release.
12659	4.1.1.4	Query Optimizer	"ERROR", "XX000", stanumbers is not a 1-D float4 array Under rare circumstances, invalid statistics caused some statements to fail with the above error. This issue has been resolved in this release.

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
PRFMN-4	4.1.1.4	Performance Monitor	<p>Performance Monitor errors when using IE8</p> <p>When connecting to Performance Monitor using Internet Explorer 8 the web server started streaming errors as soon as a connection to the web server from the browser was made. These errors were being logged in <code>lighttpd-errors.log</code>, potentially causing the partition to run out of disk space.</p> <p>This issue has been resolved in this release.</p>
13023	4.1.1.4	Monitoring: Performance Monitor	<p>Performance Monitor performance issues</p> <p>Performance Monitor's python process was using large amount of memory which was causing other python-based management utilities to throw various errors.</p> <p>This issue has been resolved in this release.</p>
13879	4.1.1.4	Query Execution	<p>Window function using order by/range returned incorrect results</p> <p><code>max() over (order by ... range ...)</code> returned incorrect results when applied to a numeric column.</p> <p>This issue has been resolved in this release.</p>
13817	4.1.1.4	Loaders: gpfdist	<p>Write performance issues when using writable external table</p> <p>gpfdist (loader) was causing performance issues when opening the external writable table target.</p> <p>Prior to this release the default behavior for gpfdist was to open files for synchronous I/O. This meant that any writes on the resulting file descriptor would block gpfdist until the data has been physically written to the underlying hardware.</p> <p>With this release, opening files for synchronous I/O must be explicitly specified by using a new option that has been added to gpfdist (<code>-S</code>).</p> <p>This issue has been resolved in this release.</p>
13596	4.1.1.3	Catalog and Metadata	<p>pg_type corruption</p> <p>Some queries were causing corruption of the <code>pg_type</code> system catalog table that resulted in the database being unusable.</p> <p>This issue has been resolved in this release.</p>
13414	4.1.1.3	Query execution	<p>Out of memory error</p> <p>In some queries that involve Append-only indexes on large number of partition tables, these queries might fail with "out of memory" error.</p> <p>This issue has been resolved in this release.</p>
12617	4.1.1.3	Query Optimizer	<p>Problems with queries with nest loop joins</p> <p>When <code>BitmapAppendOnlyPath</code> is in the inner side of nested loop join, an incorrect query plan was generated causing queries to fail.</p> <p>This issue has been resolved in this release</p>
11695	4.1.1.3	Performance/ Query Execution	<p>Error: "Too many files open"</p> <p>In earlier versions of 4.0.x, some queries that contained multi-way HashJoin in the plan, errored out with a "too many files open" error when HashJoin needed to create a lot of temporary work files for its intermediate state.</p> <p>If you encounter this error, set the following GUC:</p> <pre>SET gp_workfile_type_hashjoin=buffile;</pre> <p>then re-run the query.</p>

Table 1 Resolved Issues in 4.1.2

Issue Number	Resolved In	Category	Description
13461	4.1.1.2	Loaders	<p>Wildcard Character in External Table Name Caused a 404 Not Found Error</p> <p>When loading an external table on Windows systems with <code>gpfdist</code>, if the file name contained a wildcard character, the system returned a 404 Not Found error.</p> <p>This issue has been resolved in this release.</p>
13433	4.1.1.2	Loaders	<p>Bad Rows Might Cause Error</p> <p>When accessing data with <code>gpfdist</code>, bad rows in the last file on the list might cause the system to return the error <code>segment reject limit reached</code>.</p> <p>This issue has been resolved in this release.</p>

Known Issues in Greenplum Database 4.1.2

This section lists the known issues in Greenplum Database 4.1.2. A work-around is provided where applicable.

Table 2 All Known Issues in 4.1.2

Issue	Category	Description
N/A	Performance Monitor	<p>Performance Monitor 4.1.2 asks for IPv6 support during setup. IPv6 is not supported by Greenplum Database 4.1.x</p> <p>IPv6 is supported by Greenplum Database 4.2.</p> <p>Workaround: Enter No to continue in the IPv4 environment.</p>
11297	Query Optimizer	<p>Aggregate functions without Preliminary functions slowing query execution</p> <p>In SQL, an aggregate function definition can optionally be supplemented with a preliminary function that describes how intermediate results are combined. Not providing a preliminary function in the definition can in some circumstances affect performance.</p> <p>The causes of the loss of performance are:</p> <ul style="list-style-type: none"> • The planner may choose GroupAgg instead of HashAgg • More data may need to be moved between nodes. <p>Workaround: If you encounter slow query execution when using aggregate functions, add a preliminary function to the definition.</p>
12919	Loading Data	<p>Using TRUNCATE option with MERGE or UPDATE loses data</p> <p>If <code>TRUNCATE</code> is set to true and the mode is <code>UPDATE</code> in the <code>YAML</code> file, all existing data is wiped out, and nothing is updated when <code>gpload</code> is executed. All data in the target table is lost, and the target table is empty.</p> <p>If <code>TRUNCATE</code> is set to true and the mode is <code>MERGE</code> in the <code>YAML</code> file, all existing records are wiped out when <code>gpload</code> is executed, and all records in the source file are inserted. The target table only contains the new records, but all the previously existing data is lost.</p>
12917	Loading Data	<p>Table Names Cannot Contain Special Characters in the YAML File</p> <p>If a table name contains a special character or space and the <code>YAML</code> configuration file sets <code>REUSE_TABLES</code> to true, the <code>gpload</code> utility fails.</p>

Table 2 All Known Issues in 4.1.2

Issue	Category	Description
12915	Loading Data	<p>Count of Data Formatting Errors Is Incorrect</p> <p>If an error tolerance is set in a <code>YAML</code> file, errors are successfully written to an error table, but the log and output messages incorrectly display zero errors, as shown below:</p> <pre>data formatting errors = 0</pre>
12846	Documentation	<p>Incorrect/missing OS system settings in GPDB 4.1 Installation Guide</p> <p>The following OS System settings were missing from the installation guide:</p> <pre>kernel.msgmni = 2048 net.ipv4.ip_local_port_range = 1025 65535</pre> <p>The <code>kernel.sem</code> OS System setting had an incorrect value in the Installation Guide. The correct value is:</p> <pre>kernel.sem = 250 512000 100 2048</pre>
12823	Backup and Restore	<p>Cannot Restore Functional Index and Functional Trigger</p> <p>When performing a partial schema backup and restore, such as <code>gp_dump</code> using the <code>-t</code> option to dump only the selected tables, a function index or functional trigger cannot be restored, since the underlying functions are not missing from the dump files.</p> <p>Workaround: Restore the required functions separately, prior to restoring the tables than contain a functional index or functional trigger.</p>
12788	Recovery	<p>Mirror Location Not Validated</p> <p>The <code>gprecoverseg</code> utility does not validate the input file for mirrors on same host as the primary segment.</p>
12646	Loaders	<p>MATCH_COLUMNS requires declared distribution key columns</p> <p>For update or delete operations using <code>gpload</code>, the <code>MATCH_COLUMNS</code> section in the <code>YAML</code> file still requires all distribution key columns to be declared.</p>
12590	Email Alerts	<p>Invalid SMTP Server for Email Alerts Causes Degraded Performance</p> <p>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.</p> <p>Work-around: If you notice degraded performance after configuring email alerts, either disable the email alerting feature, or make sure that the specified email server is valid and accessible from the Greenplum master host.</p>
12288	System Catalogs and Metadata	<p>gpcheckcat Error: gp_persistent_* state check found</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>Work-around: Contact EMC Technical Support.</p>

Table 2 All Known Issues in 4.1.2

Issue	Category	Description
12250	Data Loading	<p>Connections to Greenplum Database Remain Open After an ETL Operation</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>Work-around: Manually close the open connections.</p>
12233	Query Execution / Partitioning	<p>DEFAULT Partition Always Scanned Regardless of Query Predicate</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>Work-around: After loading data into a partitioned table, make sure that any data landing in the default partition is split out into a new partition instead of remaining in the default partition.</p>
12192	Query Execution	<p>Sub-optimal Performance when Joining on Columns of Different Data Types</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>Work-around: Columns that are joined frequently should be of the same datatype. If this is not possible, explicitly cast joined columns to the same datatype as part of your SQL statement.</p>
12165, 12173	Query Execution	<p>Queries with Multi-row Subqueries Are Not Supported</p> <p>Currently, Greenplum Database does not support queries that contain multi-row subqueries.</p> <p>Work-around: Rewrite to query statement to avoid using multi-row subqueries.</p>
12120	Installation	<p>Greenplum Database directories show the incorrect group for the owner user if the owner name and the group name are different</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>Work-around: Change the group ownership manually by running <code>chgrp</code>.</p>
12099	DDL / Partitioned Tables	<p>Using ALTER TABLE...DROP COLUMN to Alter the Schema of a Partitioned Table Causes Errors</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>Work-around: If you need to drop columns on a partitioned table and want to avoid this issue, run <code>ALTER TABLE...DROP COLUMN, CTAS</code> the partitioned table, <code>DROP</code> the old table, and rename the new table to the old table name.</p>

Table 2 All Known Issues in 4.1.2

Issue	Category	Description
12082	Query Execution	<p>Error When GROUP BY and a SELECT Clauses Refer to a Column in Different Ways</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>Work-around: Supply an alias for the join result (either in ANSI <code>JOIN</code> syntax in the <code>FROM</code> clause or by expressing the <code>JOIN</code> 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>
11690	Fault Detection and Recovery	<p>Greenplum Database Self-Healing Process Can Create Zero Byte Files</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>Work-around: Contact EMC Technical Support.</p>
10278	System Catalogs and Metadata	<p>pg_relation_size and pg_database_size do not Account for Custom Filespaces</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>
9968	Backup and Restore	<p>Slow gp_dump / gpccrondump Performance</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>Work-around: 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>
8445	DDL	<p>CREATE DATABASE ERROR: "template1" is being accessed by other users</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>Work-around: Use <code>gpstate -e</code> to check the status. When all segments are synchronized, you may retry the <code>CREATE DATABASE</code> command.</p>

Table 2 All Known Issues in 4.1.2

Issue	Category	Description
6279	DDL	<p>Append-Only Table Error After ALTER TABLE...DROP COLUMN</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>Work-around: 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>
5647	Upgrade	<p>Upgrade Utility Requires Standard Prefixes for Segment Data Directories</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>
5517	DDL	<p>Deadlock Detected when Concurrently Altering Tables with Bitmap Indexes</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>Work-around: 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>Transaction Within a Function Not Recognized as a Sub-Transaction</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>
1589	System Catalogs and Metadata	<p>PostgreSQL Usage Statistics Views and Functions do not Work as Expected</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>

Upgrading to Greenplum Database 4.1.2

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

- [Upgrading from 4.1.x.x to 4.1.2.x](#)
- [Upgrading from 4.0.x.x to 4.1.2.x](#)
- [Upgrading From 3.3.x.x to 4.1.2.x](#)
- [Server Configuration Parameters Changes](#)

Upgrading from 4.1.x.x to 4.1.2.x

These instructions are for users currently running 4.1.x.x. An upgrade from 4.1.x.x to 4.1.2.x involves removing the standby master from your Greenplum system (if configured), stopping Greenplum Database, updating the Greenplum Database software binaries, and restarting Greenplum Database.

1. Log in to your Greenplum Database master host as the Greenplum administrative user:

```
$ su - gadmin
```

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

```
$ gpinitstandby -r
```

3. Perform a smart shutdown of your current Greenplum Database 4.1.x.x system (there can be no active connections to the database):

```
$ gpstop
```

4. Run the installer for 4.1.2.x on the Greenplum Database master host. When prompted, choose an installation location in the same base directory as your current installation. For example: `/usr/local/greenplum-db-4.1.2.x`

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

```
source /usr/local/greenplum-db-4.1.1.1/greenplum_path.sh
```

to:

```
source /usr/local/greenplum-db-4.1.2.x/greenplum_path.sh
```

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

```
$ rm /usr/local/greenplum-db
```

```
$ ln -s /usr/local/greenplum-db-4.1.2.x
/usr/local/greenplum-db
```

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

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

7. Log in as root and install the 4.1.2.x binaries on all of the segment hosts. Tar the Greenplum installation directory, use `gpscp` to copy it to the segment hosts, and use `gpssh` to untar it at the segment hosts. Make sure that the `gpadmin` user owns the new installation directory. Also update any symbolic links to point to the new version. For example:

```
$ su -
# gtar -cvf gp.tar /usr/local/greenplum-db-4.1.2.x
# gpscp -f seghostname_file gp.tar =:/
# gpssh -f seghostname_file
=> gtar --directory / -xvf /gp.tar
=> chown -R gpadmin /usr/local/greenplum-db-4.1.2.x
=> rm /gp.tar
=> rm /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.1.2.0
    /usr/local/greenplum-db
```

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

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

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

```
$ gpinitstandby -s standby_hostname
```



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

10. Make sure you have the latest versions of Greenplum Performance Monitor, Greenplum Client, Loader, and Connectivity installed. See “[Greenplum Database Client Tool Packages and Performance Monitor](#)” on page 23 for the latest version number. Installation packages are available for download from the [EMC Download Center](#).

To update the Greenplum Performance Monitor, install the new 4.1.2.x Performance Monitor Web Application and update your environment to point to the new 4.1.2.x Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.1.2.x installation). Greenplum Performance Monitor 4.1.2.x Web Application Packages are available at the [EMC Download Center](#). See the *Greenplum Performance Monitor 4.1 Administrator Guide* for additional installation information.

Upgrading from 4.0.x.x to 4.1.2.x

For detailed upgrade procedures, see the following section:

- [Upgrade Procedure](#)

Upgrade Checklist

This checklist provides a quick overview of all the steps required for an upgrade from 4.0.x.x to 4.1.2.x. Detailed upgrade instructions are also provided in the [Upgrade Procedure](#) section.

Pre-Upgrade Preparation (on your current system)	
<i>* 4.0.x.x system is up and available</i>	
<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 4.1.2.x binaries on all Greenplum hosts.
<input type="checkbox"/>	Copy any custom modules from your existing installation to your 4.1.2.x 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.
Upgrade Execution	
<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 4.1.2.x 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 4.1.2.x system will be up and running.

Post-Upgrade (on your 4.1.2.x system)

* The 4.1.1.x system is up

- | | |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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 compatible version 4.1.2.x Load Tools package. |
| <input type="checkbox"/> | Install the new 4.1.2.x Performance Monitor Web Application and update your environment to point to the 4.1.2.x 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 4.1.2.x installation (if necessary). |

Upgrade Procedure

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



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

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

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

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


```
$ su - gadmin
```
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-4.1.2.x-PLATFORM.zip
```
3. Launch the installer using `bash`. For example:


```
# /bin/bash greenplum-db-4.1.2.x-PLATFORM.bin
```
4. The installer will prompt you to accept the Greenplum Database license agreement. Type `yes` to accept the license agreement.
5. The installer will prompt you to provide an installation path. Press `ENTER` to accept the default install path (for example: `/usr/local/greenplum-db-4.1.2.x`), 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 4.1.2.x installation. For example:

```
$ source /usr/local/greenplum-db-4.1.2.x/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.x.x installation to your 4.1.2.x installation on all Greenplum hosts. For example, shared library files for user-defined functions in `$GPHOME/lib` or PostgreSQL add-on modules (such as `plr.so` or `pgcrypto.so`) in `$GPHOME/lib/postgresql`.

Upgrade Execution

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

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

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

12. (*optional but strongly recommended*) Back up all databases in your Greenplum Database system using `gpcrondump` (or `zfs` snapshots on Solaris systems). See the *Greenplum Database Administrator Guide* for more information on how to do backups using `gpcrondump`. Make sure to secure your backup files in a location outside of your Greenplum data directories.

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

```
$ gpinitstandby -r
```

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

```
$ gpstop
```

15. Source the path file from your new 4.1.2.x installation. For example:

```
$ source /usr/home/greenplum-db-4.1.2.x/greenplum_path.sh
```

16. Update your environment so that it is sourcing your new 4.1.2.x installation.

- a. For example, update the `greenplum-db` symbolic link on the master and standby master to point to the new 4.1.2.x installation directory. For example (as root):

```
# rm -rf /usr/local/greenplum-db
# ln -s /usr/local/greenplum-db-4.1.2.x/usr/
# local/greenplum-db
# chown -R gpadmin /usr/local/greenplum-db
```

- b. Using `gpssh`, also update the `greenplum-db` symbolic link on all of your segment hosts. For example (as root):

```
# gpssh -f segment_hosts_file
=> rm -rf /usr/local/greenplum-db
=> ln -s /usr/local/greenplum-db-4.1.2.x
   /usr/local/greenplum-db
=> chown -R gpadmin /usr/local/greenplum-db
```

```
=> exit
```

- 17.** As `gpadmin`, run the 4.1.2.x version of the migration utility specifying your old and new `GPHOME` locations. If your system has mirrors, use `gpmigrator_mirror`. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

```
$ su - gpadmin
$ gpmigrator_mirror /usr/local/greenplum-db-4.0.4.0
/usr/local/greenplum-db-4.1.2.x
```

- 18.** The migration can take a while to complete. After the migration utility has completed successfully, the Greenplum Database 4.1.2.x system will be running and accepting connections.

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

Post-Upgrade (on your 4.1.2.x system)

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

```
$ gpinitstandby -s standby_hostname
```

- 20.** If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the compatible Greenplum Database 4.1.2.x Load Tools package. Application Packages are available at the [EMC Download Center](#).

- 21.** If you are using the Greenplum Performance Monitor, install the new 4.1.2.x Performance Monitor Web Application and update your environment to point to the new 4.1.2.x Performance Monitor binaries (source the `gpperfmon_path.sh` file from your new 4.1.2.x installation). Greenplum Performance Monitor 4.1.2.x Web Application Packages are available at the [EMC Download Center](#).

- 22.** See “[Server Configuration Parameters Changes](#)” on page 22 for parameter changes introduced in 4.1.0.0. Check your `postgresql.conf` server configuration files for any settings that should be removed or changed. You can use the `gpconfig` utility to check and set parameters.

Troubleshooting a Failed Upgrade

If you experience issues during the migration process, go to the Support page at <http://powerlink.emc.com> or contact Greenplum customer support at one of the following numbers:

United States: 800-782-4362 (1-800-SVC-4EMC)

Canada: 800-543-4782

Worldwide: +1-508-497-7901

Be prepared to provide the following information:

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

Upgrading From 3.3.x.x to 4.1.2.x

Users on a release prior to 4.0.x.x cannot upgrade directly to 4.1.2.x. You must first upgrade from your current release to 4.0.x.x (follow the upgrade instructions in the latest Greenplum Database 4.0.x.x release notes available on <http://powerlink.emc.com>). After you have successfully upgraded to 4.0.x.x, follow the upgrade instructions for [Upgrade Checklist](#).

Server Configuration Parameters Changes

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

- [Deprecated Server Configuration Parameters](#)
- [New Server Configuration Parameters in 4.1.0.0](#)

Deprecated Parameters

Table 3 Deprecated Server Configuration Parameters

Parameter Name	Description of Change
maintenance_work_mem	If using the new memory management features of 4.1.x.x (<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.
max_work_mem	
work_mem	

New Parameters

Table 4 New Server Configuration Parameters in 4.1.0.0

Parameter Names	Description
application_name	Sets the application name for a client session. For example, if connecting via <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.
gp_connectemc_mode	Controls the ConnectEMC event logging and dial-home capabilities of Greenplum Performance Monitor on the EMC Greenplum Data Computing Appliance (DCA). ConnectEMC must be installed in order to generate events. Allowed values are: <ul style="list-style-type: none"> • <code>on</code> (the default) - log events to the <code>gpperfmon</code> database and send dial-home notifications to EMC Support • <code>off</code> - turns off ConnectEMC event logging and dial-home capabilities • <code>local</code> - log events to the <code>gpperfmon</code> database only
gp_fts_probe_timeout	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>).
gp_resqueue_memory_policy	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.

Table 4 New Server Configuration Parameters in 4.1.0.0

Parameter Names	Description
gp_workfile_checksumming	Adds a checksum value to each block of a work file (or spill file) used by <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> .
max_statement_mem	Sets the maximum memory limit for a query. This parameter can only be set by database superusers. It prevents regular users from oversubscribing <code>statement_mem</code> . Default is 2000MB.
statement_mem	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.
smdw_aliases	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 (smdw). For example: <code>smdw_aliases=smdw-1,smdw-2</code>

Greenplum Database Client Tool Packages and Performance Monitor

Greenplum releases the Greenplum Database Performance Monitor and a number of client tool packages on various platforms that can be used to connect to Greenplum Database. The following table describes the Performance Monitor and client tool packages compatible with this Greenplum Database release. Client tool packages are available at the [EMC Download Center](#).

Table 5 Greenplum Database Performance Monitor and Client Tools Version Compatibility

Client Package	Description of Contents	Client Version	Server Versions
Greenplum Clients	Greenplum Database Command-Line Interface (psql) Greenplum MapReduce (gpmapreduce) ¹	4.1.2	4.1.2
Greenplum Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC) PostgreSQL Client C API (libpq)	4.1.2	4.1.2
Greenplum Loaders	Greenplum Database Parallel Data Loading Tools (gpfdist, gpload)	4.1.2	4.1.2
Greenplum Database Performance Monitor	Greenplum Database monitor for query and system performance	4.1.2	4.1.2

1. gpmapreduce is not available on Windows.

Greenplum Database 4.1.2 Documentation

The following Greenplum Database documentation is available from [Powerlink](#). Greenplum documentation is provided in PDF format.

Table 6 Greenplum Database Documentation

Title	Part Number	Revision
Greenplum Database 4.1 Installation Guide	300-012-429	A02 - revised
Greenplum Database 4.1 Administrator Guide	300-012-428	A04 - revised
Greenplum Database Performance Monitor 4.1 Administrator Guide	300-012-430	A02 - revised
Greenplum Database 4.1 Client Tools for UNIX	300-012-432	A01
Greenplum Database 4.1 Client Tools for Windows	300-012-433	A01
Greenplum Database 4.1 Connectivity Tools for UNIX	300-012-434	A01
Greenplum Database 4.1 Connectivity Tools for Windows	300-012-435	A01
Greenplum Database 4.1 Load Tools for UNIX	300-012-436	A01
Greenplum Database 4.1 Load Tools for Windows	300-012-437	A01
Greenplum Database 4.1.2 Release Notes	300-013-403	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

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