

# Moab HPC Suite Enterprise 8.0.0 release notes

The release notes file contains the following sections:

- [New features on page 1](#)
- [Differences on page 6](#)
- [Installation and upgrade information on page 9](#)
- [Known issues on page 10](#)
- [Resolved issues on page 11](#)
- [Key documentation changes on page 13](#)

## New features

The following is a summary of key new features in Moab HPC Suite – Enterprise Edition.

- [Moab Workload Manager on page 1](#)
- [Moab Web Services on page 4](#)
- [Moab Accounting Manager on page 5](#)
- [TORQUE Resource Manager on page 5](#)

### Moab Workload Manager

#### *Data staging*



The old method of data staging has been deprecated in Moab Workload Manager 8.0.0 and will be removed from the product in a future release.

Moab data staging has undergone a major redesign in 8.0.0. Consequently, Moab includes several new features that improve the data staging experience. These features include the following:

- Moab data staging system jobs, separately schedulable from the user job, that increase system performance and utilization by not reserving compute nodes during input or output data-staging unless a compute node's local file system is involved.
- New `msub` options that are required to submit data staging jobs. At least one of `--stagein`, `--stageinfile`, `--stageout`, and/or `--stageoutfile` is required to stage data for your job.

- `--stagein` and `--stageout` specify individual files and/or directories to stage in or out, respectively.
- `--stageinfile` and `--stageoutfile` specify the path to an individual file that contain the paths of files and/or directories to stage in or out, respectively.
- `--stageinsize` and `--stageoutsize` options that give Moab an estimate of the size of the files or directories to stage in or out, respectively, so it can more accurately schedule the associated data staging jobs. For more information, see "[msub](#)".
- A customizable reference data staging job submit filter that sums the size of all input files and passes the sum to Moab. It uses the `--stageinsize` option so you do not have to estimate an input data size and Moab can more accurately estimate wallclock time when scheduling the input data staging system job. For more information, see "[Configuring data staging](#)".
- Customizable reference scripts that use `scp` or `rsync` to stage data (`/opt/moab/tools/data-staging/ds_move_scp` and `/opt/moab/tools/data-staging/ds_move_rsync`, respectively). The scripts work in an out-of-the-box environment, but you should modify one or both to work with your unique environment. You can use the reference scripts as a guide to create your own script that supports data staging with a different Linux file transfer utility or commercial file transfer utility, such as one from Aspera.
- The ability to set generic metrics on partitions (for more information, see "[Per-Partition Settings](#)").
  - The new `DATASTAGINGBANDWIDTH_MBITS_PER_SEC` metric, required for data staging, specifies the transfer rate of the partition in megabits per second. Moab uses it and the `--stageinsize` and/or `--stageoutsize` `msub` options submitted with the job to estimate the wallclock time for better scheduling of the input and output data staging system jobs.
  - The new metric can be dynamically updated by a resource manager or configured by an administrator.
- When you run `checkjob -v` on a data staging job, Moab returns the source and destination of data staging files and their total size. See "[checkjob](#)" for more information.
- Advanced configuration options allow you to rename the default template, support multiple file transfer script utilities in a grid on a per-partition basis, configure a notification email to be sent after a data staging job completes, add a non-default template via `msub`, and use `msub` to return all job IDs in the workflow at submission time. This is useful when you dynamically generate user workflows using scripts that must create dependencies on data-staging system jobs (when input or output data-staging has completed, for example), not the user job. For more information, see "[Configuring data staging with advanced options](#)".

The features associated with data staging in previous Moab releases have been deprecated in this release and will be removed in the next. For more information about data staging in Moab Workload Manager, see "[About data staging](#)".

### *CPU frequency control*

Moab can now ask the resource manager to change the CPU frequency on allocated nodes for submitted jobs. The request is made with the new `CPULOCK` resource manager extension. A user can specify a desired clock frequency in megahertz, a Linux power governor policy name, or an ACPI performance state (P-state) number.

The `mjobctl -m` command has been updated to allow modification of the requested CPU frequency on an already submitted job.

CPUCLOCK has also been added as a job template extension. The job template extension overrides the job script CPUCLOCK extension and the job submission CPUCLOCK option.

### *Improved Performance and Scalability*

Moab 8.0.0 includes new multi-threaded scheduling routines and is now compiled with optimizations enabled (-O). These improvements should increase scalability, efficiency, and performance for the majority of sites. The size of the thread pool can be throttled using the [THREADPOOLSIZE](#) parameter.

### *Finer-grained Logging Timestamps*

The timestamps in the Moab logs now include milliseconds, which can be helpful with higher [LOGLEVEL](#) settings.

### *Deleting checkpoint file doesn't reset IDs*

Moab now persists counters for job and reservation IDs to an external file (`<moab home>/counters`), in addition to persisting the counters to checkpoints. The new external counter file is created during installation. On startup, Moab reads the ID counters from either or both sources and uses the maximum of the two values that it sees as the resumption point for new IDs. This means that you can safely delete a checkpoint file without causing your IDs to be reset.

If both the external counter file and the checkpoint file are missing, Moab refuses to start. This is a precaution to protect the integrity of a relational database or external processes that rely on the IDs. If you need to override this behavior, you can do so by manually creating a new external counter file that starts IDs at an arbitrary number.

### *CANCELFAILEDDEPENDENCYJOBS scheduler flag*

The new `CANCELFAILEDDEPENDENCYJOBS` scheduler flag automatically cancels dependency jobs that will never run because of an unmet requirement. For more information, see [Job Dependencies](#).

### *Reduction in command processing time*

If your system's scheduling cycle regularly takes longer than the `CLIENTTIMEOUT` value, you can configure Moab to fork a copy of itself that will respond to certain information-only client commands (`checkjob`, `showbf`, and `showstart`). This enables you to run intense diagnostic commands while Moab is in the middle of its scheduling process. For more information, see [Reduce command processing time](#).

### *New --workflowjobids option for msub*

A new `--workflowjobids` option for the `msub` command returns all the job IDs, including data-staging system job IDs, in a data-staging workflow at submission time (Moab creates an internal workflow for jobs submitted with data-staging options). For more information, see [Configuring data staging with advanced options](#).

### *NAMI interface improvements*

The Native Accounting Manager Interface now implements all charge policies and chargeable properties implemented by the MAM Accounting Manager Interface.

### *New support diagnostic script*

The `support.diag.pl` script used for gathering information for Adaptive support representatives has been deprecated with the 8.0.0 release. Use the new `support-diag.py` script instead. For more information, see [Diagnostic Scripts](#).

## Moab Web Services

### *PAM authentication support*

Support for PAM (pluggable authentication module) has been added to MWS. PAM treats the user as if it is local to the machine doing the authenticating, and it uses whatever the user is authenticating with, whether it be LDAP or NIS. For information about configuring PAM with MWS, see [PAM \(pluggable authentication module\) configuration using mws-config.groovy](#).



There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and is not supported by Adaptive Computing.

### *OAuth authentication support*

Support for OAuth has been added to MWS. OAuth allows trusted client applications to securely delegate authentication to MWS. Once MWS has authenticated a user by verifying the username and password in LDAP, PAM, or NIS, MWS returns an access token to the client. The client then presents this access token to MWS to access resources. For information about configuring PAM with MWS, see [OAuth configuration using mws-config.groovy](#).

### *Node power management plugin*

The MWS power management plugin acts as a power resource manager; it monitors the power states of the nodes in the Node Configuration File using the TORQUE `pbsnodes` command and a query script. The plugin also allows you to customize the power state of each node when it should be "turned off." For more information, see the documentation included with the plugin in Moab Web Services.

### *Fairshare policy added to MWS Policies object*

The Moab "fairshare" policy has been added to the Policies object in MWS. With this addition, you can get fairshare policy information and make fairshare policy modifications through the MWS API. For more information, see .

### *Modifiable attributes added to the MWS job resource*

You can now use MWS to modify an idle job's system priority, the nodes that the job requests, and the resources per task required for the job. For more information, see [Modify job attributes](#).

### *View and modify single credentials*

You can now query a single credential in MWS. Additionally, you can modify a credential's fields and use the `change-mode` parameter to modify the credential's list fields. For more information, see [Credentials](#).

### *View and modify scheduler priorities*

You can now query and modify the scheduler priorities through the Priority resource in MWS. For more information, see [Priority](#).

## Moab Accounting Manager

### *Accounting mode*

A new **accounting.mode** parameter has been introduced which modifies the way in which accounting actions (e.g. charges, lien, quotes, refunds, etc.) are processed. The accounting mode may be selected at initial product configuration or changed later. The following are valid values for the accounting mode.

- `strict-allocation` - Use this mode if you wish to strictly enforce allocation limits. Under this mode, jobs can be prevented from running if the end-users do not have sufficient funds. This is the default.
- `fast-allocation` - Use this mode if you wish to debit allocations, but need higher throughput by eliminating the lien and quote of strict-allocation mode.
- `notional-charging` - Use this mode if you wish to calculate and record charges for workload usage, but not keep track of fund balances or allocation limits.
- `usage-tracking` - Use this mode if you wish to record workload usage details, but not to calculate a charge nor keep track of fund balances or allocation limits.

### *Commands reference*

The Moab Accounting Manager Administrator Guide now contains a [Appendix A: Commands Reference](#) section that provides the equivalent of a man page for each command.

## TORQUE Resource Manager

### *CPU frequency control*

TORQUE can now set the CPU frequency on requested nodes for submitted jobs. The request is made with the new resource extension. A user can specify a desired clock frequency in megahertz, a Linux power governor policy name, or an ACPI performance state (P-state) number.

The [`galter -l`](#) command has been updated to allow modification of the requested CPU frequency on an already submitted job.

The [`pass\_cpuclock`](#) server parameter was added which allows administrators to track, but not grant, the CPU frequency request portion of a job submission.

### *qrerun all command*

When you execute the `qrerun all` command, you will be prompted for confirmation. TORQUE will then place all running jobs in a queued state without contacting the MOMs. You should only use this when the entire cluster is down and you cannot contact it.

### Node power state control

TORQUE can now set the power state of a node. Depending on the hardware and software capabilities of the node, TORQUE can set the power state to Running, Standby, Suspend, Hibernate, or Shutdown. A [new option](#) was added to the `pbsnodes` command to make this work. The syntax of the command is:

```
pbsnodes -m [running|standby|suspend|hibernate|shutdown] <space delimited list of
nodes to alter>
```

In order to wake a node from a low-power state, Wake-on-LAN must be supported and configured on the node. For more information, see [Changing node power states](#).

## Differences

This section contains differences in previously existing features that require a change in configuration or routine.

- [Moab Workload Manager on page 6](#)
- [Moab Web Services on page 7](#)
- [Moab Accounting Manager on page 7](#)
- [TORQUE Resource Manager on page 8](#)

### Moab Workload Manager

#### *Green policy scripts might need updated values*

If you do string comparisons in your green scripts, you might need to convert the incoming strings from Moab to all uppercase or lowercase.

#### *NAMI scripts and interface changes*

The Native Accounting Manager Interface scripts and interface now use the same SSS XML sent over the wire to MAM for the MAM Accounting Manager Interface type. The default Native AM configuration uses the following supplied NAMI scripts:

```
AMCFG [mam] TYPE=NATIVE
AMCFG [mam] CREATEURL=exec://$TOOLS DIR/mam/usage.quote.mam.pl
AMCFG [mam] STARTURL=exec://$TOOLS DIR/mam/usage.reserve.mam.pl
AMCFG [mam] UPDATEURL=exec://$TOOLS DIR/mam/usage.charge.mam.pl
AMCFG [mam] PAUSEURL=exec://$TOOLS DIR/mam/usage.charge.mam.pl
AMCFG [mam] RESUMEURL=exec://$TOOLS DIR/mam/usage.reserve.mam.pl
AMCFG [mam] ENDURL=exec://$TOOLS DIR/mam/usage.charge.mam.pl
AMCFG [mam] DELETEURL=exec://$TOOLS DIR/mam/lien.delete.mam.pl
```

These new scripts utilize the MAM Perl API and logging; they no longer use the `nami.cfg` or `nami.log` files. If you have created custom NAMI scripts, you will need to adjust them to parse the SSS XML. See [Charging and Allocation Management](#) for examples of the new SSS syntax.

In previous releases, the `UPDATEURL` NAMI script fulfilled the dual function of charging for the previous interval and obtaining a lien for the next interval. In this release, the `UPDATEURL` script only charges for

the previous interval and the **RESUMEURL** script obtains a lien for the next interval (following both **PAUSEURL** and **UPDATEURL**).

#### *mshow -a policy flag deprecated*

The `mshow -a` policy flag (`mshow -a --flags=policy`) has been deprecated and may be removed in a future release.

#### *RPM installations of MAM based on PostgreSQL database*

Because the Moab Suite utilizes a PostgreSQL database in this release, new RPM installations of Moab Accounting Manager are based on a PostgreSQL database instead of MySQL.

#### *Moab versions have been condensed to one build*

There is now only one Moab build to download, install, and configure. In past releases, customers have had to choose between Moab builds that were compiled against ODBC and/or TORQUE libraries. Moab is still compatible with ODBC and TORQUE libraries, however there is now only one package to download. Use of these libraries is enabled/disabled with the existing configuration parameters in `moab.cfg` (e.g., `USEDATABASE ODBC`).

#### *CLASSCFG MAXPROCPERNODE no longer solely global*

You can now specify individual nodes with the **CLASSCFG MAXPROCPERNODE** attribute, which limits the number of processors a class can have on a node.

```
CLASSCFG [batch] MAXPROCPERNODE [n1, n2, n3]=2
```

*The batch class is limited to 2 processors on nodes n1, n2, and n3.*

#### *Complicated workflows contained in a single VC*

Jobs that are created from a workflow template are now associated with a single VC (Virtual Container). Previously, for complex workflows, multiple VCs were created that each contained portions of the workflow jobs.

## Moab Web Services

#### *API version 1 removed*

API version 1 was deprecated in 7.5.0 and has been removed from Moab Web Services in this release. It is highly recommended that you use the most recent API version (version 3). For more information, see [Requesting specific API versions](#).

## Moab Accounting Manager

#### *The low-level fund balance action has been removed*

The low-level fund balance action has been removed since equivalent functionality exists by using the allocation query action. The `--total`, `--available` and `--ignore-liens` options have been removed from the `gbalance` command.

### *Common logging format*

Logging has been enhanced to follow the new formatting guidelines for the Adaptive Computing Standard Log Format. This allows added predictability for common log parsing across Moab suite components.

### *Consumption Rate passed via attribute*

Charging routines look for a consumptionRate attribute on each datum property individually instead of looking for a datum property called ConsumptionRate. This way, consumption rates can be attributed to the specific property to which it applies instead of indiscriminantly applying to all preadd-rate properties.

### *SQLite database no longer supported*

The SQLite embedded database is no longer an option as of this release. It lacked sufficient functionality to allow object customizations and upgrades.

### *PostgreSQL 9.3 as default database*

PostgreSQL 9.3 is now offered as the new default database. Customers that are already using MySQL or earlier versions of PostgreSQL may continue to run on these databases but new customers are encouraged to use PostgreSQL 9.3.

## **TORQUE Resource Manager**

### *The job\_stat\_rate parameter has a new default and function*

Before this release, pbs\_server asked the mother superior of every job for an update on the job every **job\_stat\_rate** seconds. The mother superior now sends updates on every job with its regular status, so there is no need for pbs\_server to regularly poll. Instead, this parameter sets a timeout.

### *Two threadpools and new default for max\_threads*

The **max\_threads** parameter has a new default: The value of min\_threads  $((2 * \text{the number of procs listed in /proc/cpuinfo}) + 1) * 20$ . In previous versions, 20 was 10.

Additionally, threadpools are now split. One-fourth of the threads are allocated for background tasks and three-fourths of the threads are allocated for incoming requests from MOMs and through the API (client commands, Moab, and so forth). Additionally, incoming requests no longer build up indefinitely. If a new request comes in, pbs\_server evaluates whether it is too busy to service the request. For managers, the request is serviced as long as there are at least two threads available in the threadpool. For non-managers, the request is serviced as long as at least 5% of the threadpool is available. When pbs\_server is too busy, it returns the error code PBSE\_SERVER\_BUSY with the message: "Pbs Server is currently too busy to service this request. Please retry this request."

### *Job status polling removed from TORQUE*

Pbs\_server now polls a mom for a job's information only if it hasn't received the information in 5 minutes. Otherwise, the information is communicated with the mom's status information.



*TORQUE no longer searches linearly for the nodes in a node list*

TORQUE now recognizes when a request to run a job specifies a node list. It directly accesses those nodes instead of searching for them linearly.

*The exec\_host list has one entry per node*

The exec\_host list has been condensed to contain one entry per node instead of one entry per execution slot. The node entry contains a string specifying each execution slot index.

TORQUE no longer displays the value of exec\_port in qstat.

*The qstat -f output for CPUs has been reduced*

The output of qstat -f has been condensed to reduce clutter in the CPU section.

## Installation and upgrade information



When installing or upgrading, it is *strongly* recommended that administrators configure Moab with mauth authentication with a complex key value. See [Mauth Authentication](#) for more information.

### Installing Moab HPC Suite – Enterprise Edition 8.0.0

Please see [Installing the RPMs](#) for Moab HPC Suite 8.0.0 – Enterprise Edition for manual or RPM-based installation instructions, respectively.

### Installing TORQUE 5.0.0

#### Installing TORQUE 5.0.0 on RHEL 5

TORQUE 5.0.0 requires the download of boost version 1.36 or later header files in order to build on most RHEL 5 installations. TORQUE 5.0.0 needs the `unordered_map` from the boost library. This did not become part of the boost standard until boost version 1.36. Standard RHEL 5 installations only have boost version 1.33.1. TORQUE 5.0.0 has added a configure option named `--with-boost-path`. This option allows the user to be able to download a newer version of boost without the need to install boost.

You can download a new boost library from [www.boost.org](http://www.boost.org). Extract the tarball to a directory of your choosing. When you are configuring TORQUE you then add the `--with-boost-path=<boost_path>` to the configuration line. The `boost_path` is the location where you unzipped the boost library. Within the unzipped files is a directory named `boost`. This directory contains the `.hpp` files with the needed boost classes. Once the `--with-boost-path` option has been added to the configuration line you can continue to install TORQUE as per the installation guide.

Note that this procedure is not required on RHEL 6 and later versions.

## Upgrading to Moab HPC Suite 8.0.0 – Enterprise Edition

**i** Upgrading an RPM-based installation of the Moab HPC Suite does not work correctly in 8.0.0. See [Known issues on page 10](#) for more information and tips.

The current release is only compatible with a Moab Accounting Manager database schema version of 8.0.0. You can determine your current database schema version by running `goldsh System Query`. If your current database schema version is less than 8.0.0, you will need to migrate your database to 8.0.0 by following the instructions in .

Please see [Upgrading Moab Workload Manager](#) for Moab HPC Suite 8.0.0 – Enterprise Edition for upgrade instructions.

The Moab database schema has been updated for 8.0.0:

Table name	New additions	Changes
<b>Nodes</b>	<ul style="list-style-type: none"> <li>CPUClock (VARCHAR (64))</li> </ul>	<ul style="list-style-type: none"> <li>Column name "Partition" changed to "PartitionName". The word "partition" became a reserved word as of MySQL 5.6</li> </ul>
<b>Requests</b>	<ul style="list-style-type: none"> <li>CPUClock (VARCHAR (64))</li> </ul>	n/a
<b>NodeStats</b>	<ul style="list-style-type: none"> <li>CPUClock integer</li> <li>CPUMaxClock integer</li> <li>CPUMinClock integer</li> </ul>	n/a
<b>Reservations</b>	n/a	<ul style="list-style-type: none"> <li>Column name "Partition" changed to "PartitionName". The word "partition" became a reserved word as of MySQL 5.6</li> </ul>

To migrate your database schema to the newest version, see [Migrating Your Database to Newer Versions of Moab](#).

## Known issues

The following are known issues in Moab HPC Suite – Enterprise Edition. Following each issue description is an associated issue number in parentheses.

- Connecting Moab to MySQL via ODBC can cause Moab to slow down. You can work around this issue by creating the tables with the old storage engine (ENGINE = MyISAM). Also note that other factors will also affect DB performance, such as the type of DB connection (remote vs. local), the DB server's hardware specifications, Moab's configuration for how much to write to the DB (see REALTIMEDBOBJECTS), etc (MOAB-6316).
- Sites that used RPMs to install Moab HPC Suite will not be able to upgrade to 8.0.0 RPMs until 7.2.9 is released. The issue exists in all pre-8.0 RPMs and cannot be worked around without uninstalling the suite. Thank you for your patience as we work to resolve these upgrade issues. Until the issues are addressed, we suggest a fresh installation (either [manual](#) or [RPM](#)) on a separate machine (MOAB-7192).
- Verification and validation of the Moab 8.0.0 integration with Oracle 12c is not yet complete (AC-7407).
- When you submit jobs to Moab with a proxy user but no group specified via MWS, Moab uses the root group rather than the group of the proxy user (WS-2111).
- When you submit jobs with MWS, you must set the commandFile field to the absolute path of the job script on the MWS server. MWS must have read access to the file given in commandFile. Submitting remote job scripts (that is, including the script as part of the JSON payload) is not currently supported (WS-2112).
- Jobs submitted with invalid credentials are put in a held state, instead of rejected, until the administrator can respond. The checkjob command gives administrators further information regarding why the job is held. Blindly assuming that all held jobs should in fact be running RIGHT NOW is not only unsafe, but circumvents intentional Moab policies and workflow. An administrator should exercise care when resolving held jobs. (CVE-2014-5375, MOAB-7478, MOAB-7526)
- When installing or upgrading, it is *strongly* recommended that administrators configure Moab with mauth authentication with a complex key value. See [Mauth Authentication](#) for more information. (CVE-2014-5376, MOAB-7525, MOAB-7480)

## Resolved issues

The following is a list of some key bugs fixed in Moab HPC Suite – Enterprise Edition. Following each issue description is an associated issue number in parentheses.

Resolved issues are aggregated and grouped by the release version in which they were resolved.

- **Moab attempted to power off drained nodes.** Moab now skips drained nodes when it powers off nodes (MOAB-6036).
- **Moab Authentication Bypass issue.** This has been fixed. (CVE-2014-5300, MOAB-7100, MOAB-7524)
- **In environments where a license RM was configured and created a global node, Moab would try to power the global node on and off.** Now Moab does not allow the global node to be powered down as part of a pool of eligible nodes (MOAB-6608).
- **The mdiag -n command did not return the correct amount of available memory on the node.** Running mdiag -n once again returns the correct amount of available memory (DOC-6708).

- **Moab did not honor features configured in the `DEFAULTNODEFEATURES` parameter.** The `DEFFAULTNODEFEATURES` parameters once again works as expected.
- **When Moab would rerun a job after the job migrated to another partition, it attempted the rerun on the wrong partition.** Moab reruns migrated jobs in the correct partition (MOAB-6734).
- **When using a `JOBMIGRATEPOLICY` of `IMMEDIATE`, Moab did not call the `CREATEURL` Native Accounting Manager Interface script.** Moab now calls the `CREATEURL` NAMI script (MOAB-6867).
- **Running `msub` and `qsub` in an interactive environment produced different results.** Running `msub -l` no longer hangs in an interactive environment (MOAB-7165).
- **Moab ignored requirements set using the "hostlist" parameter.** Moab now honors hostlist requirements (MOAB-6475).
- **The MAM GUI did not display as expected.** The MAM GUI was fixed to display the right blue pagination arrow properly (GOLD-238).
- **Certain commands did not recognize accounts that contained a '+'.** Plus signs (and other non-alpha characters) now work correctly in account names (GOLD-249).
- **MAM liens were not clearing.** The charge now removes liens if the lien amount is zero (GOLD-250).
- **The `scan_for_terminated` action in TORQUE affected the speed of `main_loop`.** The `scan_for_terminated` action in TORQUE has been placed on its own thread so that the MOM will be able to service incoming requests in a timely manner (TRQ-2408).
- **When the queue contained more than 50,000 jobs, TORQUE slowed down substantially.** TORQUE no longer slows down with a large number of jobs in the queue (TRQ-2345).
- **When `pbs_server` had a high load, it would get stuck polling.** `pbs_server` no longer gets stuck polling under these conditions (TRQ-2620).
- **When a job with a dependent job was deleted with `qdel`, TORQUE did not clean up the dependent job.** TORQUE now removes the dependent job when you delete its parent (TRQ-2621).
- **When a lot of jobs were run at once, one job would get stuck in an exiting state.** This error no longer occurs (TRQ-2622).
- **The `afterok` dependency did not work as expected.** `afterok` now works correctly (TRQ-2626).
- **When running a large amount of jobs, the server would crash during job recycle.** The crash no longer occurs (TRQ-2628).
- **`pbsdsh` required FQDNs even when they were working in `pbs_server`.** The short names in `pbs_server` are now enough to run `pbsdsh` (TRQ-2632).
- **A deadlock would occur on `job_save` failure.** This error no longer occurs (TRQ-2645).
- **Asynchronous job starts queued in TORQUE but not yet serviced caused Moab to reschedule jobs that would eventually run.** A new job sub-state in TORQUE prevents this from occurring (TRQ-2715).

# Key documentation changes

This section lists some key documentation changes made for Moab - Enterprise Edition 8.0.0TORQUE Resource Manager 5.0.0. Following each change description is a link to the affected page and an associated issue number if applicable.

Features new to this release have been documented but are not listed on this page. For information about those new features and links to their respective documentation, see [New features on page 1](#).

- [Moab Suite on page 13](#)
- [TORQUE Resource Manager on page 14](#)

## Moab Suite

### Additions

- The `JOBMAXSTARTPERITERATION` parameter has been returned to (DOC-1604).
- `MAXIJOB` and `MAXINODE` have been returned to [Usage Limits/Throttling Policies](#).
- [Appendix A: Moab Parameters](#) contains a new parameter called `PERPARTITIONSCHEDULING` (MOAB-6731).
- The `CREATECRED` attribute of `IDCFG` has been returned to .
- A new opening ports section has been added to (DOC-1960).

### Removals

- The "Time-Based Algorithms" section has been removed from [Node Allocation Policies](#) Node Allocation Policies (DOC-1905).
- Removed all references to `QUOTEURL` from [Charging and Allocation Management](#), including the parameter definition. (DOC-1916)
- References to `MINBANDWIDTH` and `MAXLATENCY` have been removed from [Quality of Service \(QoS\)](#) and [Managing Networks](#) (DOC-1920).
- `--with-torque` has been removed from [Installing Moab Workload Manager](#) (DOC-1971).

### Improvements

- The list of parameters that help manage large systems in [Appendix I: Considerations for Large Clusters](#) now includes the `FASTRSVSTARTUPTIME` flag (DOC-1829).
- The [mdiag -C](#) documentation explains that validation is syntax-based, not functional (MOAB-6695).
- The [showstats](#) command page contains an example of specifying an absolute time frame and specifying a relative time frame (DOC-1223).
- A note in the [NCPUS](#) RM extension table differentiates between it and `NODES` (DOC-1596).

- The note about setting the [fairshare target](#) to 0 more accurately describes what happens (DOC-1888).
- The **STOREJOB SUBMISSION** parameter in [Appendix A: Moab Parameters](#) now links to the TORQUE job logging documentation (DOC-1897).
- Several steps and examples in [Connecting to an Oracle Database with an ODBC Driver](#) were modified to clarify and prevent incorrect configuration (DOC-1880).
- The description of the **THREADPOOLSIZE** parameter in [Appendix A: Moab Parameters](#) more accurately describes its function (MOAB-6740).
- **OSCREDLLOOKUP** in [Appendix A: Moab Parameters](#) contains information about parameters required to make the feature work (DOC-1900).
- [Applying the msub submit filter](#) contains more detailed information about configuring your own msub filter and differentiates between server-side filters and client-side filters (AC-6710).
- [Node Location](#) now explains that node range prefixes are alphabetical characters only (MOAB-6918).
- Several job template attribute descriptions on [Job template extension attributes](#) have been improved (DOC-1924-1925).
- Several options for **PRIORITY** in [Node Allocation Policies](#) now explicitly express the unit as MB (DOC-1914).
- Information about using MB/GB when using MEM has been added to [Resource Manager Extensions](#) (MOAB-6708).

## TORQUE Resource Manager

### Additions

- The new `jobdirectory_sticky` parameter is listed in [pbs\\_mom](#) (TRQ-2313).
- The `pbs_server -l` option for HA environments is now in [pbs\\_server](#).

### Improvements

- [Adjusting node state based on the health check output](#) now correctly describes what happens to a node when the health check script no longer reports an error (DOC-1650).
- The `ncpus qsub` resource option was added to [Queue attributes](#) and with notes explaining the difference between it and the `nodes` option (DOC-1596).
- The [node\\_check\\_rate](#) server parameter documentation more clearly states that it is the duration of time a node can fail to send a status update (DOC-1889).
- [Configuring MOMs to launch a health check](#) explains that `status_update_time` controls `node_check_interval` (DOC-1882).
- The overview for the [Prologue and epilogue scripts](#) table clearly explains Mother Superior and sister nodes in the table (DOC-1912).

## Key documentation changes

- The documentation for the [mem](#) qsub resource option explains that it works for single-node jobs only and recommends pmem for multi-node jobs (DOC-1878).
- The pbs\_mom parameters were removed from the pbs\_mom command page and added to [Parameters](#) to resolve linking problems.