

Moab Workload Manager 6.1 and Moab Viewpoint 6.1 Readme

July 2011

The readme file contains the following sections:

- [New Feature Highlights](#)
- [New Feature Details](#)
- [System Requirements](#)
- [Installation Information](#)
- [Known Issues](#)
- [Fixed Bugs \(Customer-submitted\)](#)
- [Documentation and Support](#)

New Feature Highlights

The following is a summary of key new features in Moab Workload Manager 6.1 and Moab Viewpoint 6.1. Note that linked content in the following section points to technical documentation on the associated topics.

- [Powerful 10x scalability improvements](#)
 - Improvements to [msub job submission](#) through multi-threading
 - Practical limits on the [number of queue jobs](#) increased from 40,000 to 500,000
 - Increased client commands scalability with the ability to diagnose [reservations](#) and [triggers](#) with very fast user responses even during scheduling
- [Extended/simplified job array control to ensure service and workload priorities are met and improve the ease of use for users](#)
 - Enhanced grid [workload management options](#) enabling users to select optimal placement across clusters or require single cluster placement
 - New client commands to show only the master job: [showq](#) and [mdiag -j](#)
 - [checkjob](#) now displays sub-job statistics
 - Extended control over job array submissions to ensure SLAs and resource sharing policies are enforced with new usage limit policies: max array job ([MAXARRAYJOB](#)) and max idle array job ([MAXIARRAYJOB](#))
 - [Simplified reporting and management](#) of job arrays for end users with filtering that allows jobs to be summarized tracked and managed at the master job level
- [Advanced GPU scheduling/management to maximize utilization](#)
 - Automatic detection, scheduling and policy-based management of individual GPGPUs for each job with statistics
 - [GPU metrics](#) recorded

- Policy-based management and scheduling using GPU metrics
- Statistics on GPGPU utilization and key metrics
- Exclusively schedule GPGPUs on a per-job basis
- Workload-driven model (for cloud and data center environments)
 - [Workload templates](#) improve self-service automation and cloud service reliability by automating customers' best practices workflows
 - [New holistic management](#) intelligently manages and applies policies, decisions and actions across the set of resources for a service
 - Viewpoint empowers users with the ability to [create and manage environments with virtual and physical machines](#)
 - [Consolidated events reporting capabilities](#) provide better visibility into resource and workload usage trends and issues
 - [Events reporting](#) tracks, filters, and analyzes workload and resource activities across the cloud
 - [Report-specific permissions](#) allow users to access certain reports based on their permission level
 - [Workload-driven model billing](#) integrates with Gold and NAMI
 - New flexibility allows administrators to configure [generic commands](#) that perform actions on the Web server that are not part of Viewpoint's built-in commands.
- [Adaptive Solution Installer](#)
 - Packaged installer simplifies install process for Moab
 - Allows focused client command install on additional machines
 - Provides a template-based install enabling off-site install assistance
- Easier portal customization enabled so organizations can tailor the portal to their specific needs
- [Enhanced chargeback flexibility allows organizations to set chargeback rates for custom billing](#)
- [Optional portal provides easier customization](#) with new user interface framework enhancements
 - Increased productivity by simplifying [job submission](#), [reporting](#) and training for users and administrators

New Feature Details

This section contains detailed information pertaining to specific new features. Information includes new feature implementation instructions and supplies references to the technical documentation for additional help.

Moab Client Command-only Installation

Moab now has the ability to create a "client commands-only" tarball that administrators can use to install just the Moab client commands on a login/client node. This tarball allows an administrator to install the binary Moab client command files, with their man pages, using a single shell command. In addition, the tarball contains a moab.cfg file pre-configured with the Moab host name and port number so the administrator does not have to manually configure this information on the login/client node.

To create the Moab client commands tarball after installing Moab on the head node, enter the following commands. The tarball will have the name "client.tgz".

```
# make client-pkg
```

Copy the tarball to the root directory of the client node, log in to the client node as root, and install the client commands using the following shell commands:

```
# cd /  
# tar xvf client.tgz
```

The Moab client commands are now available on the login/client node. If the client node has a different architecture or operating system than the Moab head node, download the Moab build to the client node and then execute the following commands to create the Moab client commands tarball for the architecture and operating system.

```
# ./configure --prefix=xxx --with-serverhost=yyy --with-serverport=zzz  
# make client-pkg
```

The "yyy" is the Moab headnode host name and the "zzz" is the port number the Moab client commands use to communicate with Moab.

Note: If there is no moab.cfg file installed when making the client commands tarball, the make command will create one using the ./configure arguments or other auto-generated defaults. If there is one already installed, the make will package it in the client commands tarball.

New BecameEligible Field in checkjob Output

New BecameEligible field in checkjob output that displays the timestamp of when a job last changed from a "blocked" state to the "eligible" state; also available in mdiag -j XML output.

```
BecameEligible: Wed Apr 6 08:30:50
```

Blocked Job Priority Increase on Per-credential (QOS) Basis

Job array sub-jobs that hit the job array "concurrent sub-jobs limit" need to continue increasing in priority while they are blocked in a hard-policy violation. This would allow blocked sub-jobs to execute immediately in place of sub-jobs that just completed, instead of allowing unrelated jobs to take the place of the completed sub-jobs as currently occurs since the unrelated jobs increase in priority since they are simply queued, not blocked.

Moab has a new policy configurable on a per-credential basis (QOS) that allows jobs violating policies to still gain in priority so that as soon as the policy violation disappears, the jobs have the priority expected by users instead of starting at the bottom of the queue as currently happens.

The JOBPRIOACCRUALPOLICY and JOBPRIOEXCEPTIONS policies currently exist only at the global level. Adding the options to assign the JOBPRIOACCRUALPOLICY and/or JOBPRIOEXCEPTIONS policies on a QOS basis allows jobs within a specific QOS to still gain in priority even if they are violating a policy or are in some other state beside Idle. Examples of such per-QOS configurations appear follow:

```
QOSCFG[arrays] JOBPRIOACCRUALPOLICY=ACCRUE  
QOSCFG[arrays] JOBPRIOEXCEPTIONS=HARDPOLICY, SOFTPOLICY, IDLEPOLICY  
QOSCFG[arrays] JOBPRIOACCRUALPOLICY=ACCRUE  
JOBPRIOEXCEPTIONS=HARDPOLICY, SOFTPOLICY, IDLEPOLICY
```

Block Moab Backfill on per-QOS Basis

Some users need the ability to submit very important jobs to a specific QOS and not only be at the top of the queue but block all other jobs from starting until all the very important jobs are running; that is, all jobs with the specific QOS are running. This will cause less utilization while the very important jobs are in the queue, but it is a required trade-off for some service-level agreements.

The QOS qflags=ntr (Next-to-Run) option now turns off backfill to ensure resources are available to the very important jobs and are the next to run in all situations.

Show Completed Sub-jobs when Displaying Array Job Sub-jobs

The `mdiag -j -v` command now shows all job array sub-jobs in its normal and XML-based output, including completed sub-jobs that have not been purged via `JOBPURGETIME`.

JobID in Output File Names

Moab now has the ability to substitute parts of the job array ID in sub-job file names such as for standard output and standard error. For example, if a job array submission specifies an output file as

```
msub -t [1-1000] -o /tmp/%J-%I
```

and the job array sub-job ID is `Moab.123[4]`, Moab will perform job ID substitutions for the sub-job's output file name in the following manner:

- `%J` with `Moab.123`
- `%I` with `4`

and the output file name becomes `/tmp/Moab.123-4`. The following job submission transforms an error file name

```
msub -t [1-1000%20] -e /tmp/foo.%I
```

for the sub-job with index = 5 into `/tmp/foo.5`

Job Array Sub-job Execution Order

Moab now guarantees job array sub-jobs start up in their sub-job index order; for example: sub-job 1 starts first, sub-job 2 starts second, sub-job 3 starts next, and so forth.

Allow `mjobctl` to Change Concurrent Job Array Sub-jobs Limit

The Moab [mjobctl command](#) now has the ability to dynamically change the concurrent sub-jobs limit of an array job. If a user submits an array job using the following `msub` command:

```
msub -t myname[1-100]%25
```

no more than 25 sub-jobs will execute concurrently; unless, of course, there are other policies limiting the user's job array. If after the array job submission, the user wants to raise the 25 concurrent sub-jobs limit to 50, the following command would accomplish it.

```
mjobctl -m arraylimit=50 <jobID>
```

Allow Moab to Spread Job Array Sub-jobs Across Multiple Clusters in a Grid

By default, Moab tries to span array jobs across partitions when possible. There are now two ways to force Moab to lock a job array to a single partition, through the [ARRAYJOBPARLOCK parameter](#) in the `moab.cfg` file and through the job submission flags option [arrayjobparlock](#).

To force Moab to lock all array jobs to a single partition when scheduling them, add the following parameter to the `moab.cfg` file:

```
ARRAYJOBPARLOCK TRUE
```

Setting the parameter to FALSE allows Moab to spread the sub-jobs of an array job across multiple partitions.

Users can request Moab to lock an array job to a single partition through the new `arrayjobparlock` flag, as in the following example:

```
msub -t myjob[1-5]%3 -l walltime=30,flags=arrayjobparlock
```

If the default Moab behavior is to lock job array sub-jobs to a single partition, the user can override this scheduling behavior at array job submission time using the [arrayjobparspan flag](#). For example:

```
msub -t myjob[1-5]%3 -l walltime=30,flags=arrayjobparspan
```

Requeue a Running Job and Pull it Back to the Grid-head

Grid users now have the ability to requeue, at the grid level, a job running on a cluster using the following command:

```
mjobctl -R <jobID>
```

To enable this functionality, the class configuration must have the `RESTARTABLE` flag:

```
CLASSCFG[low] JOBFLAGS=RESTARTABLE
```

The job submission must indicate the job can be restarted:

```
msub -r y ...
```

Same Job IDs across Grid Systems

A grid can synchronize job IDs across all clusters with the TORQUE and SLURM resource managers so a single job ID appears in the grid head, the cluster head, and the resource manager job, logs, and output file names.

This feature is enabled by the Moab `SYNCJOBID` parameter on the resource manager configuration and by configuring Moab to only use integers for the job ID. The Moab RM configuration (`RMCFG`) for the SLURM and TORQUE resource managers require additional configuration parameters specific to the resource managers in order to correctly enable the single job ID feature, documented in section 13.2.1.1, Resource Manager Attributes, under the [SYNCJOBID](#) parameter.

Ability to Use Multi-dimensional Throttling Capabilities with Identity Manager

An Identity Manager can now define and use multi-dimensional usage limits. See the following sections of the technical documentation for more details:

- [Usage Limits/Throttling Policies](#)
- [Identity Managers](#)

GPGPU Metrics

Moab receives the GPGPU metrics from TORQUE and assigns shorter names that start with "gpuX_" where "X" is the GPU number within the node, shown in the table that follows. For example, if a node has four GPGPUs, the GPU temperature generic metric names for the four GPGPUs would be `gpu0_temp`, `gpu1_temp`, `gpu2_temp`, and `gpu3_temp`. Note the timestamp is not qualified by GPGPU number since it is the time when TORQUE polled all GPGPUs within the node.

Because each possible GMETRIC name is considered a different GMETRIC type, administrators must ensure

the MAXGMETRIC parameter in the moab.cfg file is increased by the maximum GPGPUs within any single node. For example, if the current MAXGMETRIC parameter has a value of 100 and a cluster/cloud adds nodes with four GPGPUs, there will be an additional $(4 \times 9) + 1 = 37$ new GMETRIC types and the current MAXGMETRIC value of 100 must be increased to 137.

GPU Metric Name	GMETRIC Name
gpu_fan_speed	gpuX_fan
gpu_memory_total	gpuX_mem
gpu_memory_used	gpuX_usedmem
gpu_utilization	gpuX_util
gpu_memory_utilization	gpuX_memutil
gpu_ecc_mode	gpuX_ecc
gpu_single_bit_ecc_errors	gpuX_ecc1err
gpu_double_bit_ecc_errors	gpuX_ecc2err
gpu_temperature	gpuX_temp
timestamp	gpu_timestamp

Since GPU metrics consume memory and Moab will attach GPU metrics to its internal record of each node (even nodes without GPGPUs) recording GPU metrics requires explicit enablement in the moab.cfg file for any resource manager that has the ability to record GPU metrics. [RECORDGPUMETRICS](#) is the Moab configuration parameter for enabling the recording of GPU metrics for a TORQUE resource manager.

```
RMCFG[torque1] flags=RECORDGPUMETRICS
```

When this flag is configured for the resource manager, Moab enables recording [GPU metrics](#) for nodes.

Advanced GPU Scheduling

The Moab msub command's -l option now supports the specification of a GPU mode and/or an error count reset in support of advanced GPU scheduling. For example:

```
msub -l nodes=1:gpus=1:mode:reseterr
```

where mode is optional and defaults to "exclusive" for each job/task. Allowable modes are exclusive, exclusive_thread, and exclusive_process. Likewise, the reseterr is optional. The Moab and TORQUE default is to not reset the GPU ECC memory error counts. See [GPUs in Resource Manager Extensions](#) for more information.

The moab -e Command Checks for Database Connection

When the strict Moab startup command [moab -e](#) evaluates the moab.cfg file, it now also checks for a connection to a configured database and if it cannot obtain a database connection, Moab stops with a clear error message indicating the inability to connect to the database.

New Configurable Option to Disable Memory Cache showq until Jobs Load on a Moab Boot (from checkpoint)

Immediately after a Moab head node reboot or a Moab restart, the non-blocking (and speedy) versions of the following commands do not show jobs or all jobs because Moab does not populate its internal jobs cache used to respond to the commands until after it loads the job checkpoint files and performs one or more scheduling cycles:

- showq
- mdiag -j

The new moab.cfg file parameter [DIAGCACHESTARTITERATION](#) can specify the iteration after which Moab uses its internal jobs cache to return results to these commands. For example, setting DIAGCACHESTARTITERATION to 4 causes Moab to wait until the start of the fourth scheduling cycle or iteration to report jobs using the cache; during the first three iterations, Moab reports jobs directly from its own internal data structures, which requires the user command to block until Moab responds with job data. Since loading job checkpoint files can take some time, the value supplied to this new Moab parameter varies from site to site.

New MAXGRES Usage Limit

Administrators now have the ability to limit a user's simultaneous use of generic resources. New usage limits, MAXGRES and MAXIGRES, allow administrators to restrict the concurrent use of expensive resources, such as software licenses, so a single credential does not hog them. These new usage limits apply to the standard five credential types to which the current usage limits apply. In addition, they support hard (nn) and soft limits (mm), just like all other existing usage limits.

```
ACCOUNTCFG[name] MAXGRES[gresName]=mm,nn MAXIGRES[gresName]=mm,nn
CLASSCFG[name] MAXGRES[gresName]=mm,nn MAXIGRES[gresName]=mm,nn
GROUPCFG[name] MAXGRES[gresName]=mm,nn MAXIGRES[gresName]=mm,nn
QOSCFG[name] MAXGRES[gresName]=mm,nn MAXIGRES[gresName]=mm,nn
USERCFG[name] MAXGRES[gresName]=mm,nn MAXIGRES[gresName]=mm,nn
```

Examples:

```
USERCFG[joe] MAXGRES[matlab]=2
USERCFG[jim] MAXGRES[matlab]=2,4
```

Job Templates Can Now Specify a NODESET

Job templates now have a new [NODESET](#) attribute that permits the template to specify a [node set](#) for any jobs submitted with the template.

System Requirements

Component Compatibility

- Moab Workload Manager 6.1
- Viewpoint 6.1
- Moab Adaptive HPC Suite 6.1
- MCM 6.1
- TORQUE 2.5.7 and TORQUE 3.0.2

Backward Compatibility

When running Moab 6.1 in a grid configuration, administrators must upgrade all grid instances to Moab 6.0 or Moab 6.1. Moab 6.1 is incompatible with versions prior to 6.0.

Hardware Requirements

Adaptive Computing recommends a quad-core system with 8 GB of RAM and at least 100 GB of disk space; such a configuration is sufficient for most operating environments. If you have questions about unique configuration requirements, contact your account representative.

Supported Platforms

Moab works with a variety of platforms. Many commonly used resource managers, operating systems, and architectures are supported.

Resource Managers that Integrate with Moab

The following resource managers integrate with Moab:

- SLURM
- TORQUE

Supported Operating Systems

Moab supports variants of Linux, including:

- Debian
- Fedora
- RedHat
- SuSE

Moab supports variants of UNIX, including:

- AIX
- Solaris

Supported Architectures

Supported hardware architectures:

- AMD x86
- AMD Opteron
- HP
- Intel x86
- Intel IA-32
- Intel IA-64
- IBM i-Series
- IBM p-Series
- IBM x-Series

Installation Information

If you want to use the new Adaptive Solution Installer to install Moab 6.1 as a new installation, perform steps to install Moab using ASI (in the instructions that follow). To manually install Moab, refer to the [Installation and Initial Configuration](#) section of the Moab Workload Manager Administrator Guide.

Instructions for [installing Moab Viewpoint](#) are available in the Moab Viewpoint Administrator Guide.

To install Moab using ASI

1. Go to the Adaptive Computing [download](#) page and click Moab Workload Manager.
2. Log in to your account. (Register if you haven't already.)
3. Choose the version of Moab you want to download and extract the contents of the tarball.

```
> tar -xzf moab-6.1-linux.x86_64-torque.tar.gz
```

4. Run the install file.

```
> ./install
```

5. When asked to see the instructions, enter Y.

Upgrading Moab from Version 6.0 to 6.1

Refer to the help topic [Upgrading Moab](#) for step-by-step instruction on upgrading from version 6.0 to 6.1. Also note that you may need to update your database schema when upgrading from one version to the next. For instruction on migrating your database, see [Migrating Your Database to Newer Versions of Moab](#).

Upgrading Moab from Version 5.4 to 6.1

Because of default directory structure changes between Moab Workload Manager 5.4 and 6.0, you must first upgrade to version 6.0 before upgrading to 6.1 if you are currently running version 5.4. See the [upgrade instructions](#) in the Administrator Guide for more information. Also note that you may need to update your database schema when upgrading from one version to the next. For instruction on migrating your database, see [Migrating Your Database to Newer Versions of Moab](#).

Known Issues

The following are known issues in the 6.1 release. Following each issue description is an associated Moab issue number in parentheses.

- Job templates do not support match filtering against VMEM. (MOAB-1685)
- Account limits not enforced when specified in fairshare XML file. (MOAB-2128)
 - *Workaround:* Specifying the limit in the Moab configuration file (moab.cfg). For example:
ACCOUNTCFG[chemistry] MAXJOB=2
- The globalvars trigger flag only imports node variables when nodes are explicitly defined. (MOAB-2275)
 - *Workaround:* Use the following sample format to import variables: globalvars+<node_id>
- The mjobctl -r command always returns successfully even if the command fails execution. (MOAB-2281)
- Unable to change location of TOOLSDIR and have DataRM work because Moab does not find the Tools.pm file when Perl scripts are not in default location. (MOAB-2322)
 - *Workaround:* Create a symbolic link to the Perl scripts location so Moab finds scripts in the

location it expects.

- Moab not recognizing default domain when sending notifications. (MOAB-2362)
 - *Workaround:* Add an entry to the Moab configuration file (moab.cfg) with a valid email address for the specified user. For example: `USERCFG[root]
EMAILADDRESS=user@adaptivecomputing.com`
- In the Applications Management page in Viewpoint, where the active jobs are shown, clicking on the header of the "VM" column does not result in the list being properly sorted by VM. (VEW-1923)

Fixed Bugs (Customer-submitted)

The following is a list of some key bugs fixed in the 6.1 release. Following each issue description is an associated Moab issue number in parentheses.

- RUNNOW QFLAGS stopped working for zero-tasks jobs - Setting the QoS QFLAGS option to RUNNOW allows zero-tasks jobs to run. (MOAB-1492)
- On XT4 systems, node list ordering lost when standing reservation set - Node list is now sorted according to the priority function before passing the list to ALPS. (MOAB-1646)
- Could not specify more than one momFeature - The Cray XT4 script tools/xt4/job.query.xt4.pl works with multiple TORQUE features and momFeatures. (MOAB-1673)
- Moab attempted to stage the STDERR and STDOUT of interactive jobs - Interactive jobs do not have STDERR or STDOUT files, so Moab no longer attempts to stage them. (MOAB-1695)
- Unloaded fairshare tree for all partitions for a job on a Moab restart - All of the fairshare trees for a job in a multi-partition hierarchical fairshare tree configuration now load. (MOAB-1830)
- Jobs submitted with GRES from the grid-head did not run on a Cray cluster - Moab no longer tries to modify the job through the wrong partition. (MOAB-1854)
- User held jobs evaluated every scheduling iteration - Fixed a problem in a grid with PARALLELLOCATIONPOLICY set to something other than NONE (the default) where held jobs were being evaluated for scheduling (although they were never scheduled). (MOAB-1897)
- Jobs submitted to a reservation without a walltime would not get a default walltime - Jobs submitted without a walltime to a reservation are now assigned the default walltime defined by DEFAULT.WCLIMIT. (MOAB-1950)
- A calloc() failed error occurred when starting Moab - Empty lines were being created in the checkpoint file, causing calloc() errors when Moab started. The problem was fixed so empty lines are no longer created. (MOAB-1961)
- Moab client commands would not work on the jobid of any job with a failed JOBPROLOG - Jobs with a failing JOBPROLOG script no longer disappear from Moab. (MOAB-1985)
- Cray environments could not import environment variables from `msub -E` - The `msub -E` option is now enabled for Cray environments. (MOAB-2055)
- Mult-dimensional policies were ignored when used with an Identity Manager - Mult-dimensional throttling capabilities can now be used with an Identity Manager. (MOAB-2056)
- With CHECKPOINTWITHDATABASE on, the data field of the resource manager's database row in mcheckpoint corrupted on shutdown/restart of Moab - The mcheckpoint table in the Moab database functions properly. (MOAB-2095)
- Moab did not make a reservation for suspended jobs. - Jobs submitted with a higher priority after a job with a lower priority put the lower priority job into a suspended state. (MOAB-2185)
- Massive MRsvDestroyCredLock overhead in simulation mode slowed system - Improvements to simulation mode increased iteration speed. (MOAB-2229)
- JUSTINTIME migration broken with multiple partitions - JOBMIGRATEPOLICY JUSTINTIME now works correctly when Moab is scheduling a large number of jobs. (MOAB-2232)
- Unrecognized character in identity manager file - Moab now interprets an apostrophe correctly in the comments of an identity manager file. (MOAB-2264)

- Jobs in a source-destination grid would not migrate - The active processor count calculated for each job start has been fixed and no longer prevents Moab from migrating jobs. (MOAB-2341)
- DEFAULTCLASS on internal RMs overwrote REMAPCLASS - A fix has been implemented that prevents jobs from being remapped to the source resource manager's default class. (MOAB-2342)
- Setting MAX.WCLIMIT did not cause Moab to recognize the limit - Job's violating the CLASSCFG[] MAX.WCLIMIT now receive a batch hold and do not run. (MOAB-2344)
- Cray jobs of size 0 never migrated in a source-destination grid with a JUSTINTIME migration policy - Moab now correctly calculates the available GRES on a grid partition when attempting to schedule Cray jobs of size 0. (MOAB-2345)
- Moab deferred (not batchhold) jobs with sufficient funds upon job reservation creation but insufficient funds when the job started - An "Insufficient funds" message displays as the reason a job cannot start if funds are consumed between the time of job submission and job execution. (MOAB-2521)
- PREEMPTMINTIME not honored through restart of Moab - Moab now recognizes a job's start time when checking against PREEMPTMINTIME and does not preempt jobs before reaching PREEMPTMINTIME. (MOAB-2525)
- Setting QoS QFLAGS RESERVEALWAYS on Preemptor prevented job preemption - Fixed a problem with jobs not being able to preempt when the preemptor also had the RESERVEALWAYS flag. (MOAB-2555)
- Blocked jobs were being displayed in the eligible queue as NotQueued - Jobs with the NotQueued state now display in the blocked queue and not the eligible queue. (MOAB-2706)

Documentation and Support

Technical Documentation

The online help for the Moab 6.1 release is available in a new framework that offers improvements in navigation and searching. In addition, there is now an [online archive](#) for the previous version of help for Moab Workload Manager and Moab Viewpoint. Use the following links to point to online and PDF versions of the technical documentation for Moab Workload Manager 6.1 and Moab Viewpoint 6.1:

Moab Workload Manager®

Administrator Guide - [HTML](#) - [PDF](#)

Moab Viewpoint®

Administrator Guide - [HTML](#) - [PDF](#)

User Guide - [HTML](#) - [PDF](#)

Customer Support

If you currently have a support services contract with Adaptive Computing and encounter a problem that you can't resolve using the online help documentation, please [submit a support ticket](#) and a technical support specialist will contact you.

For additional assistance, you may also contact us by phone:

+1 (801) 717-3710 (Americas and Asia Pacific)
 +44 (0) 1483 243578 (Europe, Middle East, and Africa)

