



UNIVA CORPORATION

GRID ENGINE DOCUMENTATION

Grid Engine Release Notes

Author:
Univa Engineering

Version:
8.2.0

August 26, 2014

Contents

1	License	1
2	Fixes and Enhancements	6
2.1	Summary	6
2.2	Native Windows Port	6
2.2.1	Supported Functionality on Hosts Running Windows Operating Systems	6
2.2.2	Prerequisites to Use a Windows Hosts in an Univa Grid Engine Cluster	6
2.3	Architectural Changes in Univa Grid Engine	7
2.3.1	Areas of Improvement	7
2.3.2	New Architecture	7
2.3.3	Sessions	7
2.4	Request Limits	8
2.5	Cgroup Support	8
2.6	Distributed Resource Management Application API, version 2.0 (DRMAAv2.0)	8
2.7	Miscellaneous Enhancements	9
2.7.1	Scalability and Scheduling	9
2.7.2	Job Accounting	9
2.7.3	Cluster Diagnostics	9
2.7.4	Job Resource Control	9
2.7.5	Other	10
2.8	Full List of Fixes and Enhancements	11
3	Supported Platforms and Upgrade Notes	15
3.1	Supported Operating Systems, Versions and Architectures	15
3.2	Upgrade Requirements	15
4	Known Issues and Limitations	17
4.1	License Orchestrator below 1.0.2 and Univa Grid Engine 8.2	17
4.2	Job ID's in command output	17
4.3	Required changes for existing scripts when read-only threads are enabled	17
4.4	NUMA specific functionality on AMD processors	18
4.5	Univa Grid Engine on native Windows	18

4.5.1	Restricted functionality of administration and submit commands	18
4.5.2	Restricted functionality of job execution	19
4.6	Univa Grid Engine, accounting file format, Univa UniSight and (ARCo) reporting	19

1 License

TERM SOFTWARE LICENSE AND SUPPORT AGREEMENT

This agreement is between the individual or entity agreeing to this agreement and Univa Corporation, a Delaware corporation (Univa) with its registered office at 2300 N Barrington Road, Suite 400, Hoffman Estates, IL 60195.

1. SCOPE: This agreement governs the licensing of the Univa Software and Support provided to Customer.
 - Univa Software is defined as the Univa software described in the order, all updates and enhancements provided under Support, its software documentation, and license keys (Univa Software), which are licensed under this agreement. This Univa Software is only licensed and is not sold to Company.
 - Third-Party Software/Open Source Software licensing terms are addressed on the bottom of this agreement.
2. LICENSE. Subject to the other terms of this agreement, Univa grants Customer, under an order, a non-exclusive, non-transferable, renewable term license up to the license capacity purchased to:
 - (a) Operate the Univa Software in Customer's business operations and
 - (b) Make a reasonable number of copies of the Univa Software for archival and backup purposes.

Customer's contractors and majority owned affiliates are allowed to use and access the Univa Software under the terms of this agreement. Customer is responsible for their compliance under the terms of this agreement.

The initial term of this license is for a period of one year from date hereof to be automatically renewed at each anniversary unless a written notification of termination has been received 60 days prior to each anniversary.
3. RESTRICTIONS. Univa reserves all rights not expressly granted. Customer is prohibited from:
 - (a) assigning, sublicensing, or renting the Univa Software or using it as any type of software service provider or outsourcing environment or
 - (b) causing or permitting the reverse engineering (except to the extent expressly permitted by applicable law despite this limitation), decompiling, disassembly, modification, translation, attempting to discover the source code of the Univa Software or to create derivative works from the Univa Software.
4. PROPRIETARY RIGHTS AND CONFIDENTIALITY.
 - (a) Proprietary Rights. The Univa Software, workflow processes, designs, know-how and other technologies provided by Univa as part of the Univa Software are the proprietary property of Univa and its licensors, and all rights, title and interest in and to such items, including all associated intellectual property rights, remain only with Univa.

The Univa Software is protected by applicable copyright, trade secret, and other intellectual property laws. Customer may not remove any product identification, copyright, trademark or other notice from the Univa Software.

- (b) Confidentiality. Recipient may not disclose Confidential Information of Discloser to any third party or use the Confidential Information in violation of this agreement.
- (c) Confidential Information means all proprietary or confidential information that is disclosed to the recipient (Recipient) by the discloser (Discloser), and includes, among other things:
 - any and all information relating to Univa Software or Support provided by a Discloser, its financial information, software code, flow charts, techniques, specifications, development and marketing plans, strategies, and forecasts
 - as to Univa the Univa Software and the terms of this agreement (including without limitation, pricing information).
- (ii) Confidential Information excludes information that:
 - was rightfully in Recipient's possession without any obligation of confidentiality before receipt from the Discloser
 - is or becomes a matter of public knowledge through no fault of Recipient
 - is rightfully received by Recipient from a third party without violation of a duty of confidentiality
 - is independently developed by or for Recipient without use or access to the Confidential Information or
 - is licensed under an open source license.

Customer acknowledges that any misuse or threatened misuse of the Univa Software may cause immediate irreparable harm to Univa for which there is no adequate remedy at law. Univa may seek immediate injunctive relief in such event.

5. PAYMENT. Customer will pay all fees due under an order within 30 days of the invoice date, plus applicable sales, use and other similar taxes.
6. WARRANTY DISCLAIMER. UNIVA DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTY OF TITLE, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE UNIVA SOFTWARE MAY NOT BE ERROR FREE, AND USE MAY BE INTERRUPTED.
7. TERMINATION. Either party may terminate this agreement upon a material breach of the other party after a 30 day notice/cure period, if the breach is not cured during such time period. Upon termination of this agreement or expiration of an order, Customer must discontinue using the Univa Software, de-install it and destroy or return the Univa Software and all copies, within 5 days. Upon Univa's request, Customer will provide written certification of such compliance.
8. SUPPORT INCLUDED. Univa's technical support and maintenance services (Support) is included with the fees paid under an order. Univa may change its Support terms, but Support will not materially degrade during any paid term. More details on Support are located at www.univa.com/support

9. **LIMITATION OF LIABILITY AND DISCLAIMER OF DAMAGES.** There may be situations in which, as a result of material breach or other liability, Customer is entitled to make a claim for damages against Univa. In each situation (regardless of the form of the legal action (e.g. contract or tort claims)), Univa is not responsible beyond:

- (a) the amount of any direct damages up to the amount paid by Customer to Univa in the prior 12 months under this agreement and
- (b) damages for bodily injury (including death), and physical damage to tangible property, to the extent caused by the gross negligence or willful misconduct of Univa employees while at Customer's facility.

Other than for breach of the Confidentiality section by a party, the infringement indemnity, violation of Univa's intellectual property rights by Customer, or for breach of Section 2 by Customer, in no circumstances is either party responsible for any (even if it knows of the possibility of such damage or loss):

- (a) loss of (including any loss of use), or damage to: data, information or hardware
- (b) loss of profits, business, or goodwill or
- (c) other special, consequential, or indirect damages

10. **INTELLECTUAL PROPERTY INDEMNITY.** If a third-party claims that Customer's use of the Univa Software under the terms of this agreement infringes that party's patent, copyright or other proprietary right, Univa will defend Customer against that claim at Univa's expense and pay all costs, damages, and attorney's fees, that a court finally awards or that are included in a settlement approved by Univa, provided that Customer:

- (a) promptly notifies Univa in writing of the claim and
- (b) allows Univa to control, and cooperates with Univa in, the defense and any related settlement.

If such a claim is made, Univa could continue to enable Customer to use the Univa Software or to modify it. If Univa determines that these alternatives are not reasonably available, Univa may terminate the license to the Univa Software and refund any unused fees.

Univa's obligations above do not apply if the infringement claim is based on the use of the Univa Software in combination with products not supplied or approved by Univa in writing or in the Univa Software, or Customer's failure to use any updates within a reasonable time after such updates are made available.

This section contains Customer's exclusive remedies and Univa sole liability for infringement claims.

11. **GOVERNING LAW AND EXCLUSIVE FORUM.** This agreement is governed by the laws of the State of Illinois, without regard to conflict of law principles. Any dispute arising out of or related to this agreement may only be brought in the state of Illinois. Customer consents to the personal jurisdiction of such courts and waives any claim that it is an inconvenient forum. The prevailing party in litigation is entitled to recover its attorney's fees and costs from the other party.

12. **MISCELLANEOUS.**

- (a) **Inspection.** Univa, or its representative, may audit Customer's usage of the Univa Software at any Customer facility. Customer will cooperate with such audit. Customer agrees to pay within 30 days of written notification any fees applicable to Customer's use of the Univa Software in excess of the license.
- (b) **Entire Agreement.** This agreement, and all orders, constitute the entire agreement between the parties, and supersedes all prior or contemporaneous negotiations, representations or agreements, whether oral or written, related to this subject matter.
- (c) **Modification Only in Writing.** No modification or waiver of any term of this agreement is effective unless signed by both parties.
- (d) **Non-Assignment.** Neither party may assign or transfer this agreement to a third party, except that the agreement and all orders may be assigned upon notice as part of a merger, or sale of all or substantially all of the business or assets, of a party.
- (e) **Export Compliance.** Customer must comply with all applicable export control laws of the United States, foreign jurisdictions and other applicable laws and regulations.
- (f) **US Government Restricted Rights.** The Univa Software is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the U.S. government or any agency thereof is subject to restrictions as set forth in subparagraph (c)(I)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at 48 C.F.R. 52.227-19, as applicable.
- (g) **Independent Contractors.** The parties are independent contractors with respect to each other.
- (h) **Enforceability.** If any term of this agreement is invalid or unenforceable, the other terms remain in effect.
- (i) **No PO Terms.** Univa rejects additional or conflicting terms of a Customer's form-purchasing document.
- (j) **No CISG.** The United Nations Convention on Contracts for the International Sale of Goods does not apply.
- (k) **Survival.** All terms that by their nature survive termination or expiration of this agreement, will survive.

Additional software specific licensing terms:

Grid Engine incorporates certain third-party software listed at the URL below. These licenses are accepted by use of the software and may represent license grants with restrictions in which Univa is bound to provide. We are hereby notifying you of these licenses.

Unicloud Kits

- Third Party Software is defined as certain third-party software which is provided along with the Univa Software, and such software is licensed under the license terms located at: <http://www.univa.com/resources/licenses/>
- Open Source Software is defined as certain opens source software which is provided along with the Univa Software, and such software is licensed under the license terms located at: <http://www.univa.com/resources/licenses/>

Grid Engine

- Third Party Software is defined as certain third-party software which is provided along with the Univa Software, and such software is licensed under the license terms located at: <http://www.univa.com/resources/licenses/>
- Open Source Software is defined as certain opens source software which is provided along with the Univa Software, and such software is licensed under the license terms located at: <http://www.univa.com/resources/licenses/>

Rev: August 2014

2 Fixes and Enhancements

2.1 Summary

2.2 Native Windows Port

2.2.1 Supported Functionality on Hosts Running Windows Operating Systems

Univa Grid Engine now supports hosts that run certain versions of the Microsoft Windows Operating System as administration, submit or execution host, without the need to install and setup SFU/SUA or Cygwin. Most administration and submit commands of Univa Grid Engine are available on Windows, although some of them with limited functionality. It's also possible to execute native Windows applications under full control of Univa Grid Engine, even GUI applications can show a GUI on the Windows Desktop of the currently logged in user if necessary, e.g. to show MessageBoxes in case of errors.

The Univa Grid Engine master host functionality is NOT available on hosts running Windows Operating Systems, i.e. neither the QMaster, nor the Shadow Daemon, nor the DBWriter functionality are available on Windows. This means that Windows hosts that act as execution, administration or submit hosts have to be connected to a cluster where the QMaster component is running on a UNIX/Linux host. Read further for details about other prerequisites.

2.2.2 Prerequisites to Use a Windows Hosts in an Univa Grid Engine Cluster

Following list shows the supported Microsoft operating system versions and architectures:

Operating System	Version	Architecture
Windows XP Professional (SP3)	XP	32bit
Windows Server	2003, 2003 R2	32bit
Windows Vista	Enterprise, Ultimate	32bit, 64bit
Windows Server	2008, 2008 R2	32bit, 64bit
Windows 7	Professional, Enterprise, Ultimate	32bit, 64bit

Table 1: Supported Windows Systems, Versions and Architectures

Full support for Windows 8 and 8.1 (32bit, 64bit) is being validated and expected to be released with one of the upcoming patches for Univa Grid Engine.

Please note that the following prerequisites need to be fulfilled before a host running one of the operating systems mentioned above can be used:

- All execution hosts have to be members of one Active Domain
- All user accounts of users that should interact with the Univa Grid Engine system have to be domain users.

- Passwords for those users have to be registered at the Univa Grid Engine system.
- The certificates that are used to encrypt these passwords have to be available on the Windows hosts.
- All user names have to be the same on Unix/Linux and Windows hosts.
- The Univa Grid Engine admin user needs full network access, to the \$SGE_ROOT directory, to the certificate directory (if these are shared and not copied over) and to the network shares where job output files have to be created.
- During installation, for each Microsoft Windows host, the account of a user with permissions to write to the C:\Windows directory and to the registry is needed. This usually is the local Administrator, but can be any other user with sufficient permissions.

2.3 Architectural Changes in Univa Grid Engine

2.3.1 Areas of Improvement

Several architectural changes have been applied to Univa Grid Engine 8.2 that improve time required for job submission, scheduling performance, job dispatching and the overall cluster throughput. Compared to previous versions of the product Univa Grid Engine 8.2 is up to 3x faster.

In particular big clusters with a large user base and a huge amount of short and medium-sized workload will greatly benefit from these enhancements. For end users of such clusters this will be visible by improved responsiveness of all client and daemon application. Administrators will see improved utilization of multi-core hardware used for the qmaster component as well as by rapid job throughput.

2.3.2 New Architecture

Improved utilization of the underlying qmaster hardware is the reason for the performance improvements realized in Univa Grid Engine 8.2. This is achieved by an additional pool of threads in the qmaster process. The new thread pool (reader threads) is responsible for processing read-only requests exclusively that are triggered by commands such as qstat, qhost, qselect. Other threads (worker threads), that were already available in previous versions of Univa Grid Engine, can now exclusively process read-write requests. Such requests are generated by commands such as qsub, qalter, qmod. Decoupling read-write and read-only requests are the key for the improved performance because up to 64 reader-threads can now work in parallel.

In addition to the above changes, the internal memory architecture has been changed. Reader and worker thread pools hold one copy of the configuration/status information. Both datastores are synchronized via events. Reader threads might have a 'slightly stale' view of the master state. The result is that all reader threads and also worker threads can work in parallel. A new Univa Grid Engine object type named session has been introduced that removes the 'slightly stale' view for read requests when this must be avoided.

2.3.3 Sessions

Sessions enforce additional synchronization between client and reader threads to avoid polling that is required to maintain a consistent view. Sessions (may) slightly slow down read requests to

ensure consistency but they do not thwart internal operations of the Univa Grid Engine system itself. Usually, synchronization happens so fast that it is not noticed by the end user. Therefore, there is no need to use sessions at all in small cluster.

2.4 Request Limits

Request limits allow administrators to define limits for incoming qmaster requests sent by client commands. Requests that are sent by command line clients might get rejected when a limit is exceeded. This allows regulation and control over client commands before things get critical in the Univa Grid Engine system.

Requests can be filtered according to request type (GET, ADD, MOD, DELETE), request object (Job, Job Class, Queue, . . .), client command name (qsub, qstat, qalter, qconf), user and hostname. Limits are ignored for managers and administrators to avoid lockout.

2.5 Cgroup Support

Cgroups is a Linux kernel feature to limit, account and isolate resource usage of process groups. Univa Grid Engine is integrated with this facility because it provides irrevocable CPU isolation, NUMA domain isolation, safer job suspension, job reaping and additional ways to limit main and virtual memory for jobs. Univa Grid Engine uses this functionality and it allows to do additional modifications of existing Cgroups through customizable prolog scripts.

64bit Linux distributions (like RHEL 6.0 / CentOS 6.0 / Ubuntu 12.4 / SUSE 12.3) support Cgroups when the *libcgroups* library is installed.

If Cgroups functionality is enabled in Univa Grid Engine then it is used for:

- memory limitation (m_mem_free)
- virtual memory limitation (h_vmem)
- automatic cpuset creation (when -binding is specified during job submission)
- NUMA domain isolation (when -mbind is specified during job submission)
- process reaping when jobs get deleted (due to qdel or when h_rt is reached)
- process suspension (triggered by manual/subordiante/suspend_threshold suspension)

2.6 Distributed Resource Management Application API, version 2.0 (DRMAAv2.0)

DRMAA2 defines an open standard for an API that supports the creation of job workflows as well as cluster monitoring applications. It was evolved from the widely adopted DRMAA1 specification by the Open Grid Forum (<http://www.ogf.org>) and offers a set of around 100 standardized C functions. It has a notion of queues, slots, machines, job classes, advance reservations and more. Applications may hold multiple, concurrent and persistent sessions that do not only allow job control but also cluster monitoring of machines, queues and non-DRMAA jobs. The internal architecture is event-driven to avoid performance drawbacks through polling. DRMAA2 offers extensible data structures so that Univa Grid Engine specific functionality can be added in later versions of the library without breaking compatibility with existing applications.

The DRMAAv2 specification is currently under final review.

Univa Grid Engine 8.2 comes with a developer preview version of a C implementation of the DRMAA2 C language specification. The C API is currently only available for the 64-bit Linux operating system. The specification of other language bindings is currently in progress.

DRMAA1 is fully supported in Univa Grid Engine 8.2 but users are encouraged to adopt the new standard. If you have questions or requirements for specific language bindings then please contact our support team.

2.7 Miscellaneous Enhancements

2.7.1 Scalability and Scheduling

Several bug fixes and improvements have been applied to Univa Grid Engine 8.2. Corrections of the sharetree usage calculation for array tasks as well as fixes for job dependency nets and internal thread synchronization improve the scheduler performance.

With this version of the product, it is also possible to enforce the release of resources that are booked for advance reservations so that intended jobs can consume the underlying resources.

2.7.2 Job Accounting

Job timestamps are recorded in milliseconds in accounting and reporting. User name and host are recorded for job deletions and available in the accounting file as well as the submit host, submit switches used at the commandline and the specified working directory of a job.

Additional memory metrics can be accessed in the accounting file as well as during runtime of a job. Job usage information is stored as 64bit values.

Univa Grid Engine 8.2 supports 32bit job ID numbers with a configurable rollover.

2.7.3 Cluster Diagnostics

Annotations for queue state changes can be logged to inform other users or managers for reasons of unavailability.

Details about event clients have been added that make it easy for managers to identify users and hosts that trigger certain commands.

2.7.4 Job Resource Control

Users can now specify dynamic runtime limits for jobs. The limit enforcement of resources is now configurable.

2.7.5 Other

Server side JSV scripts can now use any client command (like qstat) to retrieve more information from the Univa Grid Engine system. This does not cause delay due to deadlocks and deadlock detection like it was in previous versions when Univa Grid Engine command line clients were started in JSV routines.

HP Insight CMU integration is added to Univa Grid Engine. For more information, please contact our sales or support team.

Univa Grid Engine supports the Cray XC-30 system architecture. For more information, please contact our sales or support team.

2.8 Full List of Fixes and Enhancements

Univa Grid Engine 8.1.7p1 - 8.1.7p5

- GE-4996 job reporting entry "waiting for license" created in non-LO system
- GE-4982 scheduler param MAX_SCHEDULING_TIME can get exceeded as long as jobs can be dispatched
- GE-4883 d_rt limit is not documented
- GE-4599 string complex with spaces is rejected when initialized on host level
- GE-4629 Kill a job when h_rss is exceeded
- GE-4728 maxrss and maxpss should be available in online job usage
- GE-4738 stop scheduling other jobs until a high priority job has been scheduled
- GE-4744 qrsh jobs started in terminal in background are suspended and qdel does not work
- GE-4762 GE-4744 new qrsh switch to configure behavior when running in background of a job control enabled shell
- GE-4772 qrsh client which cannot obtain exit state from execution host should not terminate with exit state 0
- GE-4812 execd aborts when executing parallel jobs and execd_params ENABLE_MEM_DETAILS=true is set
- GE-4822 Execution daemon erroneously reconnects to qmaster
- GE-4828 Use system defined connection backlog value for UGE server socket setup
- GE-4831 Need option to set master task job to failed when not all slave tasks report job finish
- GE-4836 cryptic error message regarding the clash of 2 unexpected job states
- GE-4840 slave tasks of tightly integrated job running on master task host should be reported before master task termination

Univa Grid Engine 8.2.1 beta 1

- GE-3072 GUI jobs on Windows Vista only starting when there is a user logged into the system
- GE-4124 Inconsistency in job class manual pages
- GE-4141 qstat doesn't report array job concurrency limit
- GE-4202 JC's that specify a positive priority value cannot be used by non-manager to submit new jobs
- GE-4460 replace not thread safe strerror() by sge_strerror()
- GE-4704 limit of submission rate on user level
- GE-4741 garbled version information and outdated checkin date in man pages
- GE-4751 GE-3406 Create native Windows text installer
- GE-4769 qconf doesn't handle full qualified Windows user names properly
- GE-4797 gdi_request_limits should allow to define limits for certain users or hosts

- GE-4798 command, object and request parts of `gdi_request_limits` are not verified if they are valid
- GE-4799 `qstat -j '*'` takes very long with more than 100K jobs
- GE-4800 Users that are not managers cannot delete own GDI sessions
- GE-4801 source token in `gdi_request_limits` are ignored
- GE-4802 request type and object type in `gdi_request_limits` need to be uppercase
- GE-4809 wildcard character for 'source' within `gdi_request_limit` is rejected
- GE-4810 NONE as `gdi_request_limit` is rejected
- GE-4814 `qhost -si help` output is incorrect
- GE-4815 many commands do not accept NONE as `session_id` for the `-si` switch
- GE-4821 "`qconf -stl` and `-at/-kt "reader"` are missing in the help output of `qconf`"
- GE-4826 man pages do not explain GDI sessions and corresponding commands
- GE-4849 on native Windows, a job must be set to error state if the job users password can't be read
- GE-4850 on native Windows, the `execd` can't read spooled jobs after `execd` restart
- GE-4852 on native Windows, PEs that use `/bin/true` as `start_proc_arg` fail
- GE-4854 on native Windows, the UGE Starter Service fails to start the `execd` at boot time
- GE-4855 on native Windows, after the `execd` was restarted, it doesn't recognize jobs end
- GE-4857 the native Windows shepherd crashes before or when freeing the job environment
- GE-4863 on native Windows, the shepherd crashes if no explicit user home directory is defined
- GE-4865 the UGE Job Starter Service starts GUI jobs in the foreground even if the job environment variable `SGE_BACKGND_MODE=1` is set
- GE-4881 GE-3406 The resulting job environment doesn't contain the user environment from the Windows user profile and variables specified by `-v` or `-V`
- GE-4895 GE-3406 use SGE admin user and the local Administrator to install UGE on native Windows
- GE-4899 on native Windows, executing a job can cause `execd` crash if the job user can't be logged on
- GE-4901 on native Windows, any job opens a Window on the visible desktop as long as `SGE_BACKGND_MODE=1` is not specified
- GE-4902 event clients see incorrect state of JC's and GDI-get requests show incorrect JC's
- GE-4903 `qalter -mods/-adds/-clears` switches do not work
- GE-4904 Change of certain job attributes do not trigger modify event of job/task
- GE-4907 if the job users password is missing in the `sgepasswd` file, a wrong error message is written to accounting
- GE-4915 improve error logging if `sge_getpwnam_r()` fails
- GE-4916 the host isn't set to error state if the UGE Job Starter Service is not running

- GE-4927 shepherd daemon might report incorrect job exit status
- GE-4929 manual execd installation creates default queue setup with zero host slots
- GE-4934 install_execd.bat fails to install services if the QMaster port is read from /etc/services
- GE-4939 job start fails if a starter_method is configured
- GE-4942 suspend state of jobs is not visible in qstat after qmod -[u]sq and on suspend on subordinate

Univa Grid Engine 8.2.0 FCS

- GE-1039 qmaster logs warnings even when log_level is set to log_err
- GE-2544 upgrade qmake using gmake 4.0
- GE-2822 tight integration does not work with two queues on one host
- GE-3291 Adding a new PE should use NONE instead of /bin/true for start/stop_proc_args
- GE-3698 enhancement for qstat/qacct to see cwd and submission command of job
- GE-3813 user configurable max job number
- GE-3840 openmpi jobs incorrectly get killed due to memory limit
- GE-3853 IO in online usage and accounting is not explained
- GE-3927 adding a way to switch on/off the limit enforcement by execd
- GE-3990 /proc/cpuinfo file is opened when submitting job
- GE-4022 update jemalloc in 3rdparty directory of lx-amd64
- GE-4049 Use 64 bit values to hold job usage data
- GE-4076 During the modification of mail recipients in jobs derived from JC invalid mail addresses will be added.
- GE-4085 provide more event client information
- GE-4203 normal users are allowed to specify positive priority values in JC's
- GE-4209 changes to ibm-loadsensor for AIX 6 -> oslevel should be used instead to detect arch string
- GE-4246 use more precise timestamps in job reporting and accounting
- GE-4247 request a way to be able to control and manage no. of qstat calls.
- GE-4287 record 'qdel' invocation in accounting
- GE-4298 write online usage information to reporting file/database
- GE-4336 bootstrap man page does not mention Postgres spooling as supported spooling_method
- GE-4338 race condition in signalling the job at startup in shepherd
- GE-4344 improve shutdown speed of (builtin) interactive jobs
- GE-4414 General Annotate Functionality
- GE-4420 Provide an easy mechanism to drain the cluster
- GE-4475 Make it possible to set queue instances into error state via qmod command
- GE-4600 functionality to enable/disable backfilling
- GE-4670 Improvements to SGE_JSV_TIMEOUT within script or server side qmaster params.
- GE-4731 show latest resource reservation in qstat -j <job_id>

- GE-4743 packint64() and unpackint64() pack and unpack only 32 bit
- GE-4754 at most one resource reservation is done when the cluster is full (all queue instances are full)
- GE-4759 qsub -sync yes -t n-m does not print the exit code for every task
- GE-4766 qconf command line parsing shows problems when empty strings are used for command line parameters
- GE-4768 GE-4085 Enhance qconf -secl to show the owner/user of the event client
- GE-4773 Fix memory corruption in UGE Job Starter Service that causes crashes in rare cases
- GE-4835 replace confusing "User does not exist" error message if NIS is broken
- GE-4842 can start one task too much on slave host of a tightly integrated job
- GE-4858 update PostgreSQL libraries to current version 9.3.4
- GE-4859 update Berkeley DB libraries to current version 6.0.30
- GE-4860 update openssl libraries to current version 1.0.1h
- GE-4906 random connect problems for PE slave or qmake jobs when delivering job to execution daemon
- GE-4914 make d_rt a queue attribute
- GE-4920 add maxrss and maxpss to the accounting file
- GE-4924 add submit host to the accounting file
- GE-4925 add working directory to the accounting file
- GE-4926 add submission command line to accounting file
- GE-4931 qrsh client lacks -adds, -mods ... switches.
- GE-4933 arseqnum file is not backed up by inst_sge -bup
- GE-4946 on native Windows, qrsh output is broken if much output is transferred at once
- GE-4950 qmake does not inherit -q switch
- GE-4962 online usage is lost for some jobs
- GE-4963 broken quoting of job arguments with spaces on win-x86 (native Windows)
- GE-4966 The reporting man page has invalid information for the job log
- GE-4972 provide a means to identify jobs which lead to high scheduling times
- GE-4975 reader event client automatically reregisters after "qconf -kec 3"
- GE-4979 installation changes improve install experience and lower CPU+memory impact
- GE-4980 improve man page on thread creation/killing options
- GE-4982 scheduler param MAX_SCHEDULING_TIME can get exceeded as long as jobs ...
- GE-4988 submission of a jc, which contains wrong entries triggers a qmaster crash
- GE-4996 job reporting entry "waiting for license" created in non-LO system
- GE-5021 m_topology_inuse is lost in case of complex_values changes

3 Supported Platforms and Upgrade Notes

Univa Grid Engine 8.2 supports various hardware architectures and versions of operating systems.

3.1 Supported Operating Systems, Versions and Architectures

Operating System	Version	Architecture
SLES	10,11 x86,	x86-64
RHEL	5 or higher, 6 or higher, 7	x86, x86-64
CentOS	5 or higher, 6 or higher, 7	x86, x86-64
Oracle Linux	5 or higher, 6 or higher, 7	x86, x86-64
Ubuntu	10.04LTS - 14.04LTS	x86, x86-64
Oracle Solaris	10, 11	x86_64, SPARC 64bit
HP-UX	11.0 or higher	64bit
IBM AIX	6.1 or later	64bit
Apple OS X	10.8 (Mountain Lion) or higher	x86, x86-64
Microsoft Windows	XP Professional (SP3)	32 bit
Microsoft Windows	Server 2003 / 2003 R2	32 bit
Microsoft Windows	Vista Enterprise / Ultimate	32 and 64bit
Microsoft Windows	Server 2008 / 2008 R2	32 and 64bit
Microsoft Windows	7 Professional / Enterprise / Ultimate	32 and 64bit

Table 2: Supported Operating Systems, Versions and Architectures

PLEASE NOTE: Hosts running the Microsoft Windows operations system cannot be used as master or shadow hosts.

PLEASE NOTE: Univa Grid Engine 8.2 qmaster is fully supported on Linux and Solaris. We provide binaries in Univa Grid Engine 8.2 for running the qmaster on other operating systems but they are not supported and delivered as a courtesy. If you require qmaster support on other architectures please contact us at support@univa.com.

PLEASE NOTE:: if you require Univa Grid Engine support for older versions of the above operating systems please contact our sales or support team.

3.2 Upgrade Requirements

This is a summary of the Upgrade Matrix that describes how you can carry out the transition from Sun or Oracle Grid Engine 6.2uX, Univa Grid Engine 8.0.X, Univa Grid Engine 8.1.X to

Univa Grid Engine 8.2 when you are currently using classic, BDB local spooling or PostgreSQL spooling. If the current version of Grid Engine you are using is missing in the overview, then please look at the full Upgrade Matrix located in the section Updating Univa Grid Engine in the Installation Guide.

Version	Upgrade Method
Univa Grid Engine 8.1.X	Backup/Restore
Univa Grid Engine 8.0.X	Backup/Restore
Oracle Grid Engine 6.2u6-6.2u8	Backup/Restore
Sun Grid Engine 6.2u5	Backup/Restore
Sun Grid Engine 6.2u1-6.2u4	Upgrade to SGE 6.2u5 and then Backup/Restore
Sun Grid Engine 6.2 FCS	Upgrade to SGE 6.2u5 and then Backup/Restore

Table 3: Upgrading from SGE, OGE, UGE 8.0.X and UGE 8.1.X to Univa Grid Engine 8.2.X

4 Known Issues and Limitations

4.1 License Orchestrator below 1.0.2 and Univa Grid Engine 8.2

Univa Grid Engine 8.2 uses the full range of 32bit values as ID's for jobs and advance reservation. License Orchestrator below version 1.0.2 cannot handle ID's of that size.

There are two options to address this limitation:

- Upgrade the License Orchestrator cluster to version 1.0.2 before you install/upgrade to Univa Grid Engine 8.2

or

- Define the variable `MAX_JOB_ID` in the `qmaster_params` attribute of the global configuration of your Univa Grid Engine 8.2 cluster after upgrade or installation. Set `MAX_JOB_ID` to 9999999 there before you connect the Univa Grid Engine 8.2 cluster to License Orchetrator 1.0 or 1.0.1

4.2 Job ID's in command output

Univa Grid Engine now uses the full 32-bit range for job ID's. Due to this the output format of client commands has changed to be able to display the job ID completely. Existing scripts that parse the output of commands like `qstat/qhost` might need to be adapted before they can be used with Univa Grid Engine 8.2.

4.3 Required changes for existing scripts when read-only threads are enabled

Existing scripts that use commands to add/modify/delete Univa Grid Engine objects (like `qsub`, `qalter`, `qmod`, ...) and commands that only get information (like `qstat`, `qhost`, `qselect`, ...) *might not* work as expected if they are used unmodified in Univa Grid Engine 8.2 with enabled read-only threads.

The reason for this is that read-only and read-write requests are then executed independently from each other so that read-only requests (like `qstat`, `qhost`, `qselect`, ...) might not see the outcome of previously executed read-write requests.

To solve this issues the scripts should use sessions for all commands where an execution dependency exists. This can be done by creating a session key with `qconf -csi` command and by passing this session key to all commands that depend on each other using the `-si` switch of the corresponding command.

Example:

```
> qconf -csi
5615436
```

```
> qsub -si 5615436 ...
Your job 82763 ("JobName") has been submitted

> qstat -si 5615436 -j 82763
```

The Univa Grid Engine system guarantees then that dependent commands can see the outcome of previously executed commands (e.g. `qstat` will see the previously submitted job 82763)

Find more information concerning sessions in section 8.2 “Using sessions to communicate with the system” of the UGE Users Guide.

4.4 NUMA specific functionality on AMD processors

AMD processors have a different NUMA model than Intel processors. Currently the NUMA implementation (per socket memory management) is aligned to the Intel NUMA model. Other features and functions are not affected.

4.5 Univa Grid Engine on native Windows

4.5.1 Restricted functionality of administration and submit commands

- These options will fail or be ignored if a job is submitted to a Windows host:
 - `qalter`, `qsub`, `qresub`, `qrsh`, `qsub`
 - * `-c` - Checkpointing is not supported on Windows
 - * `-ckpt` - Checkpointing is not supported on Windows
 - * `-m` - Mail sending is not yet implemented
 - * `-M` - Mail sending is not yet implemented
 - * `-notify` - There are no notification signals on Windows
 - * `-noshell` - The shell concept works differently on Windows
 - * `-pty yes` - There is no `pty` on Windows
 - * `-shell yes` - The shell concept works differently on Windows
 - * `-S` - The shell concept works differently on Windows
 - `qlogin` is not implemented
 - `qrsh` is available only with command, `qrsh` without a command is not implemented
- These options will fail or be ignored when run on a Windows host:
 - `qacct`
 - * `-g [group_id]` - not possible to resolve the UNIX group ID on Windows
 - `qconf`
 - * all options with `-m` fail, because opening an editor is not yet implemented
 - `qlogin` is not implemented
 - `qrsh` can be used only with a command, not for an interactive login.
 - using access lists that contain UNIX groups will possibly fail

4.5.2 Restricted functionality of job execution

- Checkpointing is not supported
- There is no online usage of running jobs
- Changing the process priority of running jobs is not possible

4.6 Univa Grid Engine, accounting file format, Univa UniSight and (ARCo) reporting

Univa Grid Engine timestamps have changed from seconds to milliseconds in the Univa Grid Engine accounting file.

The Univa Grid Engine reporting parameters configured by `reporting_params` have changed. All timestamps that were previously in seconds are now reported in milliseconds. This change affects the reporting file format, UniSight reporting and ARCo.

Users using Unisight should not upgrade to Univa Grid Engine until an update to Unisight is available. Users who use dbwriter to process the Grid Engine reporting data or who created tools which directly process the output of the UGE reporting file should adapt their backend tools to properly process the new time stamps.