





Getting Started Guide (Printable)

The information presented here is also available in individual pages in the [Getting Started](#) section. This page is designed to enable you to easily export the Getting Started information to a PDF or Word document.

To print a version of this document, log in to wikis.sun.com, click Tools, then select Export to PDF or Export to Word.

Contents	
• 	Getting Started
• 	How the System Operates
• 	How Resources Are Matched to Requests
• 	Choosing a User Interface
• 	Users and User Categories
•	Using the Wiki
•	How to Add a Comment
•	How to Monitor Documentation Changes
• 	How to Print



Getting Started



[Getting Started Guide \(Printable\)](#)

Introducing the Sun Grid Engine System

A grid is a collection of computing resources that performs tasks. In its simplest form, a grid appears to users as a large system that provides a single point of access to powerful distributed resources. In its more complex form, a grid can provide many access points to users.

The Sun Grid Engine software enables you to apply resource management strategies to distribute jobs across a grid. Users can submit millions of jobs at a time without being concerned about where the jobs run. The system supports clusters with up to 63,000 cores.

Sites configure the system to maximize usage and throughput, while the system supports varying levels of timeliness and importance. Job priority and user share are instances of importance.

The Sun Grid Engine software provides advanced resource management and policy administration for UNIX and Windows environments that are composed of multiple shared resources. For more on Sun Grid Engine's features, see the [product site](#).

To familiarize yourself more with the product and the wiki, explore the resources below:

Topic	Description
How the System Operates	Familiarize yourself with the Sun Grid Engine System components.

How Resources Are Matched to Requests	Learn how the Grid Engine system matches resources to requests, what jobs and queues are, and how usage policies assist in managing workload.
Choosing a User Interface	Learn about Sun Grid Engine's graphical user interface (QMON), the command-line interface, and the Distributed Resource Management Application API.
Users and User Categories	Learn about Sun Grid Engine's user categories: managers, operators, users, and owners.

Using the Wiki

Topic	Description
Using the Wiki	Learn how to monitor documentation changes, how to print the wiki, and and the Wiki FAQ.

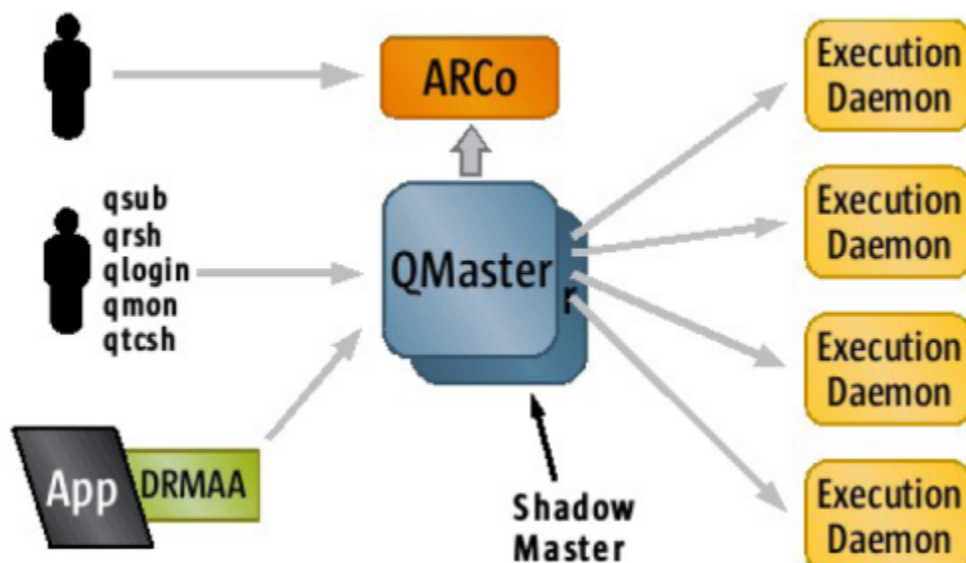
 To print this section, see the [Getting Started Guide \(Printable\)](#).



How the System Operates

The Grid Engine system does the following:

- Accepts jobs. Jobs are users' requests for computer resources. Each job includes a description of what to do and a set of property definitions that describe how the job should be run. Users can submit jobs via the command line interface or Grid Engine's graphical user interface, QMON. Users can also use the optional Distributed Resource Management Application API (DRMAA) to automate grid engine functions by writing scripts to submit and control jobs.
- Holds jobs. The Sun Grid Engine master daemon holds jobs until the needed compute resources become available.
- Sends jobs. When the compute resources become available, the master daemon sends the job to the appropriate execution host. The execution daemon on that host then executes the job.
- Manages running jobs. The master daemon manages running jobs. At a fixed interval, the master daemon receives reports from each execution daemon.
- Logs the record of job execution when the jobs are finished. The master daemon stores raw data. Users can also use the Accounting and Reporting Console (ARCo) to gather live reporting data from the Grid Engine system and to store the data for historical analysis in the reporting database, which is a standard SQL database.



Component	Description	More Info
Cluster	A collection of machines, called hosts, on which Grid Engine system functions occur.	See Configuring Clusters .
Master Host	The master host is central to cluster activity. The master host runs the master daemon and usually also runs the scheduler. The master host requires no further configuration other than that performed by the installation procedure. By default, the master host is also an administration host and a submit host.	For information about how to initially set up the master host, see How to Install the Master Host . For information about how to configure dynamic changes to the master host, see Configuring Hosts .
Master Daemon	The master daemon does the following: <ul style="list-style-type: none"> • Accepts incoming jobs from users. • Maintains tables about hosts, queues, jobs, system load, and user permissions. • Performs scheduling functions and requests actions from execution daemons on the appropriate execution hosts. • Decides which jobs are dispatched to which queues and how to reorder and reprioritize jobs to maintain share, priority, or deadline 	See Configuring Hosts .
Execution Host	Systems that have permission to run Grid Engine system jobs. These systems host queue instances, and run the execution daemon. Execution hosts are systems that have permission to execute jobs. Therefore, queue instances are attached to the execution hosts.	An execution host is initially set up by the installation procedure, as described in How to Install Execution Hosts . For installation planning guidance, see Host System Requirements . See Configuring Hosts for more information on managing your cluster.
Execution Daemon	The execution daemon receives jobs from the master daemon and executes them locally on its host. An execution daemon is responsible for the queue instances on its host and for the running of jobs in these queue instances. Periodically, the execution daemon forwards information such as job status or load on its host to the master daemon.	See Configuring Hosts .
Scheduler	The scheduler is responsible for prioritizing pending jobs and deciding which jobs to schedule to which resources.	For more information on the scheduler, see Managing the Scheduler .
Administration Host	Administration hosts are hosts that have permission to carry out any kind of administrative activity for the Grid Engine system.	See Configuring Hosts .
Submit Host	Submit hosts enable users to submit and control batch jobs only. In particular, a user who is logged in to a submit host can submit jobs with the <code>qsub</code> command, can monitor the job status with the <code>qstat</code> command, and can use the Grid Engine system OSF/1 Motif graphical user interface QMON, which is described in QMON, the Grid Engine System's Graphical User Interface .	See Configuring Hosts .
Shadow Master Host	Shadow master hosts reduce unplanned cluster downtime. One or more shadow master hosts may be running on additional nodes in a cluster. In the case that the master daemon or the host on which it is running fails, one of the shadow masters will promote the host on which it is running to the new master daemon system by locally starting a new master daemon.	See [How to Configure Shadow Master Hosts] .
DRMAA	The optional Distributed Resource Management Application API (DRMAA) automates Sun Grid Engine functions by writing scripts that run Sun Grid Engine commands and parse the results.	See Automating Grid Engine Functions Through DRMAA .
ARCo	The optional Accounting and Reporting Console (ARCo) enables you to gather live reporting data from the Grid Engine system and to store the data for historical analysis in the reporting database, which is a standard SQL database.	For more information, see Accounting and Reporting Console .

SDM	The optional Service Domain Manager (SDM) module distributes resources between different services according to configurable Service Level Agreements (SLAs). The SLAs are based on Service Level Objectives (SLOs). SDM functionality enables you to manage resources for all kind of scalable services.	See Service Domain Manager for more information.
-----	--	--



How Resources Are Matched to Requests

- [A Banking Analogy](#)
- [Usage Policies](#)

A Banking Analogy

As an analogy, imagine a large "money-center" bank in one of the world's capital cities. In the bank's lobby are dozens of customers waiting to be served. Each customer has different requirements. One customer wants to withdraw a small amount of money from his account. Arriving just after him is another customer, who has an appointment with one of the bank's investment specialists. She wants advice before she undertakes a complicated venture. Another customer in front of the first two customers wants to apply for a large loan, as do the eight customers in front of her.

Different customers with different needs require different types of service and different levels of service from the bank. Perhaps the bank on this particular day has many employees who can handle the one customer's simple withdrawal of money from his account. But at the same time the bank has only one or two loan officers available to help the many loan applicants. On another day, the situation might be reversed.

The effect is that customers must wait for service unnecessarily. Many of the customers could receive faster service if only their needs were immediately recognized and then matched to available resources.

If the Grid Engine system were the bank manager, the service would be organized differently:

- On entering the bank lobby, customers would be asked to declare their name, their affiliations, and their service needs.
- Each customer's time of arrival would be recorded.
- Based on the information that the customers provided in the lobby, the bank might serve the following customers in the following order:
 1. Customers whose needs match suitable and immediately available resources
 2. Customers whose requirements have the highest priority
 3. Customers who were waiting in the lobby for the longest time
- In a "Grid Engine system bank," one bank employee might be able to help several customers at the same time. The Grid Engine software would try to assign new customers to the least-loaded and most-suitable bank employee.
- As bank manager, the Grid Engine system would allow the bank to define service policies. Typical service policies might be the following:
 - To provide preferential service to commercial customers because those customers generate more profit
 - To make sure a certain customer group is served well, because those customers have received bad service in the past
 - To ensure that customers with an appointment get a timely response
 - To provide preferential treatment to certain customers because those customers were identified by a bank executive as high priority customers
- These policies would be implemented, monitored, and adjusted automatically by a Grid Engine system manager. Customers that have preferential access would be served sooner. Such customers would receive more attention from employees. The Grid Engine manager would recognize if the customers do not make progress. The manager would immediately respond by adjusting service levels to comply with the bank's service policies.

Jobs and Queues

In a Grid Engine system, jobs correspond to bank customers. Jobs wait in a computer holding area instead of a lobby. Queues, which provide services for jobs, correspond to bank employees. As in the case of bank customers, the requirements of each job, such as available memory, execution speed, available software licenses, and similar needs, can be very different. Only certain queues might be able to provide the corresponding service.

To continue the analogy, the Grid Engine software arbitrates available resources and job requirements in the following way:

- A user who submits a job through the Grid Engine software declares a requirement profile for the job. In addition, the software retrieves the identity of the user. The software also retrieves the user's affiliation with projects or user groups. The time that the user submitted the job is

also stored.

- The moment that a queue is available to run a new job, the Grid Engine software determines what are the suitable jobs for the queue. The software immediately dispatches the job that has either the highest priority or the longest waiting time.
- Queues allow concurrent execution of many jobs. The Grid Engine software tries to start new jobs in the least loaded and most suitable queue.

Usage Policies

The administrator of a cluster can define high-level usage policies that are customized according to the site. Four usage policies are available:

- Urgency – Using this policy, each job's priority is based on an urgency value. The urgency value is derived from the job's resource requirements, the job's deadline specification, and how long the job waits before it is run.
- Functional – Using this policy, an administrator can provide special treatment because of a user's or a job's affiliation with a certain user group, project, and so forth.
- Share-based – Under this policy, the level of service depends on an assigned share entitlement, the corresponding shares of other users and user groups, the past usage of resources by all users, and the current presence of users within the system.
- Override – This policy requires manual intervention by the cluster administrator, who modifies the automated policy implementation.

Policy management automatically controls the use of shared resources in the cluster to best achieve the goals of the administration. High priority jobs are dispatched preferentially. Such jobs receive higher CPU entitlements if the jobs compete for resources with other jobs. The Grid Engine software monitors the progress of all jobs and adjusts their relative priorities correspondingly and with respect to the goals defined in the policies.

Using Tickets to Administer Policies

The functional, share-based, and override policies are defined through a Grid Engine concept that is called tickets. You might compare tickets to shares of a public company's stock. The more shares of stock that you own, the more important you are to the company. If shareholder A owns twice as many shares as shareholder B, A also has twice the votes of B. Therefore shareholder A is twice as important to the company. Similarly, the more tickets that a job has, the more important the job is. If job A has twice the tickets of job B, job A is entitled to twice the resource usage of job B.

Jobs can retrieve tickets from the functional, share-based, and override policies. The total number of tickets, as well as the number retrieved from each ticket policy, often changes over time.

The administrator controls the number of tickets that are allocated to each ticket policy in total. Just as ticket allocation does for jobs, this allocation determines the relative importance of the ticket policies among each other. Through the ticket pool that is assigned to particular ticket policies, the Grid Engine software can run in different ways. For example, the software can run in a share-based mode only. Or the software can run in a combination of modes, for example, 90% share-based and 10% functional.

Using the Urgency Policy to Assign Job Priority

The urgency policy can be used in combination with two other job priority specifications:

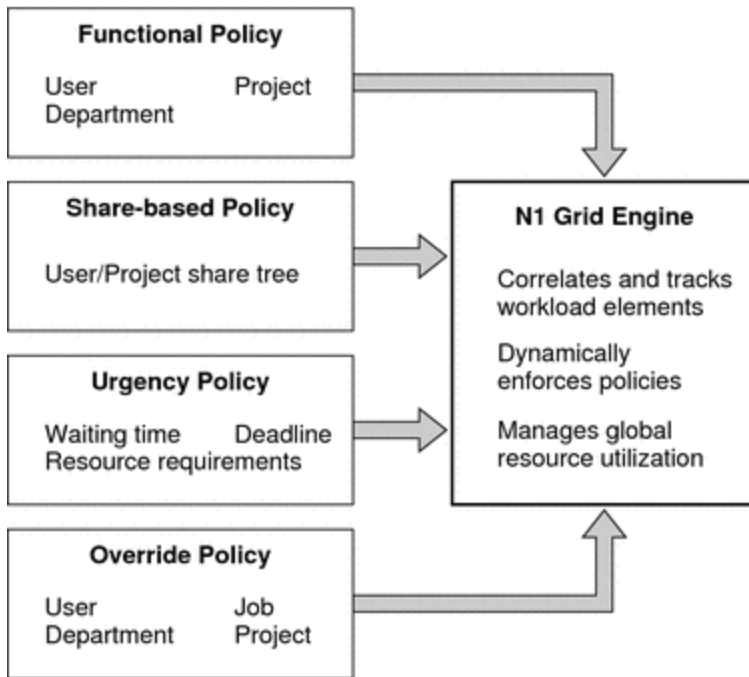
- The number of tickets assigned by the functional, share-based, and override policies
- A priority value specified by the `qsub -p` command

A job can be assigned an urgency value, which is derived from three sources:

- The job's resource requirements
- The length of time that a job must wait before the job runs
- The time at which a job must finish running

The administrator can separately weight the importance of each of these sources to arrive at a job's overall urgency value. For more information, see [Managing Policies](#).

The following figure shows the correlation among policies in a Grid Engine system.



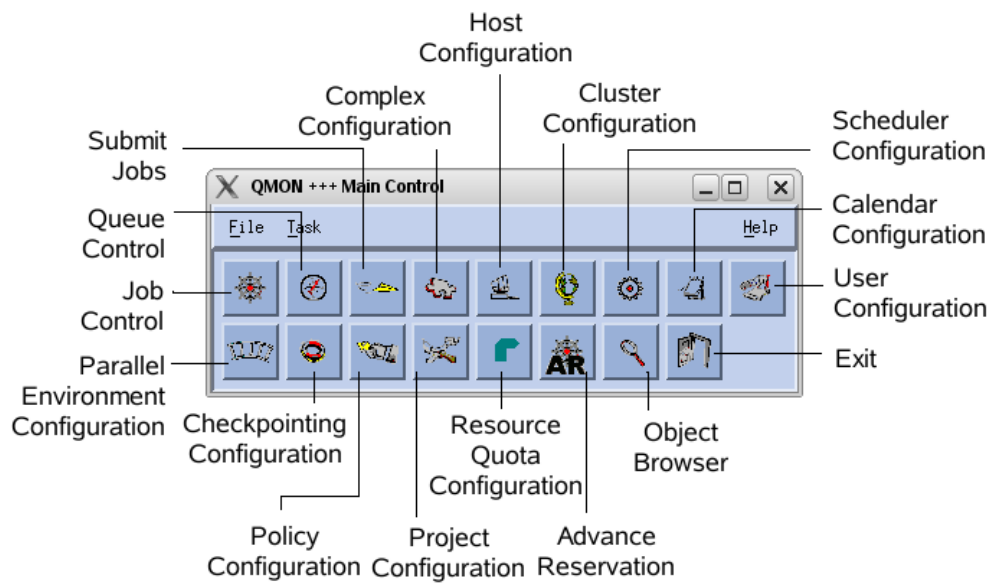
Choosing a User Interface

To meet the needs of your environment, the following interface tools are available:

- [QMON - The Graphical User Interface](#)
- [The Command Line Interface](#)
- [The Distributed Resource Management Application API \(DRMAA\)](#)

QMON - The Graphical User Interface

If you prefer using a graphical user interface, you can use QMON to accomplish most Grid Engine system tasks. The QMON Main Control window, which is shown below, is often the starting point for user and administrator functions.



For more information on QMON if you are an administrator, see [Interacting With Sun Grid Engine as an Administrator](#).
 For more information on QMON if you are a user, see [Interacting With Sun Grid Engine as a User](#).

The Command Line Interface

If you prefer using the command line, the command line user interface includes a flexible set of ancillary programs (commands) that enable you to interact with the Sun Grid Engine system.

For more information on the command line if you are an administrator, see [Interacting With Sun Grid Engine as an Administrator](#).
 For more information on the command line if you are a user, see [Interacting With Sun Grid Engine as a User](#).

For information on the ancillary programs that Sun Grid Engine provides and which users have access to these commands, see [Command Line Interface Ancillary Programs](#).

The Distributed Resource Management Application API (DRMAA)

You can automate Sun Grid Engine functions by writing scripts that run Sun Grid Engine commands and parse the results. However, for more consistent and efficient results, you can use the Distributed Resource Management Application API (DRMAA). For more information about the DRMAA concept and how to use it with the C and Java™ languages, see [Automating Grid Engine Functions Through the Distributed Resource Management Application API](#).



Command Line Interface Ancillary Programs

- [List of Ancillary Programs](#)
- [User Access to the Ancillary Programs](#)

For more information on available ancillary programs, see the [Grid Engine Man Pages](#).

List of Ancillary Programs

The Grid Engine system provides the following set of ancillary programs:

Program	Description
---------	-------------

qacct	Extracts arbitrary accounting information from the cluster log file. For more information, see Generating Accounting Statistics .
qalter	Changes the attributes of submitted but pending jobs.
qconf	Provides the user interface for cluster configuration and queue configuration. For more information, see Using qconf .
qdel	Enables a user to delete one or more jobs. A manager or operator can delete jobs belonging to any user, while regular users can only delete their own jobs. For more information, see How to Control Jobs .
qhold	Holds back submitted jobs from execution.
qhost	Displays status information about execution hosts.
qlogin	Initiates a login session with automatic selection of a low-loaded, suitable host.
qmake	A replacement for the standard UNIX <code>make</code> facility. <code>qmake</code> extends <code>make</code> by its ability to distribute independent <code>make</code> steps across a cluster of suitable machines. For more information, see Parallel Makefile Processing With qmake .
qmod	Enables the owner to suspend or enable a queue. All currently active processes that are associated with this queue are also signaled. For more information, see How to Monitor and Control Queues and How to Control Jobs .
qmon	Provides an X Windows Motif command interface and monitoring facility.
qping	Checks application status of Sun Grid Engine daemons.
qquota	Shows current usage of Sun Grid Engine resource quotas. For more information, see How to Monitor Resource Quota Utilization From the Command Line .
qrdel	Deletes Sun Grid Engine advance reservations. For more information, see How to Configure Advance Reservations From the Command Line .
qresub	Creates new jobs by copying jobs that are running or pending.
qrls	Releases jobs from holds that were previously assigned to them, for example, through <code>qhold</code> .
qrsh	Can be used for various purposes, such as the following: <ul style="list-style-type: none"> • To provide remote execution of interactive applications through the Grid Engine system. <code>qrsh</code> is comparable to the standard UNIX facility <code>rsh</code>. For more information, see Remote Execution With qrsh. • To allow for the submission of batch jobs that, upon execution, support terminal I/O and terminal control. Terminal I/O includes standard output, standard error, and standard input. • To provide a submission client that remains active until the batch job finishes. • To allow for the Grid Engine software-controlled remote execution of the tasks of parallel jobs.
qrstat	Shows the status of Sun Grid Engine advance reservations. For more information, see How to Configure Advance Reservations From the Command Line .
qrsb	Submits an advance reservation to Sun Grid Engine. For more information, see How to Configure Advance Reservations From the Command Line .
qselect	Prints a list of queue names corresponding to specified selection criteria. The output of <code>qselect</code> is usually sent to other Grid Engine system commands to apply actions on a selected set of queues.
qsh	Opens an interactive shell in an <code>xterm</code> on a lightly loaded host. Any kind of interactive jobs can be run in this shell. For more information, see How to Submit Interactive Jobs From the Command Line .
qstat	Provides a status listing of all jobs and queues associated with the cluster. For more information, see How to Monitor Jobs From the Command Line .
qsub	The user interface for submitting batch jobs to the Grid Engine system.
qtcs	A fully compatible replacement for the widely known and used UNIX C shell (<code>csh</code>) derivative, <code>tcsh</code> . <code>qtcs</code> provides a command shell with the extension of transparently distributing execution of designated applications to suitable and lightly loaded hosts through Grid Engine software. For more information see, Transparent Job Distribution With qtcs .

User Access to the Ancillary Programs

The following table shows the command capabilities that are available to the different user categories:

Command	Manager	Operator	Owner	User
qacct	Full	Full	Own jobs only	Own jobs only
qalter	Full	Full	Own jobs only	Own jobs only
qconf	Full	No system setup modifications	Show only configurations and access permissions	Show only configurations and access permissions
qdel	Full	Full	Own jobs only	Own jobs only
qhold	Full	Full	Own jobs only	Own jobs only
qhost	Full	Full	Full	Full
qlogin	Full	Full	Full	Full
qmod	Full	Full	Own jobs and owned queues only	Own jobs only
qmon	Full	No system setup modifications	No configuration changes	No configuration changes
qrexec	Full	Full	Full	Full
qselect	Full	Full	Full	Full
qsh	Full	Full	Full	Full
qstat	Full	Full	Full	Full
qsub	Full	Full	Full	Full



Users and User Categories

There are four categories of users that each have access to their own set of Grid Engine system commands:

- **Managers** – Managers have full capabilities to manipulate the Grid Engine system. By default, the superusers of all administration hosts have manager privileges.
- **Operators** – Users who can perform the same commands as managers except that they cannot change the configuration. Operators are supposed to maintain operation.
- **Users** – People who can submit jobs to the grid and run them if they have a valid login ID on at least one submit host and one execution host. Users have no cluster management or queue management capabilities.
- **Owners** – Users who can suspend or resume and disable or enable the queues they own. Typically, users are owners of the queue instances that reside on their workstations. Queue owners can be managers, operators, or users. Users are commonly declared to be owners of the queue instances that reside on their desktop workstations. See [How to Configure Owners Parameters With QMON](#) for more information.

For information on which command capabilities are available to the different user categories, see [Command Line Interface Ancillary Programs](#).

Using the Wiki

Why a Wiki?

Since the release of Sun Grid Engine 6.2, all documentation for the product can be found on wikis.sun.com. This was done for the following reasons:

- To keep the documentation as up-to-date as possible. The product team and select community members can edit the information in real time, which ensures that the information will stay as current as possible.
- To encourage community input. Anyone can comment on the documentation. Let us know what you think.
- To build a library of community contributions. The simple wiki markup language makes it easy for anyone in the Grid Engine community to add their own input to the wiki. See the [Expert Advice](#) section to add your expertise or to survey current contributions.

Wiki Tasks

Topic	Description
How to Monitor Documentation Changes	Learn how to monitor documentation changes using email notifications or RSS feeds.
How to Add a Comment	Learn how to add a comment to any page in the Sun Grid Engine wiki space.
How to Add a Page	Learn how to add pages to the Expert Advice section.
How to Print	Learn how to print what you need from the Sun Grid Engine wiki.

Questions?

Do you have any questions about how to use the wiki? Please comment on this page with your question or send us an [email](#).

How to Add a Comment

1. Log in to wikis.sun.com.
2. On the bottom of the page on which you would like to comment, Click Add Comment.
The comment box is displayed.
3. Enter your comment.
4. Click Post to save and publish your comment.
Click Cancel to close the comment box without saving your comment.

The page How to Add a Page does not exist.

How to Monitor Documentation Changes

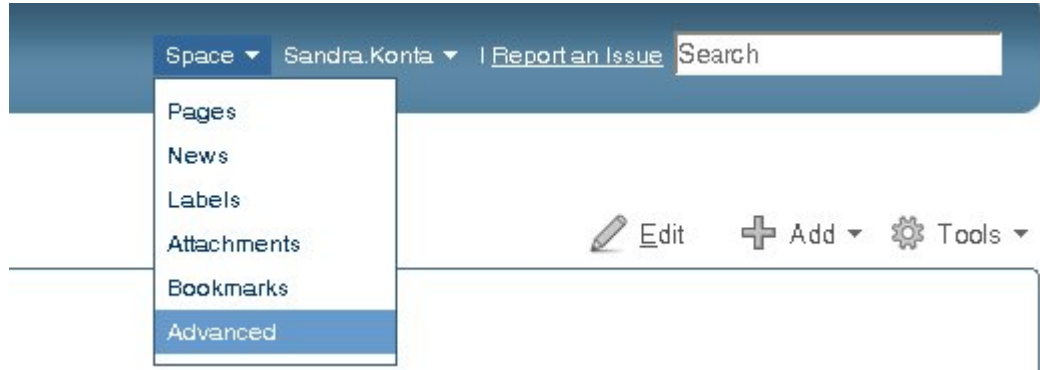


If you would like to watch a parent page and all of its child pages, you must set watches individually on the parent page and each of its child pages.

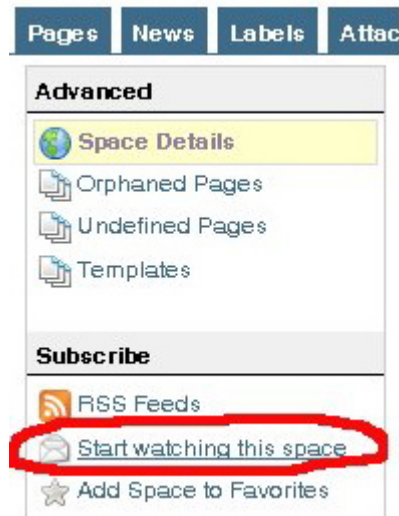
Function	Procedure
----------	-----------

Setting up email notifications for the grid engine space

1. On the top, right hand side of the page, click Space and then click Advanced.



2. On the left hand side of the page, click on "Start watching this space."



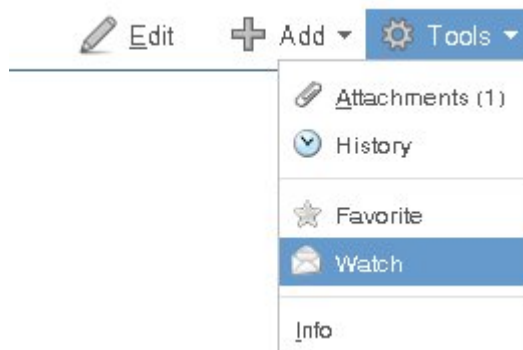
The string will immediately change to "Stop watching this space."

Removing email notifications for the grid engine space

Follow the above directions but instead of clicking "Start watching this space," click "Stop watching this space."

Setting up email notifications for a page

1. Click Tools, on the top right hand side of the page, and then click Watch.



The envelope icon will change colors to indicate the watch has been set.

Removing email notifications for a page

Click Tools and then click Watch. The envelope icon will change back to white to indicate the watch has been removed.

Review your email notifications	<ol style="list-style-type: none"> 1. Click on your name on the top of the page and then click Preferences on the drop-down menu. 2. Click the Watches tab.
Setting up RSS notifications of documentation changes	<ol style="list-style-type: none"> 1. On the top, right hand side of the page, click Space and then click Advanced. 2. On the left hand side of the page, click on RSS Feeds. You can then choose to receive RSS notification for the following: <ul style="list-style-type: none"> • Pages • News • Mail (This feature is currently disabled.) • Comments • Attachments • All Content 3. Click on Pages and then select your favorite aggregator.



How to Print

You can print a specific page, a section, or a PDF copy of the entire wiki. See below for guidance on how to print what you need.

How to Print a Page

If you would like to print a specific page, do the following:

1. Click on Tools in the upper left hand corner of the page that you would like to print.
2. Select either 'Export to PDF' or 'Export to Microsoft Word.'



3. Print from the application that you selected.

How to Print a Section of the Documentation

If you would like to print one of the major sections, select one of the following printables and then print directly from your browser:

- [Getting Started Guide \(Printable\)](#)
- [Planning the Installation Guide \(Printable\)](#)
- [Installing Guide \(Printable\)](#)
- [Upgrading Guide \(Printable\)](#)
- [Administering Guide \(Printable\)](#)
- [Using Guide \(Printable\)](#)
- [Accounting and Reporting Console Guide \(Printable\)](#)
- [Inspect Guide \(Printable\)](#)
- [Service Domain Manager Guide \(Printable\)](#)

How to Print the Entire Documentation Set

For PDF copy of the entire documentation set, click [here](#).



This PDF snapshot was taken on March 3, 2009, the day that Grid Engine 6.2u2 was released.