Introducing NQE

IN–2153 3.3

Document Number 007–3807–001

Record of Revision

Version	*Description*
1.0	December 1993. Original Printing. This manual introduces the Network Queuing Environment (NQE). This manual supplements complete usage and feature descriptions contained in the online guides available with NQE.
1.1	June 1994. Incorporates information for NQE release 1.1.
2.0	May 1995. Incorporates information for NQE release 2.0.
3.0	March 1996. Incorporates information for the NQE 3.0 release.
3.1	September 1996. Incorporates information for the NQE 3.1 release.
2/97	February 1997. Provides introductory information for using NQE.
3.3	March 1998. Provides introductory information for using NQE and incorporates information for the NQE 3.3 release.

Contents

Page

Page

Figures

Preface

This publication describes the Cray Network Queuing Environment (NQE). NQE is a set of clients and servers that lets users submit, monitor, and control work across a load-balanced network of hosts. NQE supports computing in a large network; up to 36 servers and hundreds of clients using the NQE database model, or an unlimited number of NQS servers. This grouping of servers and clients is referred to as an NQE *cluster*.

NQE consists of the following components that provide a seamless environment for client users to do their work:

- An *NQE client* provides the client user interfaces to NQE. It supports the submission, monitoring, and control of work from the workstation for job execution of the batch request on the nodes. It provides the NQE graphical user interface (GUI) that lets users submit batch requests to a central storage database, to control these requests, and to obtain status on their batch requests and file transfers.
- The Network Queuing System (NQS) initiates requests on NQS servers. An *NQS server* is the host on which NQS runs.
- The *Network Load Balancer (NLB)* provides status and control of work scheduling within the group of components in the NQE cluster. The NLB offers NQS a list of servers, in order of preference, to run a request; NQS uses the list to route the request.
- The *NQE database* provides an alternative mechanism for distributing work. Requests are submitted and stored centrally. The NQE scheduler examines each request and determines when and where the request is run.
- The *File Transfer Agent (FTA)* provides asynchronous and synchronous file transfer. You can queue your transfers so that they are retried if a network link fails.

This introduction describes basic user tasks, provides instructions on how to get help online, and provides troubleshooting information.

Related Publications

The following documents contain additional information that may be helpful (this introductory guide is also provided online):

- *NQE User's Guide*, publication SG–2148, provides detailed user-level information, such as how to perform user tasks by using either the NQE GUI or the command line interface, and how to customize your environment.
- *NQE Administration*, publication SG–2150, provides information on configuring, monitoring, and controlling NQE.
- *NQE Installation*, publication SG–5236, describes how to install or upgrade the NQE software.
- *NQE Release Overview*, publication RO–5237, provides NQE release information.

These publications can also be accessed online by using the Cray DynaWeb server application and through the Silicon Graphics Technical Publications Library World Wide Web page at the following URL:

`http://techpubs.sgi.com/library/`

For further information, see Section 3.1, page 33.

Ordering Cray Research Publications

The *User Publications Catalog*, publication CP–0099, describes the availability and content of all Cray Research hardware and software documents that are available to customers. Cray Research customers who subscribe to the Cray Inform (CRInform) program can access this information on the CRInform system.

To order a document, either call the Distribution Center in Mendota Heights, Minnesota, at +1–612–683–5907, or send a facsimile of your request to fax number +1–612–452–0141. Cray Research employees may send electronic mail to `orderdsk` (UNIX system users).

Customers who subscribe to the CRInform program can order software release packages electronically by using the `Order Cray Software` option.

Customers outside of the United States and Canada should contact their local service organization for ordering and documentation information.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names. The following list describes the identifiers:

1	User commands
1B	User commands ported from BSD
2	System calls
3	Library routines, macros, and opdefs
4	Devices (special files)
4P	Protocols
5	File formats
7	Miscellaneous topics
7D	DWB-related information
8	Administrator commands

Some internal routines (for example, the `_assign_asgcmd_info`() routine) do not have man pages associated with them.

variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.
[ ]	Brackets enclose optional portions of a command line.

The default shell in the UNICOS and UNICOS/mk operating systems, referred to in Cray Research documentation as the *standard shell*, is a version of the Korn shell that conforms to the following standards:

- Institute of Electrical and Electronics Engineers (IEEE) Portable Operating System Interface (POSIX) Standard 1003.2–1992
- X/Open Portability Guide, Issue 4 (XPG4)

The UNICOS and UNICOS/mk operating systems also support the optional use of the C shell.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, please tell us. You can contact us in any of the following ways:

- Send us electronic mail at the following address:

  `techpub@sgi.com`

- Contact your customer service representative and ask that an SPR or PV be filed. If filing an SPR, use `PUBLICATIONS` for the group name, `PUBS` for the command, and `NO-LICENSE` for the release name.

- Call our Software Publications Group in Eagan, Minnesota, through the Customer Service Call Center, using either of the following numbers:

  1–800–950–2729 (toll free from the United States and Canada)

  +1–612–683–5600

- Send a facsimile of your comments to the attention of "Software Publications Group" in Eagan, Minnesota, at fax number +1–612–683–5599.

We value your comments and will respond to them promptly.

Overview [1]

The Cray Network Queuing Environment (NQE) lets you submit batch requests to execute across a load-balanced network of hosts. NQE supports computing in a large network; up to 36 servers and hundreds of clients using the NQE database model, or an unlimited number of NQS servers. This grouping of servers and clients is referred to as an NQE *cluster*. *Batch requests* are shell scripts that are executed independently from an interactive terminal session.

You submit requests from NQE clients and they are executed at NQS server nodes. You also can log on to nodes and submit requests. You can monitor and control the progress of a batch request through the NQE components in the NQE cluster.

This chapter describes the following:

- NQE components and NQE cluster components
- Tasks you can perform using NQE

1.1 NQE Components and NQE Cluster Components

NQE includes the following components:

- The *NQE client* provides the client user interface to NQE. It supports the submission, monitoring, and control of work from the workstation for job execution of the batch request on the nodes. NQE clients are intended to run on every node in the NQE cluster where users need an interactive interface to the NQE cluster. It provides the NQE graphical user interface (GUI) (accessed through the `nqe`(1) command) and a command line interface. The interfaces are documented in the *NQE User's Guide*, publication SG–2148, and the NQE GUI has context-sensitive help.

  For information about using the NQE GUI, see Section 2.4, page 11.

- The Network Queuing System (NQS) initiates requests on NQS servers. An *NQS server* is the host on which NQS runs. Your default NQS server is designated by your system administrator and is specified in the NQE configuration file (`nqeinfo`); you can submit your request to a specific NQS server by using the NQE GUI `Config` window or by setting the `NQS_SERVER` environment variable, which overrides the default value of `NQS_SERVER` defined by your system administrator.

- The *Network Load Balancer (NLB)* provides status and control of work scheduling within the group of components in the NQE cluster. This information is then used to load balance batch requests across NQS servers in the NQE cluster. The NLB offers NQS a list of servers, in order of preference, to run a request; NQS uses the list to route the request.
- The *NQE database* provides a central repository for batch requests in the NQE cluster. The NQE scheduler uses the NQE database and an alternative mechanism for distributing work. The NQE scheduler examines each request and determines when and on which execution node the request will run. The NQE database lightweight server (LWS) verifies validation, submits the copy of a request that is in the NQE database to NQS, and obtains the exit status of completed requests from NQS.
- The *File Transfer Agent (FTA)* provides asynchronous and synchronous file transfer. You can queue your transfers so that they are retried if a network link fails.

Note: Your system administrator can tell you if all the NQE components, or only the NQS and FTA components, are available to you.

The NQE cluster can contain the following components:

- The Network Load Balancer (NLB) server, which receives and stores information from the NLB collectors in the NLB database that it manages.
- The NQE database server, which serves connections from clients, the scheduler, the monitor and lightweight server (LWS) components in the cluster to add, modify, or remove data from the NQE database. Currently, NQE uses the mSQL database.
- The NQE scheduler, which analyzes data in the NQE database and makes scheduling decisions.
- The NQE database monitor, which monitors the state of the database and to which the NQE database components are connected.
- NQE *clients* (running on numerous machines) contain software so users can submit, monitor, and control requests by using either the NQE GUI or the command line interface. From clients, users also can monitor request status, delete or signal requests, monitor machine load, and receive request output using the FTA.

The machines in your network where you run NQS are usually machines that have a large execution capacity. Job requests can be submitted from components in an NQE cluster, but they will only be initiated on an NQS server node.

FTA can be used from any NQS server node to transfer data to and from any node in the network by using the `ftpd` daemon. It also can provide file transfer by communicating with `ftad` daemons that incorporate network peer-to-peer authorization, which is a more secure method than `ftp`.

On NQS server nodes, you need to run a collector process to gather information about the machine for load balancing and request status for the NQE GUI `Status` and `Load` windows programs. The collector forwards this data to the NLB server.

The NLB server runs on one or more NQE nodes in a cluster, but it is easiest to run it initially on the first node where you install NQE. Redundant NLB servers ensure that the NLB database has a greater availability if an NLB server cannot be reached through the cluster.

Note: The NQE database must be on only one NQE node; there is no redundancy.

1.2 Tasks You Can Perform Using NQE

NQE lets you do the following tasks:

- Submit batch requests to a node in the NQE cluster that is running NQS.
- Route requests automatically without knowing the names of queues or hosts.
- Monitor the status of requests across the NQE cluster. Request status is refreshed at configurable intervals.
- Signal a request in the node cluster.
- Delete a request in the node cluster.
- Allow independent jobs to communicate with each other by using the `cevent`(1) command.
- Communicate with other UNIX systems running public domain versions of NQS.
- Monitor the machine load across the NQE cluster.
- Recover and restart your requests (see your *NQE User's Guide*, publication SG–2148, to determine which systems support this capability).
- Transfer files both interactively and from within batch requests on NQE servers without specifying a password.

- Queue file transfers so that you can go on to other work.
- Automatically retry file transfers that encounter network failures.
- Customize your NQE environment.

Getting Started [2]

This chapter provides the following information about NQE:

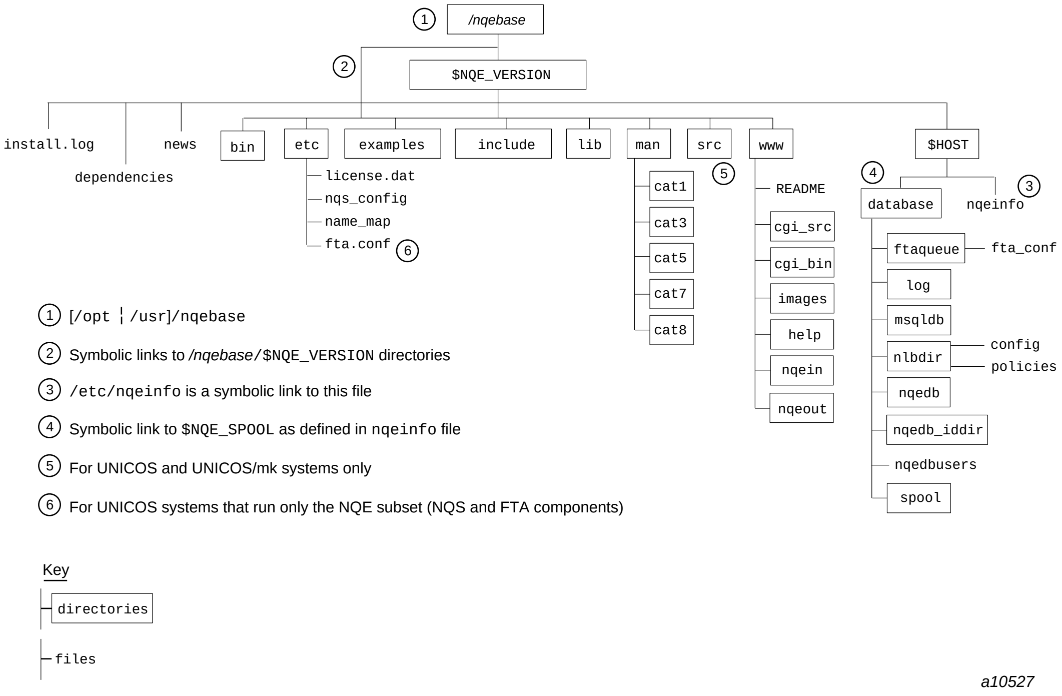
- NQE file structure
- Setting environment variables
- Setting up authorization
- User interfaces
- Creating and submitting a batch request
- Monitoring a request
- Deleting a request
- Monitoring machine load
- Recovery and restart

This introductory guide uses your default configuration (for information about changing your configuration or about using the `Config` window, see the *NQE User's Guide*, publication SG–2148). Check with your system administrator to see if system and configuration changes have been made, as this may also change the default actions described in this guide.

2.1 NQE File Structure

Throughout this guide, the path /*nqebase* is used in place of the default NQE path name, which is `/usr/craysoft/nqe` on all systems except on Solaris, UNICOS, and UNICOS/mk systems, where it is `/opt/craysoft/nqe`.

Figure 1 shows the NQE file structure.



1 [/opt ¦ /usr]/nqebase

2 Symbolic links to */nqebase/*$NQE_VERSION directories

3 /etc/nqeinfo is a symbolic link to this file

4 Symbolic link to $NQE_SPOOL as defined in nqeinfo file

5 For UNICOS and UNICOS/mk systems only

6 For UNICOS systems that run only the NQE subset (NQS and FTA components)

Figure 1. NQE File Structure

2.2 Setting Environment Variables

To use NQE, you must set the following environment variables:

- DISPLAY must be set to *local_workstation_name* :0 for the NQE graphical user interface (GUI) to work.

  Note: If your site has access control in place for using X Window System applications, contact your system administrator to determine if you need additional settings.

- PATH must include the path name of the NQE commands. The default path name is as follows: /*nqebase*/bin. System administrators also must include /*nqebase*/etc in their PATH environment variable to use certain NQE administrator commands.

- MANPATH must include the path name of the NQE man pages. The default name is /*nqebase*/man.

To verify whether your site's path names are the NQE system default, use the following command:

```
cd /nqebase/bin
```

If this command is not successful, ask your system administrator where the NQE software is located and add those directories to your `PATH` and `MANPATH` environment variables.

The commands that you use to set the environment variables depend on the shell that you use. The standard shell (or Korn shell) is the default shell for UNICOS and UNICOS/mk systems. (See the "Preface" for more information about the standard shell.)

For a list of other NQE environment variables that you can set to customize your environment, see the *NQE User's Guide*, publication SG–2148. To access the manual online, see Section 3.1, page 33.

The following example uses `sh` syntax to set and display NQE environment variables:

```
# PATH=$PATH:/nqebase/bin:/nqebase/etc; export PATH
# MANPATH=:/nqebase/man; export MANPATH
# DISPLAY=snow32:0; export DISPLAY
# env
LOGNAME=you
MAIL=/var/mail/you
USER=you
SHELL=/bin/sh
PWD=/home/snow32/you
MANPATH=:/nqebase/man
PATH=/usr/bin::/nqebase/bin:/nqebase/etc
DISPLAY=snow32:0
```

The following example uses `csh` syntax to set and display the NQE environment variables:

```
% setenv PATH /nqebase/bin:/nqebase/etc:$PATH
% setenv MANPATH /nqebase/man:$MANPATH
% setenv DISPLAY snow32:0
% env
HOME=/home/snow32/you
SHELL=/bin/csh
TERM=xterm
USER=you
LOGNAME=you
PWD=/home/snow32/you
MANPATH=/usr/man:/opt/local/man:/nqebase/man
PATH=/usr/bin:/nqebase/bin:/nqebase/etc
DISPLAY=snow32:0
```

2.3 Setting up Authorization

To submit requests to the NQE database and to execute requests on the NQS server, you must have the proper authorization, as described in the following sections.

2.3.1 NQE Database Authorization

To submit or control a request, or to get the status of a request in the NQE database, you must have a database user (`dbuser`) account with the proper authorization (user privileges). This database user account name can be the same as or different from your login (user name) on the client host; this introductory guide assumes you are using the same login across the NQE network. Your NQE administrator controls who has access to the database and from which host.

2.3.2 NQS Validation to Execute Batch Requests

By default, NQS uses file validation to authorize users. NQS can also be configured to use password validation or both file and password validation. Your NQE administrator specifies the validation method used in your NQS configuration.

NQS will always try to use your `.nqshosts` file and will use the `.rhosts` file only if the `.nqshosts` file does not exist. You should use `.rhosts` files unless your system administrator tells you that you should not use them.

If your site uses validation files, you must have a .rhosts or .nqshosts file in your home directory on each NQS node in the cluster that might process your request. NQS uses these files to authorize your user name before it sends your request to a batch queue.

If your site uses aliases for host names, you must also include those names in the .rhosts or .nqshosts file entry. For example, you may have to include ice.site.com rather than simply ice.

> **Note:** If your site uses password validation, you do not need validation files on the servers. If your site uses password validation, you must supply a password each time you submit, monitor, delete, or send a signal to a request.

2.3.3 .rhosts Validation Files Example

The following example shows how user jane would set up her .rhosts files to use the NQE database in a multiple-node NQE cluster.

> **Note:** For examples of how to set up .rhosts files so you can submit requests directly to NQS or to use alternative user names, see the *NQE User's Guide*, publication SG–2148. To access the manual online, see Section 3.1, page 33.

In this example, user jane is using the NQE GUI or NQE client commands on workstation snow, and her NQE node is wind. The NQE database resides on wind. User jane wants to submit requests to the NQE database on wind, where the NQE scheduler will select a target system to execute her request. To submit requests to the NQE database on wind, jane must have authentication on the NQE database (see Section 2.3.1, page 8).

In this example, the NQE cluster includes three other NQE nodes that have the names gust, storm, and rain. This means that jane can potentially run requests on rain, gust, storm, and wind. To use all four nodes, jane must have a .rhosts (or .nqshosts) file in her home directory on each of the four nodes.

Since jane uses the same login ID on all of the NQE nodes, the .rhosts file on snow would contain the following entries:

- On the NQE client workstation snow:

  rain jane (allows output to be returned to snow)

  gust jane (allows output to be returned to snow) `storm jane` (allows output to be returned to `snow`)

`wind jane` (allows output to be returned to `snow`)

> **Note:** This is required only if `rcp` is the output mechanism. Also, these entries need to be in the `.rhosts` file only; `rcp` does not use the `.nqshosts` file.

`jane`'s `.rhosts` file on each NQE node must contain the following entry (note that it is the same entry on each NQE node):

- On the NQE node `wind`:

  `snow jane` (permits incoming requests from `jane` on `snow`)

- On the NQE node `gust`:

  `snow jane` (permits incoming requests from `jane` on `snow`)

- On the NQE node `rain`:

  `snow jane` (permits incoming requests from `jane` on `snow`)

- On the NQE node `storm`:

  `snow jane` (permits incoming requests from `jane` on `snow`)

Once she has her `.rhosts` files in place, `jane` can submit requests to `wind`.

By default, job output will be returned from the NQS server to the NQE client, `snow`, by either FTA or `rcp`. FTA will be attempted first.

FTA may require a password for the destination host (NQE client `snow`) and user (user `jane` at `snow`), which can be supplied by creating a `.netrc` file in the login directory of user `jane` on `wind` (`~jane/.netrc` on `wind`), if it is executed there. (For further information, see the `netrc`(5) man page).

FTA can also be configured so that `jane` does not have to supply a password. This configuration in FTA is called network peer-to-peer authorization (NPPA). Check with your system administrator to see if this option is available.

`rcp` is used to return output only if FTA fails immediately to return the output. `rcp` requires a `.rhosts` file at the destination host in the recipient's home directory (`~jane/.rhosts` on `snow`).

If you have validation files on all of the nodes in the network on which your requests can run, and you still receive authorization failure messages, you should read the chapters about authorization in the *NQE User's Guide*, publication SG–2148. To access the manual online, see Section 3.1, page 33.

2.4 User Interfaces

You can use either the NQE graphical user interface (GUI) or the command line interface when using NQE. The following sections provide a brief overview of these functions. (You can also submit your request by using a World Wide Web (WWW) interface; for further information, ask your system administrator.)

Before you can use the NQE commands, you must add the NQE `/`*nqebase*`/bin` directory (and the `/`*nqebase*`/etc` directory to use administrator commands) to your search path. Before you can use the man pages (which tell you about the NQE commands and command options), you must add the NQE `/`*nqebase*`/man` directories to your search path. For a description of how to set these variables, see Section 2.2, page 6.

2.4.1 NQE Graphical User Interface

The NQE GUI is similar to a Motif interface. To access the NQE GUI, execute the `nqe`(1) command. Figure 2, shows the initial (top-level) NQE GUI button bar window that will appear. Each button (except `Exit` and `Help`) opens a window.

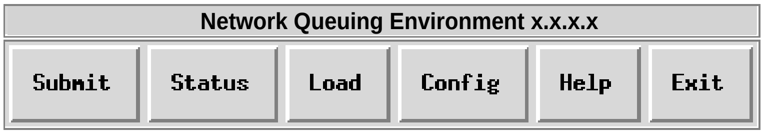


Figure 2. Initial NQE GUI Button Bar Window

You can use the NQE GUI for the following tasks:

- Use the `Submit` window to do the following:
  - Open and edit a job script
  - Save changes made to a job script
  - Submit a request to NQE
  - Launch a request on a periodic basis

  - From within the `Submit` window, reset your configuration preferences for the request you are submitting
  - View, segment, delete, or reset your NQE GUI log
  - Set or unset your password
  - Configure and save your job-related options (job profile)

- Use the `Status` window to do the following:
  - View updated status of your requests (the window is refreshed periodically)
  - View updated status of your FTA file transfers (the window is refreshed periodically)
  - Delete a request
  - Send signal to a request
  - View the detailed status of a request
  - Set or unset your password

  Context-sensitive help is displayed as you glide your mouse pointer over a menu or field name in the `Status` window; a brief description of the menu or field appears at the bottom of the display.

- Use the `Load` window to do the following:
  - Display continually updated system load information for machines in the NQE cluster, organized by host or by type of data
  - Display data about a specific host
  - Display formulae used to calculate each chart
  - Create, configure, add, and remove a chart

- Use the `Config` window to do the following:
  - Set your preferences for the following: default job profile; temporary directory; job script; job output; and NQE GUI log directories.
  - View the current settings for your preferences

- To display the current NQE version number and copyright information in the `Submit`, `Status`, and `Config` windows, use the left mouse button and click once on the Cray Research logo button.
- To access online help, use the left mouse button and click once on the `Help` button. For detailed information about using the NQE GUI and descriptions of menu options, select the `Help` button or see the *NQE User's Guide*, publication SG–2148. To access the manual online, see Section 3.1, page 33. For a summary of the NQE GUI displays and functions, see the `nqe`(1) man page.
- To exit the NQE GUI, use the left mouse button and click once on the `Exit` button.

When the mouse pointer is within a display area of a specific NQE GUI window, you can use the `ALT` key and the underscored letter from the menu bar to pop up submenus and to select more submenu options. An alternative way to do this is to use the `F10` key to activate the menu bar and then use the cursor movement keys to select submenus and options.

2.4.2 Command Line Interface

NQE provides a command line interface for user functions. Each command is documented on a man page (man pages are provided in online form only). See the *NQE User's Guide*, publication SG–2148, for more information on the user commands.

You can issue the following commands from any NQE node because all NQE nodes contain the NQE client software:

Command	Description
`cevent`	Posts, reads, and deletes job-dependency event information.
`cqdel`	Signals a request that is either running or queued.
`cqstatl`	Displays the status of NQE work through a line-mode, static display.
`cqsub`	Submits a script file to NQE for execution.
`ilb`	Executes a command on a machine chosen by the Network Load Balancer (NLB).

You can issue the following commands only at an NQE node that has installed the NQE components; if you issue them from an NQE client, they have no effect. The following commands are not installed on NQE clients; they do not recognize the `NQS_SERVER` environment variable:

Command	Description
`ftua`	Transfers a file interactively.
`qalter`	Alters the attributes of one or more NQS requests.
`qchkpnt`	Checkpoints an NQS request; this command is not available on all systems running NQE.
`qdel`	Deletes or signals an NQS request.
`qlimit`	Displays NQS batch limits for the local host; this command also displays the `qsub` command options that are used to specify each resource at the time of submission.
`qmsg`	Writes messages to `stderr`, `stdout`, or the job log file of an NQS batch request.
`qping`	Determines whether the local NQS daemon is running and responding to requests.
`qstat`	Displays the status of NQS queues, requests, and queue complexes.
`qsub`	Submits a batch request to NQS.
`rft`	Transfers a file in a batch request.

2.5 Creating and Submitting a Batch Request

A *batch request* is a series of commands, usually contained in a job script file that you create using a text editor (such as `vi`). A job script can be as simple as one command, such as the following example:

```
ls    # list files
```

or

```
who   # list users
```

Usually, however, it contains several commands and can include directions to NQS, file transfer requests, and even your programs.

For example, the following job script compiles and runs a C program:

```
date                        # Print date/time
cc loop.c    -o loop.out    # Compile loop.c
```

```
date                        # Print date/time
loop.out                    # Execute loop.out
date                        # Print date/time
echo job complete
```

You can submit a request either to the NQE database or to NQS (for information about submitting a request to run under the Distributed Computing Environment (DCE), see the *NQE User's Guide*, publication SG–2148).

> **Note:** If your request will run on a UNICOS system that has the multilevel security (MLS) feature enabled or a UNICOS/mk system that has security enhancements enabled, your NQE client may be required to be in a workstation access list (WAL). If you cannot get authenticated on the node, see your UNICOS or UNICOS/mk system administrator.

You can submit a request by using either the NQE GUI or the command line interface.

2.5.1 Submitting a Request Using the NQE GUI

To submit a request to NQE, access the NQE GUI by entering the `nqe` command. The initial NQE GUI button bar window will appear (as shown in Figure 2, page 11). Using the left mouse button, click once on the `Submit` button.

The `Submit` window shown in Figure 3 will appear.

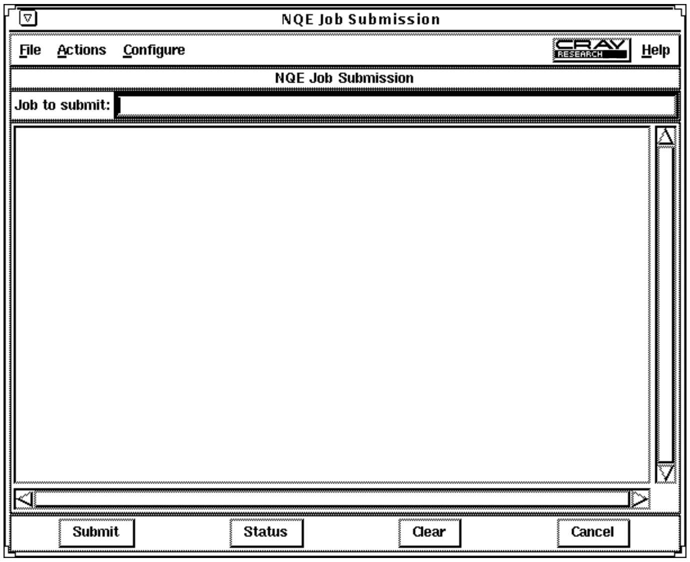


Figure 3. NQE GUI `Submit` Window

The following sections describe the components of the NQE GUI `Submit` window, and Section 2.5.1.2, page 18, describes the general steps to follow to submit an existing request.

2.5.1.1 `Submit` Window Description

The `Submit` window is composed of the menu bar, the `Job to submit` line (the job script file path name), the job edit area, and the actions button bar; each of these four segments is described as follows.

> **Note:** For detailed information about the `Submit` window options or for information about how to configure mouse button settings differently, see the *NQE User's Guide*, publication SG–2148.

- *Menu bar*. Menu buttons are located within the menu bar, which is located at the top of the window. When you select a menu, it opens a submenu that contains lists of options. To open a menu window, place the pointer on the menu name and click once on the left mouse button.

- `File` *menu*. The `File` menu lets you open, edit, launch, save, and submit a job script (or request); view, segment, delete, or reset the NQE GUI log; and exit the `Submit` window.
- `Actions` *menu*. The `Actions` menu lets you enter your password when your request runs on a host that uses password validation. This password is for the user name under which the request will execute.
- `Configure` *menu*. The `Configure` menu of the `Submit` window lets you specify options for your request and then save them to be reused as needed. You do not have to include directives in your batch request file. Options you specify on the `Configure` menu take precedence over any options in the script file.
- *Cray Research Logo*. This button opens a window that displays current NQE version and copyright information.
- `Help` *menu*. The `Help` menu lets you view the `nqe`(1) man page or lets you view information on how to access the *NQE User's Guide*, publication SG–2148.

- `Job to submit` *line*. The `Job to submit` line lets you specify the path name of a job script to be edited or submitted to the NQS server. When you enter a valid path name and press the `RETURN` key, the text from the file appears in the job edit area below the `Job to submit` line.
- *Job edit area*. You can make changes to the file displayed in the job edit area, and then save the changes by selecting `Save` or `Save as` from the `File` pull-down menu.
- *Actions bar*. The actions bar is located at the bottom of the `Submit` window. (To activate a button, place the pointer on the button and click once on the left mouse button.) When activated, these buttons perform the following actions:

Button	Description
`Submit`	Submits the specified job script; the job script will use the loaded configuration options and limits.
`Status`	Provides the same summary status display as provided by the `Status` button on the top-level NQE GUI button bar.
`Clear`	Clears the `Job to submit` entry and the job edit area of the display.

`Cancel` Cancels the `Submit` window.

2.5.1.2 General Steps to Follow to Submit an Existing Request

To submit an existing job script, do the following:

- Either enter the path name of the job script on the `Job to submit` line and press the `RETURN` key, or select `Open` from the `File` menu (the `Open` option uses a standard Motif file selection interface) and select the job script you want to submit. The job script text appears in the job edit area below the `Job to submit` line.

  You can modify the content of your job script in the job edit area of the `Submit` window. To save changes for future use, select `Save` or `Save as` on the `File` menu.

- You can submit a request directly to NQS or to the NQE database. To set the destination for your request, select either `Submit to NQE` or `Submit to NQS` on the `General Options` menu, and apply the change. If you do not select this option, the value of the `NQE_DEST_TYPE` environment variable, which you can set to be either `nqs` or `nqedb`, is used; otherwise, the value of `NQE_DEST_TYPE`, which is set in the `/etc/nqeinfo` file on your `NQS_SERVER`, is used.

- If your request will run on a host that uses password validation, use the `Actions` menu to enter your password.

- To submit the request file, click once on the `Submit` button that is located at the bottom of the `Submit` window.

  If your request is submitted successfully, you will receive a message similar to one of the following:

  – For requests submitted to NQS with the `cqsub` command, you will receive the following message:

  ```
  Request number.host submitted to queue:queue.
  ```

  – For requests submitted to NQS with the `qsub` command, you will receive the following message:

  ```
  nqs-181 qsub: INFO
    Request number.host: Submitted to queue queue by username(userid).
  ```

- For requests submitted to the NQE database, you will receive the following message:

  `Task id t`*number* `inserted into database nqedb.`

For additional information about messages received after submitting a request, see the *NQE User's Guide*, publication SG–2148.

To cancel the `Submit` window, click once on the `Cancel` button that is located at the bottom of the `Submit` window.

2.5.2 Submitting a Request Using the Command Line Interface

To use the command line interface to submit a batch request to NQE, use the `cqsub` command or the `qsub` command. For a complete list of the command options, see the `cqsub`(1) and `qsub`(1) man pages.

Simple forms of the `cqsub` and `qsub` command are as follows:

`cqsub [`*file*`]`

`qsub [`*file*`]`

The *file* option is the name of the job script file to be submitted to NQE for execution.

In the following example, you submit the `testjob` file:

```
% cqsub testjob
Request 367.coal submitted to queue: nqenlb.
```

The resulting message tells you that your NQS request ID (`367.coal`) and the name of the queue that accepted the request (`nqenlb`).

2.6 Monitoring a Request

You can monitor a request by using either the NQE GUI or the command line interface.

2.6.1 Monitoring a Request Using the NQE GUI

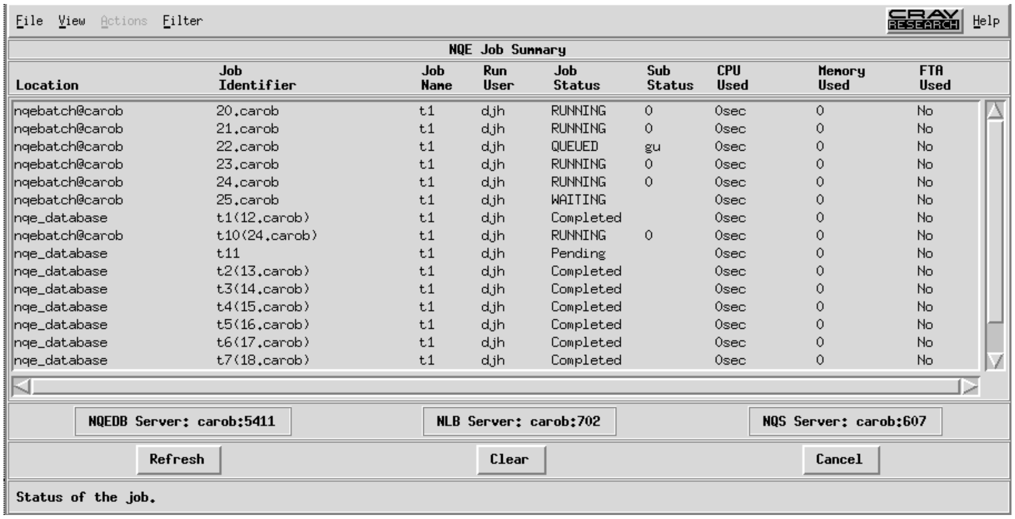
You can use the NQE GUI `Status` window to display information about your requests. The NQE GUI `Status` window provides a summary of request status that is refreshed periodically. By default, you see all requests in the group of execution nodes in the NQE cluster. However, it could be that your NQE administrator has disabled this function and that you will see only the requests you submit.

Using the NQE GUI `Status` window lets you do the following:

- Monitor status of all your requests. You do not have to know the location of your request before you request status on it. Request status is updated (refreshed) at configurable intervals.
- Tailor the display. You can specify how you want your display to look and what information is displayed.

To open the NQE GUI `Status` window, access the NQE GUI by keying in the `nqe` command and, using the left mouse button, click once on the `Status` button of the initial NQE GUI button bar.

Figure 4 shows the `Status` window.



NQE Job Summary

Location	Job Identifier	Job Name	Run User	Job Status	Sub Status	CPU Used	Memory Used	FTA Used
nqebatch@carob	20.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	21.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	22.carob	t1	djh	QUEUED	gu	0sec	0	No
nqebatch@carob	23.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	24.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	25.carob	t1	djh	WAITING		0sec	0	No
nqe_database	t1(12.carob)	t1	djh	Completed		0sec	0	No
nqebatch@carob	t10(24.carob)	t1	djh	RUNNING	0	0sec	0	No
nqe_database	t11	t1	djh	Pending		0sec	0	No
nqe_database	t2(13.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t3(14.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t4(15.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t5(16.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t6(17.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t7(18.carob)	t1	djh	Completed		0sec	0	No

NQEDB Server: carob:5411

NLB Server: carob:702

NQS Server: carob:607

Status of the job.

a10375

Figure 4. `Status` Window

The `Status` window consists of the menu bar, the job summary area, the server identification area, the context-sensitive help area, and the actions bar. Each of these segments is described as follows.

Note: For detailed information about the `Status` window options, see the *NQE User's Guide*, publication SG-2148.

- *Menu bar*. Menu buttons are located within the menu bar, which is located at the top of the window. When you select a menu, it opens a submenu that contains lists of options. To open a menu, place the pointer on the menu name and click once on the left mouse button.
  - `File` *menu*. The `File` menu provides a selection for exiting the window.
  - `View` *menu*. The `View` menu lets you display a summary of jobs (`Job Summary`) or FTA transfers (`FTA Summary`). The default `Status` window is the `Job Summary` view.
  - `Actions` *menu*. The `Actions` menu lets you delete a job, send a signal to a job, and view a detailed status of a job or its job log. This menu also lets you monitor status on hosts that use password validation.
  - `Filter` *menu*. The `Filter` menu lets you reduce the number of requests shown in the window. You can select specific hosts, user names, the originating user, the originating host, request IDs or task IDs, or locations.
  - *Cray Research Logo*. This button opens a window that displays current NQE version and copyright information.
  - `Help` *menu*. The `Help` menu provides general and administrative information related to each task and configuration option available to you through this window.
  - `NQE Job Summary` *area*. The job summary area shows a summary of information for the view option you specified by using the `View` menu. The job summary view lists jobs according to filters that you have set by using the `Filter` menu selections. The `FTA Summary` view lists FTA transfers that are in progress.

The default view (of the `Status` window) is `Job Summary`. The following data about requests is displayed by default:

Column name	Description
`Location`	The request's location, which can be either a queue or the NQE database.
`Job Identifier`	The job identifier; possible identifiers are as follows:

- NQS request ID (for example, `5703.fog`), as displayed when you submitted the request to NQS.
- NQE database ID, known as the task ID (for example, `t1`), as displayed when you submitted the request to the NQE database.
- NQE database ID with the NQS request ID. If you submitted a request to the NQE database, a copy of the request is executed while the original request remains in the NQE database. The request ID is the request identifier of the copy of the request executing in NQS and is displayed in parentheses after the `t`*id* (for example, `t4(61178.rain)`).

`Job Name` Name of the request.

`Run User` User name with which the request was submitted.

`Job Status` Status of the request.

`SubStatus` Substatus of the request.

`CPU Used` CPU usage (in seconds) for the request. On some platforms, a display of the amount of CPU that the request consumes is not available, and a 0 appears in this column.

`Memory Used` Memory usage (in words) for the request. On some platforms, a display of the amount of memory that the request consumes is not available, and a 0 appears in this column.

`FTA Used` FTA usage for the request; usage setting can be `Yes` or `No`.

To select a request for use with the `Actions` menu selections, first position the pointer on the desired job line in the job summary area and select it by clicking the left mouse button once; then select the `Actions` menu selection.

To display a detailed status of a request, first position the pointer on the desired job line in the job summary area and then double-click the left mouse button on the desired summary line.

- The server identification area displays the host names and port numbers of the NQE database server, the NLB server, and the NQS server that are providing the information displayed by the `Job Summary` or `FTA Summary` selections of the `Status` window.

- The context-sensitive `Help` area is located at the bottom of the `Status` window. This area shows one-line informational messages about the area on the `Status` window that is directly under your mouse pointer.
- *Actions bar*. The actions bar is located at the bottom of the `Status` window. To activate a button, place the pointer on the button and click once on the left mouse button. When activated, these buttons perform the following actions:

Button	Description
`Refresh`	Refreshes the status window with new status information
`Clear`	Clears the highlighted request line(s)
`Cancel`	Cancels the `Status` window

2.6.2 Monitoring a Request Using the `qstat` Command

The `qstat` command displays all of your own requests that were sent to a local host. You can use the `qstat -h` *targethost* command to display your requests that were sent to a specific NQE server. For more information about using the `qstat` command, see the *NQE User's Guide*, publication SG–2148, or the `qstat`(1) man page.

2.6.3 Monitoring a Request Using the `cqstatl` Command

The `cqstatl` command displays all of your requests that are running on your NQS server.

For a summary of `cqstatl` options, see the `cqstatl`(1) man page.

This section covers the following topics:

- Displaying summaries
- Displaying details

For information about displaying requests on other servers or specifying another user name, see the *NQE User's Guide*, publication SG–2148.

2.6.3.1 Displaying Summaries

You can display a summary of requests that are in batch queues, pipe queues, and the NQE database (requests in pipe queues are not applicable for requests sent to the NQE database).

To display summary information for specific requests sent to the NQE database, use the following command (if you have the `NQE_DEST_TYPE` environment variable set to `nqedb`, omit the `-d nqedb` option):

```
cqstatl -d nqedb tids
```

The `t`*ids* argument is the task identifier displayed when you submitted the request to the NQE database. You can specify more than one task identifier. Separate request identifiers with a space. (The `t`*id* is also displayed on the NQE GUI `Status` window.)

> **Note:** By default, if you use the `cqstatl` command without options or arguments, the output is a summary of each NQS queue on the NQS server. However, if you have the `NQE_DEST_TYPE` environment variable set to be `nqedb`, and you use the `cqstatl` command without options or arguments, the output is a summary of all your requests in the NQE database minus all terminated requests. (For more information about monitoring queues, see the *NQE User's Guide*, publication SG–2148, or the `cqstatl`(1) man page.)

To display summary information for specific NQS requests, use the following command:

```
cqstatl -d nqs requestids
```

The *requestids* argument is one of the following:

- If you submitted a request to NQS, *requestid* is the request identifier displayed when you submitted the request to NQS.
- If you submitted a request to the NQE database, *requestid* is the request identifier of the copy of the request executing in NQS. The *requestid* is displayed on the NQE GUI `Status` window in parentheses after the `t`*id* (for example, `t4(61178.rain)`).

You can specify more than one request. Separate request identifiers with a space.

To display summary details of all your requests in the NQE database, use the following command:

```
cqstatl -d nqedb -a
```

To display summary details of all your requests on your NQS server (as defined by `NQS_SERVER`), use the following command:

```
cqstatl -a -d nqs
```

For more details about the `cqstatl` displays, see the *NQE User's Guide*, publication SG–2148.

You cannot use `cqstatl` to display details about the requests and NQS activity of other users unless you are an NQE administrator or unless you are authorized to execute NQS requests under another user name. For more information, see the *NQE User's Guide*, publication SG–2148.

2.6.3.2 Displaying Details

To display the full details of all your requests, use one of the following commands.

- If you submitted a request to NQS, use the following command:

  ```
  cqstatl -d nqs -f requestids
  ```

  The *requestids* argument is the request identifier displayed when you submitted the request to NQS. You can specify more than one request. Separate request identifiers with a space.

- If you submitted a request to the NQE database, use the following command:

  ```
  cqstatl -d nqedb -f tids
  ```

  > **Note:** If you have the `NQE_DEST_TYPE` environment variable set to be `nqedb`, omit the `-d nqedb` option.

  For requests sent to the NQE database, the `t`*ids* argument is the task identifier of the request in the NQE database. You can specify more than one task. Separate task identifiers with a space. When the request is in NQS, it receives a *requestid*, which is displayed on the NQE GUI `Status` window in parentheses after the `t`*id* (for example, `t4(61178.rain)`).

Note: If a request that was sent to the NQE database is executing, `cqstatl` obtains status from NQS. If the request that was sent to the NQE database is not executing, status information is obtained from the NQE database. The detailed display of an NQE database request also includes the request's NQE task identifier (`t`*id*).

For more information about the `cqstatl` display, see the *NQE User's Guide*, publication SG–2148, or the `cqstatl`(1) man page.

2.7 Deleting a Request

You can delete a request that you have submitted to NQE. The request to be deleted can be executing or waiting to execute on an NQS server. When you delete a request, the original file is not deleted; only the request to execute the file is deleted.

> **Note:** If your site uses password validation, you must set the `NQS_PASSWORD_NEEDED` environment variable, include the `-P` option with the `cqdel` command, or, in the NQE GUI `Submit` window, select `Set Password` on the `Actions` menu to ensure that a password is sent to the server with your request; otherwise, your request will not execute. The password you supply is for the user name under which the request will execute.

After you submitted the job request to be executed, you received a response similar to one of the following; the response includes the unique ID that is assigned to the request:

- If you submitted your request to NQS with the `cqsub` command, you received a response similar to the following:

  ```
  Request 46.latte submitted to queue: nqenlb
  ```

- If you submitted your request to NQS with the `qsub` command, you will receive the following message:

  ```
  nqs-181 qsub: INFO
    Request number.host: Submitted to queue queue by username(userid).
  ```

- If you submitted your request to the NQE database, you received a response similar to the following:

  ```
  Task id t4 inserted into database nqedb
  ```

To delete a request that you have submitted but no longer want to execute, use the `Status` window of the NQE GUI, or the `cqdel` or `qdel` command.

> **Note:** You can use the NQE GUI to delete a request whether or not the request is executing.

Note: A request in the NQE database is known as a task and is assigned a task ID (for example, t4). When a copy of the request is executing under NQS, it is also assigned a request ID (for example, 288.latte), which is displayed in parentheses after the task ID in the NQE GUI Status window Job Identifier field (for example, t4(288.latte)).

Note: If the UNICOS multilevel security (MLS) feature or UNICOS/mk security enhancements are enabled on your system, for information about deleting a request from an NQS queue, see the *NQE User's Guide*, publication SG–2148.

2.7.1 Deleting a Request by Using the NQE GUI

To use the NQE GUI to delete a request, select Delete Job on the Actions menu of the NQE GUI Status window. This section describes how to delete a request that was sent to NQS or a request that was sent to the NQE database.

Note: You can use the NQE GUI to delete a request whether or not the request is executing.

Note: If your site uses password validation, you must either set the NQS_PASSWORD_NEEDED environment variable or, in the NQE GUI Submit window, select Set Password on the Actions menu to ensure that you are prompted for a password; otherwise, your request will not execute. The password you supply is for the user name under which the request will execute.

After you submitted the request to be executed, you received a response similar to one of the following:

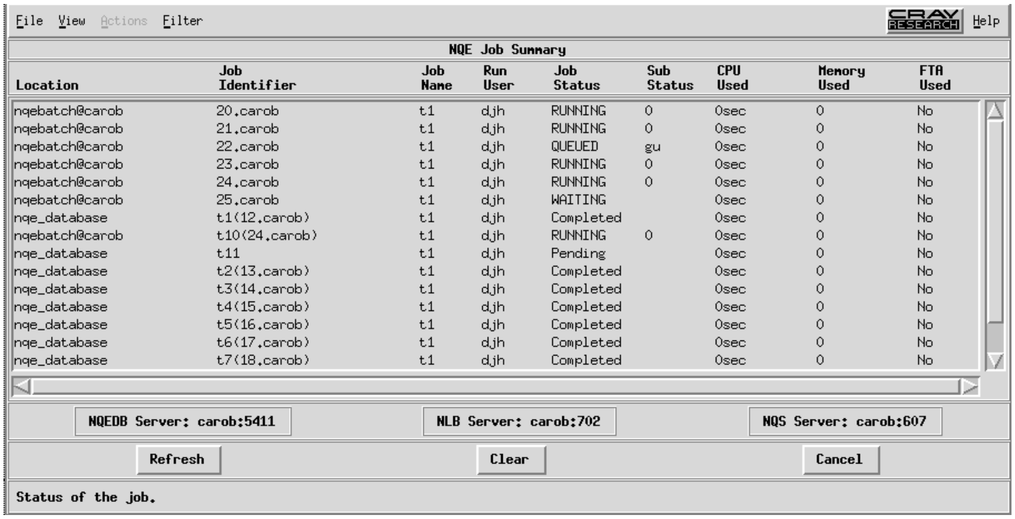
- If you submitted your request to NQS, you received a response similar to the following:

  ```
  Request 46.latte submitted to queue: nqenlb
  ```

- If you submitted your request to the NQE database, you received a response similar to the following:

  ```
  Task id t4 inserted into database nqedb
  ```

To display the status of your request, use the NQE GUI Status window. Figure 5 shows a sample NQE GUI Status window. The Location column of the display shows requests submitted to NQS (in the format of *queue@host*) and requests submitted to the NQE database (in the format of nqe_database).



Location	Job Identifier	Job Name	Run User	Job Status	Sub Status	CPU Used	Memory Used	FTA Used
nqebatch@carob	20.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	21.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	22.carob	t1	djh	QUEUED	gu	0sec	0	No
nqebatch@carob	23.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	24.carob	t1	djh	RUNNING	0	0sec	0	No
nqebatch@carob	25.carob	t1	djh	WAITING		0sec	0	No
nqe_database	t1(12.carob)	t1	djh	Completed		0sec	0	No
nqebatch@carob	t10(24.carob)	t1	djh	RUNNING	0	0sec	0	No
nqe_database	t11	t1	djh	Pending		0sec	0	No
nqe_database	t2(13.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t3(14.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t4(15.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t5(16.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t6(17.carob)	t1	djh	Completed		0sec	0	No
nqe_database	t7(18.carob)	t1	djh	Completed		0sec	0	No

NQEDB Server: carob:5411

NLB Server: carob:702

NQS Server: carob:607

Status of the job.

a10375

Figure 5. NQE GUI `Status` Window Example

Highlight the request in the job summary area, select the `Actions` menu, and then select `Delete Job`, which deletes the request currently selected from the job summary area. You will receive a response stating that your request was deleted.

For a description of each NQE GUI `Status` option, see the *NQE User's Guide*, publication SG–2148; for a summary of the NQE GUI `Status` options, see the `nqe`(1) man page.

2.7.2 Deleting a Request by Using the Command Line Interface

To use the `cqdel` or `qdel` command to delete a request that has not started executing on NQS, provide the NQE database task ID or the NQS request ID on the command line. For example, to delete request `46.latte`, which was sent to NQS, you would enter the following command:

```
% cqdel -d nqs 46.latte
Request 46.latte has been deleted.
```

If a request has already begun execution, you see the following message when you issue the `cqdel` or `qdel` command:

```
% cqdel -d nqs 5167.sequoia
QUESR: ERROR: Failed to delete request "5167.sequoia"
QUESR: ERROR: Request is running at transaction peer
```

To use the `cqdel` or `qdel` command to delete a request that has started executing, you must send the request a signal by using the `cqdel -k` or `qdel -k` command, as described on the `cqdel`(1) or `qdel`(1) man page.

The `cqdel` or `qdel` command used without options does not affect an executing request.

> **Note:** If your site uses password validation, you must include the `-P` option on the `cqdel` command line or set the `NQS_PASSWORD_NEEDED` environment variable to ensure that you are prompted for a password. The password you supply is for the user name under which the request will execute.

For more information about deleting a request, see the *NQE User's Guide*, publication SG–2148, or the `cqdel`(1) or `qdel`(1) man page.

2.8 Monitoring Machine Load

The Network Load Balancer (NLB) displays information about the hosts in the NQE cluster. This information is used for load balancing, display of machine load data, and display of request status in the network.

To access machine-load information, click once on the `Load` button of the NQE GUI interface. The window provides the following:

- Continually updated status that allows easy comparison of the workload of servers
- Visual indication that a host is not providing new data
- Pop-up windows that provide information on a server

You can view data from the main window, by individual host, and through a miniature summary display. Figure 6 shows the main `Load` window.

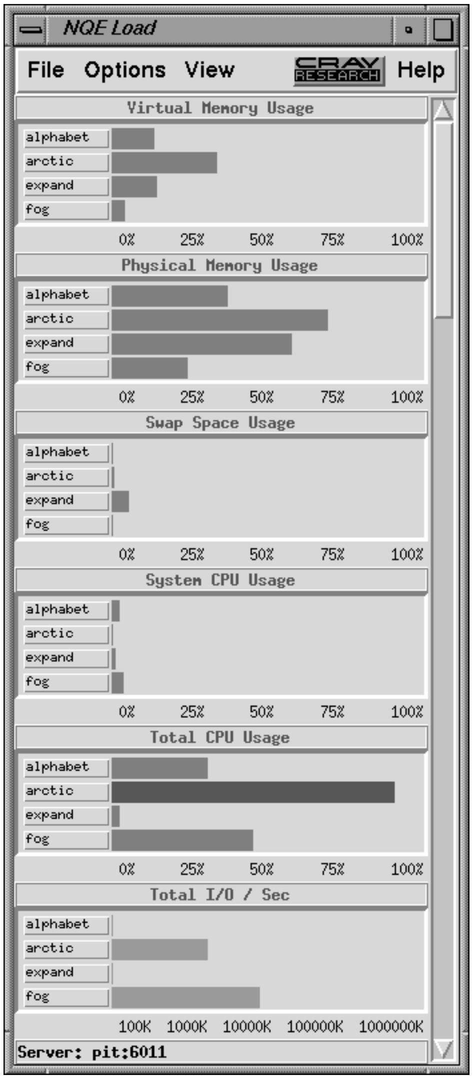


Figure 6. `Load` Window

The `Load` window is composed of the menu bar, the NQE load display, and the server name display area. Each of these segments is described as follows:

- *Menu bar*. The menu bar is located at the top of the `Load` window and displays buttons that open `Load` menus. To open a menu window, place the pointer on the menu name and click once on the left mouse button.
  - `File` *menu*. The `File` menu contains the `Exit` option that closes all of the windows associated with the NQE load display and terminates the program.
  - `Options` *menu*. The `Options` menu contains the `Host Selection` option, which provides selections for hosts that you want displayed, and the `Chart Editor` option, which creates new charts, changes the configuration of an existing chart, and adds or removes a chart from the `Load` window.
  - `View` *menu*. The `View` menu contains the `Chart Formulae` option, which displays the formulas used to calculate each of the charts.
  - `Help` *menu*. The `Help` menu lets you view the `nqe`(1) man page or view information on how to access the *NQE User's Guide*, publication SG–2148.
- *NQE load display*. The NQE load display provides continually updated machine-load data for participating machines in the cluster. Each of the charts in the display has a title, a scale, and one button per host.
- *Server name display area*. The server name display area displays the name of the server.

You can view data about a specific host or you can view the same data that is provided on the main `Load` window (memory demand, percentage of system CPU in use, idle CPU, and total I/O per second) grouped by host rather than by type of data.

For more information, see the *NQE User's Guide*, publication SG–2148.

2.9 Recovery and Restart

Note: This functionality is not available on all systems running NQE.

If the operating system or NQS system is shut down or crashes before your request completes execution, you do not necessarily have to resubmit a batch request because NQS has job recovery capabilities.

When the operating system or NQS is shut down, checkpoint images of all executing requests can be written automatically to a restart file on disk. When the system becomes available, NQS uses the checkpoint image to try to restart each of the requests from the point they had reached in their execution.

When the operating system or NQS crashes, checkpoint images cannot be written for executing requests. However, you can include the `qchkpnt`(1) command within a request to cause NQS to write a checkpoint image of the request at particular points in its execution. When the system becomes available after a crash, NQS tries to restart the request from the latest checkpoint image.

If a request has not yet begun execution at the time of the shutdown or crash, or if no checkpoint image is available, the request remains in its NQS queue and is executed from the start after the system becomes available.

For more information about recovery and restart, and to determine which systems running NQE support this capability, see the *NQE User's Guide*, publication SG–2148.

Getting Help [3]

This chapter describes the online documentation available for NQE and describes where to find troubleshooting information.

See the *NQE Release Overview*, publication RO–5237, for release-specific information.

3.1 Online Documentation

Online documentation consists of the following:

- Online publications
- Online man pages
- NQE GUI help

3.1.1 Online Publications

NQE publications are as follows:

- *Introducing NQE*, publication IN–2153 (this publication)
- *NQE User's Guide*, publication SG–2148
- *NQE Administration*, publication SG–2150
- *NQE Installation*, publication SG–5236
- *NQE Release Overview*, publication RO–5237

All NQE publications are available online through the following sources:

- Cray Research Online Software Publications Library, which is an information delivery system that is delivered on the World Wide Web. The Cray Research Online Software Publications Library is available publicly at the following URL:

  `http://www.cray.com/swpubs/`

- Silicon Graphics Technical Publications Library, which is an information delivery system that is delivered on the World Wide Web. The Silicon

Graphics Technical Publications Library is available publicly at the following URL:

```
http://techpubs.sgi.com/library/
```

- Cray DynaWeb server application, which is included on a CD-ROM with the NQE release package. The Cray DynaWeb server application lets you view documents online by using a World Wide Web (WWW) browser such as Netscape or Mosaic. To access the Cray DynaWeb server, see your Cray Research system administrator for the local URL.
- PostScript files, which are provided on the Cray DynaWeb CD-ROM that is included with the NQE release package. PostScript files are also available through anonymous FTP at:

  `ftp://ftp.cray.com/pub/nqe/nqe`*xx*

3.1.2 Online Man Pages

The `man` command provides online help on all NQE commands. To view a man page online, type `man` *commandname*. You must ensure that your `MANPATH` is set properly, as described in Section 2.2, page 6. For a list of the NQE user man pages, see Section 2.4.2, page 13.

3.1.3 NQE GUI Help Function

The NQE GUI help function provides context sensitive help. It also lets you view the `nqe`(1) man page and lets you view information on how to access the *NQE User's Guide*, publication SG–2148, online.

3.2 NQE Troubleshooting

If you have problems as you start to use NQE, see the *NQE User's Guide*, publication SG–2148, for possible actions you can take.

Index