NEC Network Queuing System V (NQSV) User's Guide

[Reference]

Proprietary Notice

The information disclosed in this document is the property of NEC Corporation (NEC) and/or its licensors. NEC and/or its licensors, as appropriate, reserve all patent, copyright and other proprietary rights to this document, including all design, manufacturing, reproduction, use and sales rights thereto, except to the extent said rights are expressly granted to others.

The information in this document is subject to change at any time, without notice.

Preface

This guide is command reference of NEC Network Queuing System V (NQSV) job management system.

The manual of NEC Network Queuing System V (NQSV) is composed by following user's guides.

Name	Contents
NEC Network Queuing System V (NQSV) User's Guide [Introduction]	This guide explains the overview of NQSV and configuration of basic system.
NEC Network Queuing System V (NQSV) User's Guide [Management]	This guide explains the various management functions of the system.
NEC Network Queuing System V (NQSV) User's Guide [Operation]	This guide explains the various functions that used by general user.
NEC Network Queuing System V (NQSV) User's Guide [Reference]	The command reference guide.
NEC Network Queuing System V (NQSV) User's Guide [API]	This guide explains the C programming interface (API) to control NQSV.
NEC Network Queuing System V (NQSV) User's Guide [JobManipulator]	This guide explains about the scheduler component : JobManipulator.
NEC Network Queuing System V (NQSV) User's Guide [Accounting & Budget Control]	This guide explains the functions of accounting.

February 2018	1st edition
January 2023	18th edition
March 2023	19th edition
June 2023	20th edition
September 2023	21st edition
March 2024	22nd edition

Remarks

(1) This manual conforms to Release 1.00 and subsequent releases of the NQSV.

(2) All the functions described in this manual are program products. The typical functions of them conform to the following product names and product series numbers:

Product Name	product series numbers
NEC Network Queuing System V (NQSV) /ResourceManager	UWAF00 UWHAF00 (support pack)
NEC Network Queuing System V (NQSV) /JobServer	UWAG00 UWHAG00 (support pack)
NEC Network Queuing System V (NQSV) /JobManipulator	UWAH00 UWHAH00 (support pack)

(3) UNIX is a registered trademark of The Open Group.

(4) Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

(5) OpenStack is a trademark of OpenStack Foundation in the U.S. and/or other countries.

(6) Red Hat OpenStack Platform is a trademark of Red Hat, Inc. in the U.S. and/or other countries.

(7) Linux is a trademark of Linus Torvalds in the U.S. and/or other countries.

(8) Docker is a trademark of Docker, Inc. in the U.S. and/or other countries.

(9) InfiniBand is a trademark or service mark of InfiniBand Trade Association.

(10) Zabbix is a trademark of Zabbix LLC that is based in Republic of Latvia.

(11) All other product, brand, or trade names used in this publication are the trademarks or registered trademarks of their respective trademark owners.

About This Manual

Notation Conventions

The following notation rules are used in this manual:

Omission Symbol	...	This symbol indicates that the item mentioned previously can be repeated. The user may input similar items in any desired number.
Vertical Bar	|	This symbol divides an option and mandatory selection item.
Brackets	{ }	A pair of brackets indicates a series of parameters or keywords from which one has to be selected.
Braces	[ ]	A pair of braces indicate a series of parameters or keywords that can be omitted.

Glossary

Term	Definition
Vector Engine (VE)	The NEC original PCIe card for vector processing based on SX architecture. It is connected to x86-64 machine. VE consists of more than one core and shared memory.
Vector Host (VH)	The x86-64 architecture machine that VE connected.
Vector Island (VI)	The general component unit of a single VH and one or more VEs connected to the VH.
Batch Server (BSV)	Resident system process running on a Batch server host to manage entire NQSV system.
Job Server (JSV)	Resident system process running on each execution host to manage the execution of jobs.
JobManipulator (JM)	JobManipulator is the scheduler function of NQSV. JM manages the computing resources and determines the execution time of jobs.
Accounting Server	Accounting server collects and manages account information and manages budgets.
Request	A unit of user jobs in the NQSV system. It consists of one or more jobs. Requests are managed by the Batch Server.

Job	A job is an execution unit of user job. It is managed by Job Server.
Logical Host	A logical host is a set of logical (virtually) divided resources of an execution host.
Queue	It is a mechanism that pools and manages requests submitted to BSV.
BMC	Board Management Controller for short. It performs server management based on the Intelligent Platform Management Interface (IPMI).
HCA	Host Channel Adapter for short. The PCIe card installed in VH to connect to the IB network.
IB	InfiniBand for short.
MPI	Abbreviation for Message Passing Interface. MPI is a standard for parallel computing between nodes.
NIC	Network Interface Card for short. The hardware to communicate with other node.

Contents

1. User's Commands

Following commands are installed under /opt/nec/nqsv/bin with NQSV-Client package.

Commands	Contents
budgetcheck(1)	Displays Budget Results
qalter(1)	Alter Attribute of Request
qattach(1)	Attach a Command to Running Request
qcat(1)	Display the error/input/output file of Jobs
qcmdconf(1)	Management of Remote execution type Interactive Function
qdel(1)	Delete Request
qdetach(1)	Delete Attached processes to Request
qhold(1)	Hold Request
qlogin(1)	Submit Interactive Request
qmove(1)	Move Request between Queues
qrerun(1)	Rerun Request
qrls(1)	Release Previously Held Request
qrsh(1)	Submit Remote Execution type Interactive Request
qsig(1)	Send Signal to Request
qstat(1)	Display the Status of NQSV
qsub(1)	Submit Batch Request
qwait(1)	Wait for the State Transition of Request Execution
qwait2(1)	Wait for the End of Multiple Requests
scacctjob(1)	Display the Job Accounting Data
scacctreq(1)	Display the Request Accounting Data
scacctrsv(1)	Display the Reservation Accounting Data
sstat(1)	Display JobManipulator Status
wdel(1)	Delete Workflow
wstart(1)	Execute Workflow
wstat(1)	Display Workflow

1.1. budgetcheck/ rbudgetcheck (1)

NAME

budgetcheck, rbudgetcheck -- displays budget results

SYNOPSIS

/opt/nec/nqsv/bin/budgetcheck [*options*]
/opt/nec/nqsv/bin/rbudgetcheck acct-server [*options*]

DESCRIPTION

budgetcheck obtains the users, groups, and account codes that have executed this command, and displays budget of the corresponding users, and, groups, and account codes. If no option is specified, this command displays only the user budgets.

rbudgetcheck accesses the accounting server and displays the budget usage information corresponding to the executing user. The 1st argument must be the host name of the accounting server and the budgetcheck's options can be specified after that.

The following options can be specified.

-N Shows the budgetcheck version information. This option takes priority even when other options are specified, and they are disregarded.

-P *priv* Specify access privilege for the command execution.
One of the following characters can be specified as *priv*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	General User privilege

When -P option is not specified, access privilege for this command execution is set by environment variable NQSV_PRIV and NQSII_PRIV. If also those environment variable are not set, the command runs with the user privilege.
Those environment variable can take the following string values.
Note that those environment variable cannot be used for rbudgetcheck.

PRIV_MGR	Manager privilege

PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	General User privilege

-u Displays user budgets. (Default)

-g Displays group budgets.

-a Displays account code budgets.

When the user's budget is exceeded, budgetcheck returns 1 as the exit status, and also returns 2 when the group's budget is exceeded, and returns 4 when the account code's budget is exceeded. A logical add is returned when two or more budget types have been exceeded.

The following is the example displaying all information.

```
# budgetcheck -uga

usr1      11223.41    120.00     12245.00
grp4          0.00      0.00  19874344.00
acct2         0.00      0.00  19874344.00
```

The order of display is user, group, account code.

The each line shows three numbers for the user, group or account code. The numbers are remains of the budget, estimate of billing and the initial budget, from the left.

SEE ALSO

sbuedit(1M-N), budgetedit(1M-N).

1.2. qalter(1)

NAME

qalter -- Alter Attribute of Request

SYNOPSIS

qalter [*options*] [*request_identifier*]

DESCRIPTION

Alter attributes of requests or interactive requests, such as resource limits and execution environments.

The following options can be specified as *options*.

-a *date_time*

Alter the request start time. The form for *date_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The two digits are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-A *account_string*

Alter the account code of a request to *account_string*. The maximum number of characters which can be specified for *account_string* is 15 bytes.

-d [*hostname:*]*path_name*

Alter the log output of a request to a file specified by *path_name*. Only the regular files can be targeted as *path_name*. A host can be specified by specifying *hostname:* as *path_name*. When the *hostname* is omitted, the name of the host which submits the request is used.

The following conversion-character can be specified as *path_name*.

%m	Converted to the Machine ID in Request ID.
%r	Converted to Request ID.
%s	Converted to the sequence number in Request ID.
%%	Converted to "%".

-D *log_level*

Alter the log level of a request. The following can be specified in *log_level*.

0	The request log is not output.
1	The logs about the request are output.
2	The logs about the request and the job are output.

-e [*hostname*:]*path_name*

Alter the standard error output of a request to a file or path specified by *path_name*. Only the regular files can be targeted as *path_name*. A host can be specified by specifying *hostname:* as path_name. When the *hostname* is omitted, the name of the host which submits the request is used. The following conversion-character can be specified as *path_name*.

%m	Converted to the Machine ID in Request ID.
%r	Converted to Request ID.
%s	Converted to the sequence number in Request ID.
%[0*n*]j	Converted to Job number. *n* is a number of digits. [Example] %j : 0,1,2... %03j : 000,001,002...
%%	Converted to "%".

-H y | n

Alter hold enable/disable. "y" will switch to a hold-able request. And "n" will switch to an unholdable request.

-j *join_list*

Merges standard output and standard error output of requests. Merged results will be output to a path specified by join_list. The following can be specified for *join_list*.

e Outputs results of merging standard output and standard error output of requests to standard errors. (JOIN_STD_ERROR).

o Outputs results of merging standard output and standard error output of requests to standard output (JOIN_STD_OUTPUT).

-J y | n

Alter migration enable/disable. "y" will switch to a migratable request. And "n" will switch to an unmigrat-able request.

-K *kernel_param_list*[,*kernel_param_list*,...]

Alter kernel parameter value.
Operator privilege or higher is needed to change kernel parameter values.
The *kernel_param_list* is as follows:

nice=*value*	nice value

-l *resource_list*[,*resource_list*,...]

Alter maximum and warning values for resource limits with requests, jobs and processes. More than one resource_list can be specified by listing delimited with conma (without putting space).
The *resource_list* are as follows. Enclose with double quotas("") when both maximum and warning values of a limit are specified. Warning value of the limit is kept current value if warning value is omitted. However, as a result of changing the maximum value, if the maximum value becomes smaller than the warning value, change the warning value to the same value as the maximum value.

Options for resource limits per request

elapstim_req=["]*max_limit*[,*warn_limit*"]

Limit by elapsed time after start of execution

Options for resource limits per job/logical host

cpunum_job=*max_limit*

Limit on the number of CPUs that can be used
In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

cputim_job=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

gpunum_job=*max_limit*

Limit on the number of GPUs that can be used

memsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VH.

vmemsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Options for resource limits per process

coresz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum core file size that can be created

cputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

datasz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum data segment size that can be used

In case of SX-Aurora TSUBASA, it is the data size used on VH.

filenum_prc=*max_limit*

Limit on the number of files that can be opened simultaneously

filesz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum file size that can be created

stacksz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum stack size that can be used

In case of SX-Aurora TSUBASA, it is the stack size used on VH.

vmemsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Limit values are sometimes converted into values that are convenient to the machine.

-m *mail_options*

Alter timing to send e-mail on changes of request status. The following can be specified in *mail_options*:

a	E-mail is sent when a request is abnormally terminated.
b	E-mail is sent when a request is started.
e	E-mail is sent when a request is terminated.
n	E-mail is not sent. (An error will result if specified with other mail_options.)

-M *mail_list*

Alter the user list for sending e- mails. The form of the *mail_list* is as follows:

mail_address[,*mail_address*, ...]

The maximum length of the *mail_list* that can be specified is 1023 bytes.

-N *name*

Alter the request name. A maximum of 63 bytes can be specified in a *name*.

-o [*hostname*:]*path_name*

Alter the standard output of a request to a file or a path specified in *path_name*. Only the regular files can be targeted as path_name. A host can be specified by specifying *hostname*: in *path_name*. When the *hostname* is omitted, it will be the name of the host which submits the request. The following conversion-character can be specified as *path_name*.

%m	Converted to the Machine ID in Request ID.		
%r	Converted to Request ID.		
%s	Converted to the sequence number in Request ID.		
%[0*n*]j	Converted to Job number. *n* is a number of digits.		
	[Example]	%j	: 0,1,2...
		%03j	: 000,001,002...
%%	Converted to "%".		

-p *priority*

Alter the request priority.

Priority must be an integer of [- 1024...1023].

-P *privilege*

Specify access privilege for command execution.

The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.

If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-r y | n

Alter rerunning enable/disable. "y" will switch to a rerunable request and "n", a request that cannot be rerun.

-S *path_name_list*

Alter the shell to execute the shell script of requests.

The form for *path_name_list* is as follows:

-T *job_topology*

Alter the mode of jobs to be executed. The following can be specified in *job_topology*.

distrib

Jobs to be executed will be distributed jobs.

If there are more than two jobs that are specified by option -b of qsub(1) command, specified scripts will be executed on each job servers (in the number of jobs).

intmpi

Jobs to be executed will be intmpi jobs.

mvapich

Jobs to be executed will be mvapich jobs.

necmpi

Jobs to be executed will be necmpi jobs.

openmpi

Jobs to be executed will be openmpi jobs.

pltmpi

Jobs to be executed will be pltmpi jobs.

-V

Display only the version of the command and NQSV API.

--accept-sigterm{=[*yes* | *no*]}

Specifies whether to enable or disable SIGTERM acceptance in scripts executed in the request. If this option is set to yes, SIGTERM is enabled for the script.

If the optional argument is omitted, it is assumed to be yes.

If this option is not specified, the default behavior is no (disabled).

--enable-cloud-bursting{=[*yes* | *no*]}

Specifies whether to burst and execute request to cloud computing resources. If this option is set to *yes*, bursting will be performed. If this option is set to *no*, bursting will not be performed. If this option is not specified, the default behavior is *no*.

--userpp-script=*location*:*path*[,*location*:*path*]

Alter the UserPP of the location specified by *location* to script path specified by *path.* If *path* not specified, delete the UserPP script specification. The following can be specified in *location.*

por	Execute location is POST-RUNNING.
prr	Execute location is PRE-RUNNING.

--warning-signal=elapstim:*signal*

Specifies a sending signal at exceeding the warning value of the limit by elapsed time after start of execution. *signal* can be specified in the following form:

Signal name: (Example: SIGHUP)
Signal name with "SIG" removed: (Example: HUP)
Signal number: (Example: 1) For detailed information. refer to signal(7).

Options for resource limits per logical host

Maximum or warning values for resource limits per logical host can be altered by the following options. Warning value of the limit is kept current value if warning value is omitted. However, as a result of changing the maximum value, if the maximum value becomes smaller than the warning value, change the warning value to the same value as the maximum value. Please refer for detail to **FORM OF RESOURCE LIMIT VALUE**.

--cpunum-lhost=*max_limit*

Alter the limit on the number of CPUs per logical host. This option corresponds to -l cpunum_job=max_limit (Limit on the number of CPUs that can be used(Job)). In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

--cputim-lhost=*max_limit*[,*warn_limit*]

Alter the limit of CPU time per logical host. This option corresponds to -l cputim_job=max_limit (Limit of CPU time (Job)). In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

--gpunum-lhost=*max_limit*

Alter the limit on the number of GPUs that can be used per logical host. This option corresponds to -l gpunum_job=max_limit (Limit on the number of GPUs that can be used (Job)).

--memsz-lhost=*max_limit*[,*warn_limit*]

Alter the limit on maximum memory size that can be used per logical host. This option corresponds to -l memsz_job=max_limit (Limit on maximum memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the memory size used on VH.

--stderrsz-lhost=*max_limit*[,*warn_limit*]

Alter the limit of stderr file size per logical host.

--stdoutsz-lhost=*max_limit*[,*warn_limit*]

Alter the limit of stdout file size per logical host.

--vecputim-lhost=*max_limit*[,*warn_limit*]

Alter the limit of VE CPU time per logical host. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-lhost=*max_limit*[,*warn_limit*]

Alter the limit on maximum VE memory size per logical host. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

--venum-lhost =*max_limit*

Alter the limit on number of VE node per job.

--vmemsz-lhost=*max_limit*[,*warn_limit*]

Alter the limit on maximum virtual memory size per logical host. This option corresponds to -l vmemsz_job=max_limit (Limit on maximum virtual memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Options for resource limits per venode

Maximum or warning values for resource limits per VE node can be altered by the following options. If the warning value is omitted, the current value is retained. However, the warning value to the same value as the maximum value if the maximum value is smaller than the current warning value as a result of changing the maximum value.

Please refer for detail to **FORM OF RESOURCE LIMIT VALUE**.

--vecputim-venode=*max_limit*[,*warn_limit*]

Alter the limit of VE CPU time per VE node. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-venode=*max_limit*[,*warn_limit*]

Alter the limit on maximum VE memory size per VE node. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

FORM OF RESOURCE LIMIT VALUE

Time Limit

Limits on time are specified in the following form:

[[*hours*:]*minutes*:]*seconds*[.*fraction*]

Specify the character string "unlimited" for unlimited.
Maximum limit of default is 2^31-1 seconds.

[Specification Example]

1234:58:21.29	1234 hours 58 minutes and 21.29 seconds
59:01	59 minutes and 1 second
12345	12345 seconds
121.1	121.1 seconds

Size Limit

Specify size limits in the following form:

integer[.*fraction*][*units*]

The following units can be specified in "units".

b	Byte
kb	Kilobyte (1024 b)
mb	Megabyte (1024 kb)
gb	Gigabyte (1024 mb)
tb	Terabyte (1024 gb)
pb	Petabyte (1024 tb)
eb	Exabyte (1024 pb)

Failing to specify in units will be construed as having specified byte. Specify the character string "unlimited" for unlimited. Maximum limit of default is 2^31-1 x bytes.

[Specification Example]

1234	1234Bytes
1234kb	1234Kilobytes
1234.5gb	1234.5Gigabytes

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege.

SEE ALSO

qstat(1), qsub(1)

NOTES

Note the following points concerning attribute alteration.

- Some of attributes cannot be changed when a request is executed in an execution queue.
- Some of attributes cannot be changed when a request is routed in a routing queue.
- It is not possible to change to values that exceed the resource limit value set on an execution queue.
- To specify a character used as a delimiter of options (such as " in -I option) as a normal character, escape.

1.3. qattach(1)

NAME

qattach -- Attach a command to running request

SYNOPSIS

qattach [-V] -c "command" [*jobnums*:]*request_identifier*

DESCRIPTION

qattach executes a command specified by command in running jobs.
Request_identifier is the request-ID of target request to attach the command.
And *jobnums* is the target job number in the request.
When *jobnums* is omitted, the specified command is attached to all jobs in the request.

jobnums has the following format:

job_number[,*job_number*]

Job_number is a job number or a range of job numbers (Ex.1-5).

Only the request owner can attach the request.
And the request must be submitted to the queue enabled attach function.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_INTERACTIVE_IP or NQSII_INTERACTIVE_IP

Client host's IP address (IPv4 numbers-and-dots notation).

SEE ALSO

qsub(1), qmgr(1M)

NOTES

- While a request is attached by qattach command, it is not available to suspend or hold the request.
- For the program executed by attaching to an interactive request, X11 forwarding cannot be

performed even if the interactive request is submitted with X11 forwarding option.

1.4. qcat(1)

NAME

qcat -- Display the error/input/output file of Jobs

SYNOPSIS

qcat [-P *privilege*] [-V] [-e | -i | -o | -p | -d | --userpp-script=<*args*>] [-b | -f] [-n *number*] [-h1 | 2] *job_identifier* ... <<by Job>>

qcat [-P *privilege*] [-V] [-e | -o] [-b | -f] [-n *number*] [-h 1 | 2] [*universe*.]*rank*@*request_identifier* ... <<by Rank>>

DESCRIPTION

When job_identifier is specified, qcat displays one of the following of the job specified by *job_identifier*.

- the input file
- the standard output file
- the error output file
- the list of running processes in the job
- the request log of a parent request
- the standard output or error output of UserPP script

job_identifier is specified in [*job_number*:]*request_identifier*. If job_number is omitted, the job with the job number 0 of *request_identifier* is assumed as a target job.

And the qcat command format by rank can be used for the request in which NEC MPI program is executed. By specifying a request ID of necmpi type request, MPI rank number and MPI universe number to *request_identifier*, rank and universe, specified MPI job's standard output/error can be displayed.
If universe is omitted, the output from universe number 0 is displayed. The qcat command format by rank can be used for necmpi type request. And at the submission, it is necessary to specify -f option to qsub command. For the request that submitted without -f option, only job display format can be used.

The following options can be specified:

-b

Display from the top of the file.

-d

Display the request log of a parent request.

-e

Display the error output file, if exists.

-f

With -o or –e option, display the appended data when the file grows.

-i

Display the input file(script file).

-n

Display number lines of the file.

-o

Display the standard output file, if exists.

-p

Display running process list of the job(ps output image).

-h *type*

Display the header.
This option is ignored if specified with the -d option.
The following can be specified for *type*.

<<Display by Job>>

1. "[*job_identifier*]" is displayed at the head of each line.
2. Display "==> job_identifier <==", and the contents of the file are displayed in the next lines.

<<Display by Rank>>

1. "[*applnum.universe.rank@request_identifier*]" is displayed at the head of each line.
2. Display"==>applnum.universe.rank@request_identifier<==", and the

contents of the file are displayed in the next lines.

* applnum is MPI program number (each mpirun spawned in a job has a different number).

-n *number*

Display number lines of the file.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privilege, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V

Display only the version of the command and NQSV API.
If neither of option(-e, -i, -o, -p, -d) is specified, the input file is displayed. If -b option is not specified, the file is displayed from the last line. If -n option is omitted, 10 lines of the file are displayed.

--userpp-script=<*args*>

Display the standard output or error output of UserPP script.
Following argument can be specified for <args>.

argument	display value
prr:o	Standard output of UserPP script on PRE-RUNNING
prr:e	Standard error of UserPP script on PRE-RUNNING
por:o	Standard output of UserPP script on POST-RUNNING
por:e	Standard error of UserPP script on POST-RUNNING

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qstat(1), qsub(1)

1.5. qcmdconf(1)

NAME

qcmdconf -- Management of Remote execution type Interactive Function

SYNOPSIS

qcmdconf -a --name=*entry_name* --cmd=*remote_cmd_line* --queue=*queue_name*
[--opt=*qrsh_options*] [-f *file_name*] [-S]
qcmdconf -o *entry_name* [-f *file_name*] [-S]
qcmdconf -l [-S]
qcmdconf -d *entry_name* [-S]

DESCRIPTION

1. -a option

qcmdconf -a registers a command line to execute as an interactive request and the option of qrsh to submit the request. Also qcmdconf -a makes a 'Remote execution type' program file that execute submission of interactive request as registered.

--cmd=*remote_cmd_line*

remote_cmd_line is the command line to execute as an interactive request. The maximum length of *remote_cmd_line* is up to 1023. If the command line includes a space character, *remote_cmd_line* must be enclosed by single or double quotation marks. And to use a control code character, escape by back-slash.

-f=*file_name*

The 'Remote execution type' program is created as *file_name*.

--name=*entry_name*

entry_name is a registration name of 'Remote execution type' program. The *entry_name* is used to identify 'Remote execution type' program. If -f option (described below) is omitted, *entry_name* is used as the file name of the 'Remote execution type' program.

The name already registered cannot be specified as *entry_name*.

--opt=*qrsh_options*

At the interactive request is submitted, *qrsh_options* is set as qrsh's option. If *qrsh_options* includes a space character, enclosed it by single or double quotation marks.

--queue=*queue_name*

queue_name to execute 'Remote execution type' program, interactive request is submitted to the queue specified by *queue_name* using qrsh command.

-S

qcmdconf is executed with manager privilege and registers/creates a 'Remote execution type' program for all users. The data is registered to the common database. Only super-user can use this option.

Without -S option, qcmdconf creates a 'Remote execution type' program for execution user's private purpose (makes a private database under execution user's home directory).

2. **-d** option

Remove the 'Remote execution type' program registration.

After removing registration, remove the 'Remote execution type' program file manually.

entry_name

The registration entry name to remove.

-S

Remove the 'Remote execution type' program registration from the common database. If -S option is omitted, the registration is removed from the execution user's private database.

3. **-l** option

List information of 'Remote execution type' program registration. The registration information is output from the common database and the execution user's private database.

-S

List information of 'Remote execution type' program registration from only the common database.

4. **-o** option

qcmdconf -o makes a 'Remote execution type' program file from a registered data. This option can restore the 'Remote execution type' program file from its registration when the command file is missing.

entry_name

qcmdconf makes the 'Remote execution type' program file from the registration data specified by *entry_name*. If -f option (described below) is omitted, *entry_name* is used as the file name of the 'Remote execution type' program.
If the registration name *entry_name* is not registered, qcmdconf -o fails.

-f *file_name*

The 'Remote execution type' program file is created as *file_name*.

-S

qcmdconf makes the 'Remote execution type' program file from the common database. Without -S option, qcmdconf creates 'Remote execution type' program file from execution user's private database.

FILES

$HOME/.nqsv/qcmd_list	The execution user's private database.
/etc/opt/nec/nqsv/qcmd_list	The common database.

SEE ALSO

qrsh(1)

1.6. qdel(1)

NAME

qdel -- Delete Request

SYNOPSIS

qdel [-P *privilege*] [-V] [-g *grace*] *request_identifier* ...

DESCRIPTION

Deletes requests specified by *request_identifier*.
The following options can be specified:

-g *grace*

First, SIGTERM is sent to jobs residing in execution hosts. After waiting for seconds specified by *grace*, SIGKILL is sent. In case *grace* is smaller than 0, only SIGKILL will be sent immediately. If this option is not specified, SIGKILL will be sent after waiting 5 seconds.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege.

SEE ALSO

qsig(1),qstat(1),qsub(1)

NOTES

If a request is specified during transferring result files, also the child network request of the request will be deleted.

1.7. qdetach(1)

NAME

qdetach -- Delete Attached processes to Request

SYNOPSIS

qdetach [-P *privilege*] [-V] [-g *grace*] *request_identifier* ...

DESCRIPTION

qdetach terminates the program attached and executing in NQSV job by qattach command. The target request is specified by *request_identifier*.
Only the owner of the request or NQSV operator can execute qdetach.

The following options are supported.

-g *grace*

Execute the detach after *grace* seconds. If 0 is specified as *grace*, the detach is executed immediately. When -g option is omitted, the default *grace* time is 5 seconds.

-P *privilege*

Specify NQSV access privilege for command execution. The following can be specified as *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, the default privilege is User privilege.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qattach(1)

1.8. qhold(1)

NAME

qhold -- Hold Request

SYNOPSIS

qhold [-P *privilege*] [-w] [-s] [-V] *request_identifier* ...

DESCRIPTION

Holds requests specified by *request_identifier*. Those requests that are in this state cannot be executed.
Requests that are put in a held state by this command can be released from a held state by Command qrls.

If the NQSV is shut down while a request is put on held, this condition will be continued in next startup.

Requests can be held in the following two states: QUEUED and WAITING.

The following options can be specified.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-s

If this option is specified with -w option, "." (period) will not be displayed during waiting completion of hold process.

-V

Display only the version of the command and NQSV API.

-w

Waits until hold process is completed without exiting the command. When hold process is successfully completed, Request <Request ID> holding complete.
Will be displayed and if failed, Request <Request ID> holding failed.
Will be displayed.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qrls(1),qstat(1),qsub(1)

NOTES

1. The running request on the execution host cannot be held.
2. If this command is executed with -w option, hold processing is not interrupted on the following situation.
   - Interrupt this command during waiting.
   - This command exited by the error during waiting

   To confirm the result of holding, use the qstat command.

1.9. qlogin(1)

NAME

qlogin -- Submit Interactive Request

SYNOPSIS

qlogin [*options*]

DESCRIPTION

qlogin submits interactive requests to NQSV. The interactive request provides job execution environment that can execute commands interactively at the execution host assigned by NQSV scheduler. By executing qlogin command, shell prompt of the execution host is displayed on the terminal qlogin invoked. At the shell prompt, any program on the execution host can be execute.

The following options are supported.

-a *date_time*

Set the interactive request scheduling start time. The form for *date_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

Each two digits of *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-A *account_string*

Set the account code of the submitted interactive request to *account_string*.

The maximum number of characters which can be specified for *account_string* is 15 bytes.
When the qlogin command is executed with no -A option, the account code name which is described in the $HOME/.acct is used as an account code. The format of $HOME/.acct is as following.

- The character string from the first character to the first space or tab or return is recognized as an account code name.

- The length of account code name specified in $HOME/.acct should not exceed 15 bytes.

When this option is not specified and there is no $HOME/.acct, account code will not to be specified.
When account code is specified in either the option or $HOME/.acct, qlogin does not check if it is an available account code.

-b *job_count*

Specify the number of jobs to be executed. When this option is not specified, the number of jobs to be executed will be 1.
If type of jobs is set to "distrib" (-T distrib), only 1 can be set to *job_count*.

-B ["]*job_condition*[","*job_condition*"...]

Specify the conditions to select jobservers when the scheduler allocates jobservers to interactive job (Jobcondition). Please refer to JobManipulator's manual for details of scheduling by this specification.

The form of *job_condition* is as follows.

[*job_number*:]*condition*

- *job_number* is the job number of the request (0 or more).

- When the *job_number* of *job_condition* is omitted or "ALL" is specified for *job_number*, the *job_condition* is effective to all jobs.

- The following can be specified for *job_number*.
  (1) Single number 0:

(2)	All jobs specified by "ALL"	ALL:
(3)	Two or more numbers using delimiters ","	0,2,5:
(4)	Consecutive numbers using "-"	0-4:
(5)	Combination of (3) and (4)	0,2,4-6:

- *job_number* is followed by scheduler's condition sentence (*condition*), separated by ":".

- The string which does not apply to the specified conditions of *job_number* is treated as condition, and job_number is treated as ALL:.

- When -B option and -b option are specified at the same time, the number of jobs becomes a number specified by -b option.
  In the numbers of jobs specified by -b option, the job of *job_number* which not specified with *job_condition* doesn't have a condition.
  If *job_number* is larger than the number of jobs specified by -b option, the syntax error will occur.

A maximum characters of *job_condition* is 255 bytes.

-l *resource*_list[,*resource_list*,...]

Set maximum value and warning value for *resource* limits with requests, jobs and processes. Two or more resource limits can be specified in one -l option by comma delimited.
The items to be specified as *resource_list* are as follows. Enclose with double quotes (") when both maximum and warning values are specified. Warning value is set to the same value as the maximum value if warning value is omitted.

Options for resource limits per request

elapstim_req=["]*max_limit*[,*warn_limit*"]

Limit by elapsed time after start of execution

Options for resource limits per job/logical host

cpunum_job=*max_limit*

Limit on the number of CPUs that can be used
In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

cputim_job=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

gpunum_job=*max_limit*

Limit on the number of GPUs that can be used

memsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VH.

socknum_job=*max_limit*

Limit of socket number

In case of SX-Aurora TSUBASA, it is the number of sockets used on VH.

This resource can be specified when the queue's numa control feature is enabled and submit_cpu_unit is set to "any" or "socket".

It cannot be specified with cpunum_job.

vmemsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

The above are legacy options for setting maximum and warning values for resource limits per job/logical host, and these options will not be expanded in the future. When setting maximum and warning values for resource limits per job/logical host, it is recommended to use the "Options for resource limits per logical host" (ex.: --cpunum-lhost) described below.

Options for resource limits per process

coresz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum core file size that can be created

cputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

datasz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum data segment size that can be used

In case of SX-Aurora TSUBASA, it is the data size used on VH.

filenum_prc=*max_limit*

Limit on the number of files that can be opened simultaneously

filesz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum file size that can be created

stacksz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum stack size that can be used

In case of SX-Aurora TSUBASA, it is the stack size used on VH.

vecputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by VE CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

vememsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum VE memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VE.

vmemsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

For interactive request, the following limitation function is supported.

idle_timer=*max_limit*

Idle timer in minutes.

Limit values are sometimes converted into values that are convenient to the machine.

-m *mail_options*

Sends e-mail about change of interactive request state.

The following can be specified in *mail_options*:

a	E-mail is sent when the request is abnormally terminated.
b	E-mail is sent when the request is started.
e	E-mail is sent when the request is terminated.
n	E-mail is not sent. 'n' cannot be specified with other mail_options.

-M *mail_list*

Specify the user list for sending e-mails. The form of *mail_list* is as follows:

mail_address[,*mail_address*, ...]

The maximum length of the *mail_list* that can be specified is 1023 bytes.
If this option is not specified, The default setting of *mail_list* is the submitting user at the qlogin host.

-N *name*

Specify interactive request name. Maximum characters of *name* is 63 bytes.
When -N option is not specified, the default request name is "QLOGIN".

-q *destination*

Specify the interactive queue name to submit the interactive requests. In case this option is not specified, the default interactive queue (specified by environment variable PBS_INTERACTIVE_QUEUE) will be used.

-R

Display only the version of the command and NQSV API.

-s *request_start_time*

Submitted interactive request will start at the time specified by *request_start_time*.
This option is available only if the scheduler is JobManipulator. Please refer to [JobManipulator] for details.
The form for *request_start_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The each two digits of *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-S *path_name*

Specify the shell to execute on the execution host. If -S option is not specified, the default shell executed on the execution host is the user's login shell.

-T *job_topology*

Specify the mode of jobs to be executed. The following can be specified in *job_topology*.

distrib

Distributed job. Number of jobs is always 1 even if -b option is specified or not.

intmpi

intmpi job. This option has to be specified when executing IntelMPI programs in the interactive request.

mvapich

Batch jobs to be executed will be mvapich jobs.

This option has to be specified when executing mvapich programs in the batch job.

necmpi

necmpi job. This option has to be specified when executing NEC MPI programs in the interactive request.

openmpi

openmpi job. This option has to be specified when executing OpenMPI programs in the interactive request.

pltmpi

pltmpi job. This option has to be specified when executing Platform MPI programs in the interactive request.

If this option is not specified, the mode of interactive jobs to be executed will be "distrib".

-U *user_attribute_list*

Specify the optional character string to the interactive request. The format of *user_attribute_list* is as follows.

user_attribute[=value][,*user_attribute*[=*value*], ...]

The total character string length of *user_attribute_list* must be 4000 bytes or less. If the existing name is specified for *user_attribute*, the *value* will be overwritten. If *value* is omitted, the *value* will be null character.
Multiple *value* must be enclosed in double quotes (").

user_attribute="value1,value2"

The string specified by this option has no influence on the behavior of the request.

-v *variable_list*

Specify the environment variables that are used in executing the interactive request. The form of *variable_list* is as follows:

variable[=*value*][,*variable*[=*value*], ...]

The total length for character string of *variable_list* must be 4000 bytes or less. If the existing variable name is specified for variable, the value will be overwritten.
Multiple *value* must be enclosed in double quotes (").

variable="value1,value2"

The following environment variables are set by default.
The value can be changed by this option except PBS_ENVIRONMENT, PBS_JOBID, PBS_SUBREQNO, PBS_JOBNAME and PBS_NODEFILE.

PBS_ENVIRONMENT

"PBS_INTERACTIVE" is set. It means the request is interactive type.

PBS_JOBID

Sets the job identifier of the request.

PBS_JOBNAME

Sets the request name.

PBS_NODEFILE

Sets a path to the file of execution host list of all jobs in the request.

PBS_O_HOME

Sets Environment Variable "HOME" in the client.

PBS_O_HOST

Sets the client host name.

PBS_O_LANG

Sets Environment Variable "LANG" in the client host.

PBS_O_LOGNAME

Sets Environment Variable "LOGNAME" in the client host.

PBS_O_PATH

Sets Environment Variable "PATH" in the client host.

PBS_O_MAIL

Sets Environment Variable "MAIL" in the client host.

PBS_O_SHELL

Sets Environment Variable "SHELL" in the client host.

PBS_O_TZ

Sets Environment Variable "TZ" in the client host.

PBS_O_WORKDIR

Sets the path name of the current working directory in the client host.

-V

Exports all environment variables of qlogin process to the environment for interactive request execution.

-W

When execution hosts cannot be assigned to the request to start execution immediately, qlogin waits until execution hosts are assigned by scheduler. If -W option is not specified, qlogin terminates immediately when execution hosts cannot be assigned.

While waiting execution hosts to be assigned, qlogin is interrupted by ctrl-C, The submitting interactive request is deleted and the qlogin terminates.

-W option is effective only when the "Real Time Scheduling" attribute of the submitted interactive queue is set to "manual". If the queue is not set to "manual", qlogin's wait behavior is define by the queue's setting.

-X

Enables X11 forwarding.

--custom *cr_name*=*n*[,*cr_name*=*n...*]

Specify the consumption of custom resource specified by *cr_name.*

Multiple custom resources can be specified with delimiting by the comma.

Integer value of 1 to 2147483647 or unused(0) can be specified for *n*.

Submitting with unused(0) is allowed for only allowed queue.

--enable-cloud-bursting{=[*yes* | *no*]}

Specifies whether to burst and execute request to cloud computing resources. If this option is set to *yes*, bursting will be performed. If this option is set to *no*, bursting will not be performed. If this option is not specified, the default behavior is *no*.

--exclusive [= *host*]

Specify the request to execute as 1 logical host (job) per 1 host (exclusive execution).

Specify the parameter "*host*". This "*host*" parameter can omit.

This option can't be used with template (--template) option of provisioning function. It will be error when it used at same time.

--group=*group_name*

Specify the group name by which a request is executed. Specify group name to *group_name*.

When this option is not specified, a request is executed by the group on executing request submission command.

In case the Designated Group Execution Function of request is set to OFF, this option is ignored.

--job-separator

Works as a delimiter line between jobs. It can be abbreviated as "---".

--mig=*gi-name*[:*ci-slice-count*[,*ci-slice-count...*]]

Specify the multi instance GPU (MIG).

Specify GPU instance (GI) profile name to *gi-name*. qstat -Ef shows GI profiles to be able to specify.

Specify *ci-slice-count* if specified GI is further divided into compute instances (CI).

If no *ci-slice-count* is specified, a CI of the same size as the GI profile is created.

[Specification Example]

--mig=4g.20gb:1,1,2

In this case the GI profile as 4g.20gb is used and the GI is divided into 3 CIs.

If multiple --mig options are specified, multiple GIs can assign to the request. If both --mig and --gpunum-lhost are specified, only --mig is applied and --gpunum-lhost is ignored.

--template=*template_name*

Specify the template of job environment with provisioning.

cpunum_job, memsz_job and gpunum_job of -l option cannot be specified with this option.

--use-hca=*num*

--use-hca=[{io | mpi | all}:*num*][,{io | mpi | all}:*num...*]

It is possible to specify the port number of HCA according to the kind of HCA (for ScaTeFS direct IO, for MPI and both for ScaTeFS direct IO and for MPI).

For it,　you have to specify "io", "mpi" or "all" with colon before the "num".

If only "num" is specified, it is assumed that "all" is specified.

This option is effective in the request which use VE.

--userpp-script=*location*:*path*[,*location*:*path*]

Execute the UserPP script specified by *path* at the *location.*

Following value can be specified for *location*.

prr	Execute location is PRE-RUNNING.
por	Execute location is POST-RUNNING.

--venode=*total_venum*

VE nodes which is specified by total_venum are assigned to a request. The number of logical host is calculated automatically using submitted queue's default number of VEs. You must specify a value 1 or greater. If "--venum-lhost" is specified the value which is specified by this option is used instead of the queue's default number of VEs. This option and the specification of the number of job (number of logical host) by -b option cannot be specified simultaneously.

--warning-signal=elapstim:*signal*

Specifies a sending signal at exceeding the warning value of the limit by elapsed time after start of execution. *signal* can be specified in the following form:

Signal name: (Example: SIGHUP)
Signal name with "SIG" removed: (Example: HUP)
Signal number: (Example: 1) For detailed information. refer to signal(7).

Default is SIGTERM if this option is omitted.

Options for resource limits per logical host

You can specify maximum or warning value for resource limits per logical host using following options. The logical hosts are generated according to designation of "-b job_count" at submitting of a request. Warning value is set to the same value as the maximum value if warning value is omitted.
You can also set the maximum and warning value for resource limits per logical host with -l option (ex.: -l cpunum_job), but since -l is a legacy setting and will not be expanded in the future, it is recommended to use the following options.
Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--cpunum-lhost=*max_limit*

Specify the limit on the number of CPUs per logical host. This option corresponds to -l cpunum_job=max_limit (Limit on the number of CPUs that can be used(Job)). In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

--cputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of CPU time per logical host. This option corresponds to -l cputim_job=max_limit Limit of CPU time (Job)). In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

--gpunum-lhost=*max_limit*

Specify the limit on the number of GPUs that can be used per logical host. This option corresponds to -l gpunum_job=max_limit (Limit on the number of GPUs that can be used (Job)).

--memsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum memory size that can be used per logical host. This option corresponds to -l memsz_job=max_limit (Limit on maximum memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the memory size used on VH.

--vecputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per logical host. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per logical host. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

--vmemsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum virtual memory size per logical host. This option corresponds to -l vmemsz_job=max_limit (Limit on maximum virtual memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

--venum-lhost =*max_limit*

Specify the limit on number of VE node per job.

Options for resource limits per venode

You can specify maximum or warning value for resource limits per VE node using following options. Warning value is set to the same value as the maximum value if warning value is omitted.

Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--vecputim-venode=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per VE node. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-venode=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per VE node. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

FORM OF RESOURCE LIMIT VALUE

Time Limit

Limits on time are specified in the following form:

[[*hours*:]*minutes*:]*seconds*[.*fraction*]

Specify the character string "unlimited" for unlimited.
Maximum limit of default is 2^31-1 seconds.

[Specification Example]

1234:58:21.29	1234 hours 58 minutes and 21.29 seconds
59:01	59 minutes and 1 second
12345	12345 seconds
121.1	121.1 seconds

Size Limit

Specify size limits in the following form:

integer[.*fraction*][*units*]

The following units can be specified in "units".

b	Byte
kb	Kilobyte (1024 b)
mb	Megabyte (1024 kb)
gb	Gigabyte (1024 mb)
tb	Terabyte (1024 gb)
pb	Petabyte (1024 tb)
eb	Exabyte　(1024 pb)

Failing to specify in units will be construed as having specified byte. Specify the character string "unlimited" for unlimited. Maximum limit of default is 2^31-1 bytes.

[Specification Example]

1234	1234Bytes
1234kb	1234Kilobytes
1234.5gb	1234.5Gigabytes

ENVIRONMENT VARIABLES

PBS_INTERACTIVE_QUEUE

The default queue that interactive request is submitted when -q option is omitted.

NQSV_INTERACTIVE_IP or NQSII_INTERACTIVE_IP

Client host's IP address (IPv4 numbers-and-dots notation).

SEE ALSO

qsub(1), qdel(1), qstat(1)

NOTES

1. In case MPI job is specified with -T option, interactive request session is not automatically terminated by idle timer.
2. To specify a character used as a delimiter of options (such as " in -I option) as a normal character, escape.

1.10. qmove(1)

NAME

qmove -- Move Request between Queues

SYNOPSIS

qmove [-f [-w] [-s]] [-P *privilege*] [-V] *o-destination request_identifier* ...

qmove [-f [-w] [-s]] [-P *privilege*] [-V] [-q *i-destination*] *o-destination*

DESCRIPTION

Moves requests specified by *request_identifier* to destination queue. Or moves all requests on the queue specified by *i-destination* to *o-destination*.
Requests who have no jobs can be moved in the following three states:

QUEUED, WAITING and HELD.

Parametric request can be moved only if it is in the state before creating sub-requests.
Sub-request cannot be moved.

The following options can be specified:

-f

The request who has any jobs is forced to move after rerunning the request and deleting jobs. If specified request has no jobs, this option is ignored. If the queue that the specified request belongs to is the same as the queue that is specified by *o-destination*, the request is not rerunning.
For parametric request, all sub-requests are deleted first and sub-requests are re-created from the beginning in the new queue if any sub-request has already be created at the qmove.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege

s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-q *i-destination*

Moves all requests that are residing in *i-destination*.
Omitting this option will move all requests residing in queues specified by Environment Variable PBS_QUEUE.

-s

It is used with -w option at the same time and does not display '.'(period) during waiting for completing moving of the specified requests with -w option. It is ignored when it is used with options other than -w option.

-V

Display only the version of the command and NQSV API.

-w

It is used with -f option at the same time and waits for completing moving of the specified requests. If it is specified more than one *request_identifier* and is used with -q option at the same time, it waits for completing the request one by one and is moved in order.
It is ignored when it is used with options other than -f option.

ENVIRONMENT VARIABLES

PBS_QUEUE

Queue name in default

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege

PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qstat(1),qsub(1)

NOTES

1. If -f option was specified and a request which was controlled by JobManipulator was moved, the request is taken off the map and reassigned at the next scheduling interval. Therefore, the request execution start time is changed.
2. The following case is occurred during waiting for the completion of queue moving by -w option, waiting state is continued until the next state transition is occurred by the scheduler.
   - The request which is EXITING state accepts a request of RERUN from the other API client.
   - An error is occurred at queue moving process after the request was rerun by -f option.

1.11. qrerun(1)

NAME

qrerun -- Rerun Request

SYNOPSIS

qrerun [-P *privilege*] [-V] [-g *grace*] *request_identifier* ...

DESCRIPTION

Terminates a request specified by *request_identifier* and returns the state to a state waiting for execution without jobs.
The signal aborts the executing jobs on the execution host.
Regardless of the status, the request with jobs can be backed to the QUEUED status.

When a request is rerun, this request will be stopped execution and re-submitted to the queue where the request was running. The same request ID previously allocated will be allocated as well.
When a parametric request is specified by *request_identifier*, all created and running sub-requests are deleted and the parametric request is re-submitted.

The following options can be specified.

-g *grace*

First, SIGTERM is sent to jobs residing in execution hosts. After pausing for seconds specified by *grace*, SIGKILL will be sent. If *grace* is smaller than 0, only SIGKILL will be sent immediately.
If this option is not specified, SIGKILL will be sent after waiting 5 seconds.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest.

When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.

If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qstat(1), qsub(1)

1.12. qrls(1)

NAME

qrls -- Release Previously Held Request

SYNOPSIS

qrls [-P *privilege*] [-V] *request_identifier* ...

DESCRIPTION

Releases a holding state of requests held by Command qhold.
Holding of requests can be released only when requests are held.

If a request is released from HELD state, it will return to the state before it was held.

For example, if the request was in QUEUED state before it was held, the request will be returned to QUEUED state and if in WAITING state before the request was held, it will be returned to WAITING state.

The following options can be specified:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qhold(1), qstat(1), qsub(1)

1.13. qrsh(1)

NAME

qrsh -- Submit Remote execution type Interactive request

SYNOPSIS

qrsh [-q *queue-name*] -c *remote_cmd-line* [*options*]

DESCRIPTION

Execute a command line specified by *remote_cmd-line* as an interactive request that is submitted to the queue specified by *queue-name*
The standard input of qrsh command is passed to the remote-execution command and the standard output/error of the remote-execution command is printed to qrsh's terminal.
qrsh command terminates with the exit code of the remote-execution command.

-c *remote_cmd-line*

The command line specified by *remote_cmd-line* is executed remotely.

-q *queue-name*

The remote-execution command is submitted as an interactive request to the queue specified by *queue-name*.
When -q is omitted, the default queue is can be set by PBS_INTERACTIVE_QUEUE environment variable.

And other options are supported as follows.

-a *date_time*

Set the interactive request scheduling start time.

The form for *date_time* is as follows:

[[[[*CC*] *YY*]*MM*]*DD*]*hhmm*[.*SS*]

Each two digits of *CC*, *YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year

MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-A *account_string*

Set the account code of submitted interactive request to *account_string*.
The maximum number of characters which can be specified for *account_string* is 15 bytes.
When the qrsh command is executed with no -A option, the account code name which is described in the $HOME/.acct is used as an account code. The format of $HOME/.acct is as following.

- The character string from the first character to the first space or tab or return is recognized as an account code name.

- The length of account code name specified in $HOME/.acct should not exceed 15 bytes.

When this option is not specified and there is no $HOME/.acct, account code will not to be specified.
When account code is specified in either the option or $HOME/.acct, qrsh does not check if it is an available account code.

-b *job_count*

Specify the number of jobs to be executed. When this option is not specified, the number of jobs to be executed will be 1. If type of jobs is set to "distributed" (-T distrib), only 1 can be set to *job_count*.

-B ["]*job_condition*[","*job_condition*"...]

Specify the conditions to select jobservers when the scheduler allocates jobservers to interactive job (Job condition). Please refer to JobManipulator's manual for details of scheduling by this specification.

The form of *job_condition* is as follows.

[*job_number*:]*condition*

- *job_number* is the job number of the request (0 or more).

- When the *job_number* of job_condition is omitted or "ALL" is specified for *job_number*, the *job_condition* is effective to all jobs.

- The following can be specified for *job_number*.

(1) Single number	0:
(2) All jobs specified by "ALL"	ALL:
(3) Two or more numbers using delimiters ","	0,2,5:
(4) Consecutive numbers using "-"	0-4:
(5) Combination of (3) and (4)	0,2,4-6:

- *job_number* is followed by scheduler's condition sentence (*condition*), separated by ":".

- The string which does not apply to the specified conditions of *job_number* is treated as condition, and *job_number* is treated as ALL:.

- When -B option and -b option are specified at the same time, the number of jobs becomes a number specified by -b option.
  In the numbers of jobs specified by -b option, the job of *job_number* which not specified with job_condition doesn't have a condition.
  If job_number is larger than the number of jobs specified by -b option, the syntax error will occur.

A maximum characters of job_condition is 255 bytes.

-l *resource_list*[,*resource_list*,...]

Set maximum value and warning value for resource limits with requests, jobs and processes. Two or more resource limits can be specified in one -l option by comma delimited.
The items to be specified as *resource_list* are as follows. Enclose with double quotes (") when both maximum and warning values are specified. Warning value

is set to the same value as the maximum value if warning value is omitted.

Options for resource limits per request

elapstim_req=["]*max_limit*[,*warn_limit*"]

Limit by elapsed time after start of execution

Options for resource limits per job/logical host

cpunum_job=*max_limit*

Limit on the number of CPUs that can be used

In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

cputim_job=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

gpunum_job=*max_limit*

Limit on the number of GPUs that can be used

memsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VH.

socknum_job=*max_limit*

Limit of socket number

In case of SX-Aurora TSUBASA, it is the number of sockets used on VH.

This resource can be specified when the queue's numa control feature is enabled and submit_cpu_unit is set to "any" or "socket".

It cannot be specified with cpunum_job.

vmemsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

The above are legacy options for setting maximum and warning values for resource limits per job/logical host, and these options will not be expanded in the future. When setting maximum and warning values for resource limits per job/logical host, it is recommended to use the "Options for resource limits per logical host" (ex.: --cpunum-lhost) described below.

Options for resource limits per process

coresz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum core file size that can be created

cputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

datasz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum data segment size that can be used

In case of SX-Aurora TSUBASA, it is the data size used on VH.

filenum_prc=*max_limit*

Limit on the number of files that can be opened simultaneously

filesz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum file size that can be created

stacksz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum stack size that can be used

In case of SX-Aurora TSUBASA, it is the stack size used on VH.

vecputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by VE CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

vememsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum VE memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VE.

vmemsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Limit values are sometimes converted into values that are convenient to the machine.

-m *mail_options*

Sends e-mail about change of interactive request state.

The following can be specified in *mail_options*:

a	E-mail is sent when the request is abnormally terminated.
b	E-mail is sent when the request is started.
e	E-mail is sent when the request is terminated.
n	E-mail is not sent. 'n' cannot be specified with other mail_options.

-M *mail_list*

Specify the user list for sending e-mails. The form of *mail_list* is as follows:

mail_address[,*mail_address*, ...]

The maximum length of the mail_list that can be specified is 1023 bytes.
If this option is not specified, the default setting of *mail_list* is the submitting user at the qrsh host.

-N *name*

Specify interactive request name. Maximum characters of *name* is 63 bytes.
When -N option is not specified, the default request name is "QRSH".

-R

Display only the version of the command and NQSV API.

-s *request_start_time*

Submitted interactive request will start at the time specified by *request_start_time*.
This option is available only if the scheduler is JobManipulator.
Please refer to [JobManipulator] for details.

The form for *request_start_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The each two digits of *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-S *path_name*

Specify the shell to execute on the execution host. If -S option is not specified, the default shell executed on the execution host is the user's login shell.

-T *job_topology*

Specify the mode of jobs to be executed. The following can be specified in *job_topology*.

distrib

Distributed job. Number of jobs is always 1 even if -b option is specified or not.

intmpi

intmpi job. This option has to be specified when executing IntelMPI programs in the interactive request.

mvapich

mvapich job. This option has to be specified when executing mvapich programs in the interactive request.

necmpi

necmpi job. This option has to be specified when executing NEC MPI programs in the interactive request.

openmpi

openmpi job. This option has to be specified when executing OpenMPI programs in the interactive request.

pltmpi

pltmpi job. This option has to be specified when executing Platform MPI programs in the interactive request.

If this option is not specified, the mode of interactive jobs to be executed will be "distrib".

-U *user_attribute_list*

Specify the optional character string to the interactive request.

The format of *user_attribute_list* is as follows.

user_attribute[=*value*][,*user_attribute*[=*value*], ...]

The total character string length of *user_attribute_list* must be 4000 bytes or less. If the existing name is specified for 2 user_attribute, the value will be overwritten. Multiple *value* must be enclosed in double quotes (").

user_attribute="value1,value2"

If value is omitted, the value will be null character. The string specified by this option has no influence on the behavior of the request.

-v *variable_list*

Specify the environment variables that are used in executing the interactive request.

The form of *variable_list* is as follows:

variable[=*value*][,*variable*[=*value*], ...]

The total length for character string of *variable_list* must be 4000 bytes or less. If the existing variable name is specified for variable, the value will be overwritten. Multiple *value* must be enclosed in double quotes (").

variable="value1,value2"

The following environment variables are set by default.
The value can be changed by this option except PBS_ENVIRONMENT, PBS_JOBID, PBS_SUBREQNO, PBS_JOBNAME and PBS_NODEFILE.

PBS_ENVIRONMENT

"PBS_INTERACTIVE" is set. It means the request is interactive type.

PBS_JOBID

Sets the job identifier of the request.

PBS_JOBNAME

Sets the request name.

PBS_NODEFILE

Sets a path to the file of execution host list of all jobs in the request.

PBS_O_HOME

Sets Environment Variable "HOME" in the client.

PBS_O_HOST

Sets the client host name.

PBS_O_LANG

Sets Environment Variable "LANG" in the client host.

PBS_O_LOGNAME

Sets Environment Variable "LOGNAME" in the client host.

PBS_O_MAIL

Sets Environment Variable "MAIL" in the client host.

PBS_O_PATH

Sets Environment Variable "PATH" in the client host.

PBS_O_SHELL

Sets Environment Variable "SHELL" in the client host.

PBS_O_TZ

Sets Environment Variable "TZ" in the client host.

PBS_O_WORKDIR

Sets the path name of the current working directory in the client host.

-V

Exports all environment variables of qrsh process to the environment for interactive request execution.

-W

When execution hosts cannot be assigned to the request to start execution immediately, qlogin waits until execution hosts are assigned by scheduler. If -W option is not specified, qlogin terminates immediately when execution hosts cannot be assigned.

While waiting execution hosts to be assigned, qlogin is interrupted by ctrl-C, the submitted interactive request is deleted and the qlogin terminates. -W option is effective only when the "Real Time Scheduling" attribute of the submitted interactive queue is set to "manual". If the queue is not set to "manual", qlogin's wait behavior is define by the queue's setting.

-X

Enables X11 forwarding.

--custom *cr_name*=*n*[,*cr_name*=*n...*]

Specify the consumption of custom resource specified by *cr_name.*
Multiple custom resources can be specified with delimiting by the comma.
Integer value of 1 to 2147483647 or unused(0) can be specified for *n*.
Submitting with unused(0) is allowed for only allowed queue.

--enable-cloud-bursting{=[*yes*|*no*]}

Specifies whether to burst and execute request to cloud computing resources. If this option is set to *yes*, bursting will be performed. If this option is set to *no*, bursting will not be performed. If this option is not specified, the default behavior is *no*.

--exclusive [= *host*]

Specify the request to execute as 1 logical host (job) per 1 host (exclusive execution).
Specify the parameter "*host*". This "*host*" parameter can omit.
This option can't be used with template (--template) option of provisioning function. It will be error when it used at same time.

--group=*group_name*

Specify the group name by which a request is executed. Specify group name to *group_name*.

When this option is not specified, a request is executed by the group on executing request submission command.

In case the Designated Group Execution Function of request is set to OFF, this option is ignored.

--job-separator

Works as a delimiter line between jobs. It can be abbreviated as "---".

--mig=*gi-name*[:*ci-slice-count*[,*ci-slice-count*...]]

Specify the multi instance GPU (MIG).

Specify GPU instance (GI) profile name to *gi-name*. qstat -Ef shows GI profiles to be able to specify.

Specify *ci-slice-count* if specified GI is further divided into compute instances (CI).

If no *ci-slice-count* is specified, a CI of the same size as the GI profile is created.

[Specification Example]

--mig=4g.20gb:1,1,2

In this case the GI profile as 4g.20gb is used and the GI is divided into 3 CIs.

If multiple --mig options are specified, multiple GIs can assign to the request. If both --mig and --gpunum-lhost are specified, only --mig is applied and --gpunum-lhost is ignored.

--template=*template_name*

Specify the template of job environment with provisioning.

cpunum_job, memsz_job and gpunum_job of -l option cannot be specified with this option.

--use-hca=*num*

--use-hca=[{io | mpi | all}:*num*][,{io | mpi | all}:*num...*]

It is possible to specify the port number of HCA according to the kind of HCA

(for ScaTeFS direct IO, for MPI and both for ScaTeFS direct IO and for MPI).

For it, you have to specify "io", "mpi" or "all" with colon before the "num".

If only "num" is specified, it is assumed that "all" is specified.

This option is effective in the request which use VE.

--userpp-script=*location*:*path*[,*location*:*path*]

Execute the UserPP script specified by *path* at the *location.*

Following value can be specified for *location*.

prr	Execute location is PRE-RUNNING.
por	Execute location is POST-RUNNING.

--venode=*total_venum*

VE nodes which is specified by total_venum are assigned to a request. The number of logical host is calculated automatically using submitted queue's default number of VEs. You must specify a value 1 or greater. If "--venum-lhost" is specified the value which is specified by this option is used instead of the queue's default number of VEs. This option and the specification of the number of job (number of logical host) by -b option cannot be specified simultaneously.

--warning-signal=elapstim:*signal*

Specifies a sending signal at exceeding the warning value of the limit by elapsed time after start of execution. *signal* can be specified in the following form:

Signal name: (Example: SIGHUP)
Signal name with "SIG" removed: (Example: HUP)
Signal number: (Example: 1) For detailed information. refer to signal(7).

Default is SIGTERM if this option is omitted.

Options for resource limits per logical host

You can specify maximum or warning value for resource limits per logical host using following options. The logical hosts are generated according to designation of "-b job_count" at submitting of a request. Warning value is set to the same value as the maximum value if warning value is omitted.

You can also set the maximum and warning value for resource limits per logical host with -l option (ex.: -l cpunum_job), but since -l is a legacy setting and will not be expanded in the future, it is recommended to use the following options.

Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--cpunum-lhost=*max_limit*

Specify the limit on the number of CPUs per logical host. This option corresponds to -l cpunum_job=max_limit Limit on the number of CPUs that can be used (Job)). In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

--cputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of CPU time per logical host. This option corresponds to -l cputim_job=max_limit Limit of CPU time (Job)). In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

--gpunum-lhost=*max_limit*

Specify the limit on the number of GPUs that can be used per logical host. This option corresponds to -l gpunum_job=max_limit (Limit on the number of GPUs that can be used (Job)).

--memsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum memory size that can be used per logical host. This option corresponds to -l memsz_job=max_limit (Limit on maximum memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the memory size used on VH.

--vecputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per logical host. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per logical host. In case of SX-Aurora TSUBASA, it is the virtual memory size used on VE.

--venum-lhost =*max_limit*

Specify the limit on number of VE node per job.

--vmemsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum virtual memory size per logical host. This option corresponds to -l vmemsz_job=max_limit (Limit on maximum virtual memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Options for resource limits per venode

You can specify maximum or warning value for resource limits per VE node using following options. Warning value is set to the same value as the maximum value if warning value is omitted.

Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--vecputim-venode=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per VE node. In case of SX-Aurora TSUBASA, it is the

CPU time used on VE.

--vememsz-venode=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per VE node. In case of SX-Aurora TSUBASA, it is the virtual memory size used on VE.

FORM OF RESOURCE LIMIT VALUE

Time Limit

Limits on time are specified in the following form:

[[*hours*:]*minutes*:]*seconds*[.*fraction*]

Specify the character string "unlimited" for unlimited.
Maximum limit of default is 2^31-1 seconds.

[Specification Example]

1234:58:21.29	1234 hours 58 minutes and 21.29 seconds
59:01	59 minutes and 1 second
12345	12345 seconds
121.1	121.1 seconds

Size Limit

Specify size limits in the following form:

integer[.*fraction*][*units*]

The following units can be specified in "units".

b	Byte
kb	Kilobyte (1024 b)
mb	Megabyte (1024 kb)
gb	Gigabyte (1024 mb)
tb	Terabyte (1024 gb)
pb	Petabyte (1024 tb)
eb	Exabyte (1024 pb)

Failing to specify in units will be construed as having specified byte. Specify the character string "unlimited" for unlimited. Maximum limit of default is 2^31-1 bytes.

[Specification Example]

1234	1234Bytes
1234kb	1234Kilobytes
1234.5gb	1234.5Gigabytes

ENVIRONMENT VARIABLES

PBS_INTERACTIVE_QUEUE

The default queue that interactive request is submitted when -q option is omitted.

NQSV_INTERACTIVE_IP or NQSII_INTERACTIVE_IP

Client host's IP address (IPv4 numbers-and-dots notation).

SEE ALSO

qlogin(1), qdel(1), qstat(1)

NOTES

1. To specify a character used as a delimiter of options (such as " in -I option) as a normal character, escape.

1.14. qsig(1)

NAME

qsig -- Send Signal to Request

SYNOPSIS

qsig [-s *signal*] [-P *privilege*] [-V] *request_identifier* ...
qsig [-s *signal*] [-P *privilege*] [-V] *job_identifier* ...

DESCRIPTION

Sends a *signal* to requests (jobs) that are specified by *request_identifier* or *job_identifier*.

The following options can be specified:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-s *signal*

Specifies a signal to be sent to requests (jobs). *signal* can be specified in the following form:

- Signal name: (Example: SIGHUP)
- Signal name with "SIG" removed: (Example: HUP)
- Signal number: (Example:1) For detailed information, refer to signal(7).

If option -s is omitted, Signal SIGTERM will be sent.

-V

Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qdel(1), qstat(1), qsub(1)

EXAMPLE

Suspend a request

```
$ qsig -s SIGSTOP 72.host1
```

Resume a request

```
$ qsig -s SIGCONT 72.host1
```

NOTES

If the following request that has been suspended by JobManipulator was resumed by Manager *privilege*, the resource confliction may be occurred because the resumed request is restarted soon.

- Suspended by an interruption.
- Suspended from smgr command.

1.15. qstat(1)

NAME

qstat -- Display The Status of NQSV

SYNOPSIS

qstat [-P *privilege*] [-V] [-f] [-n] [-l] [-d] [-m] [-s] [-a] [-u *userlist*] [-q *quelist*]
[-c *cpumode*] [-F *Item_list*] [-o *Item_list*] [-O *Item_list*] [--group[=*group_name*]]
[--adjust-column] [--planned-start-time] [*request_identifier* ...]

qstat -B [-P *privilege*] [-V] [-f] [-n] [-l] [-d] [-F *Item_list*] [-o *Item_list*] [-O *Item_list*]
[--adjust-column] [*batch_server_host* ...]

qstat -B -L -P m [-V] [-n]

qstat -D [-P *privilege*] [-V] [-f] [-n] [-l] [-F *Item_list*] [-o *Item_list*] [-O *Item_list*]
[--adjust-column] [*scheduler_identifier* ...]

qstat -E [-P *privilege*] [-V] [-f] [-n] [-l] [-t] [-g *node_group*] [-F *Item_list*] [-o *Item_list*]
[-O *Item_list*] [--adjust-column] [*execution_host* ...]

qstat -G [-P *privilege*] [-V] [-f] [-n] [-l] [-F *Item_list*] [-o *Item_list*] [-O *Item_list*]
[*node_group* ...]

qstat -J [-P *privilege*] [-V] [-f] [-n] [-l] [-d] [-e] [-t] [-m | -c *cpumode*] [-u *userlist*] [-h
execution_host]
[-F *Item_list*] [-o *Item_list*] [-O *Item_list*] [--adjust-column] [*job_identifier* ...]

qstat -Q [-P *privilege*] [-V] [-f] [-n] [-l] [-d] [-e] [-i] [-r] [-N] [-t] [-F *Item_list*]
[-o *Item_list*] [-O *Item_list*] [--group=*group_name*] [*destination* ...]

qstat -R [-P privilege] [-V] [-f] [-n] [-l] [-F *Item list*] [-o *Item list*] [-O *Item list*]
[--group[=*group_name*]] [--adjust-column] [*parametric_request_identifier* ...]

qstat -S [-P *privilege*] [-V] [-f] [-n] [-l] [-t] [-h *execution_host*] [-g *node_group*]
[-F *Item_list*] [-o *Item_list*] [-O *Item_list*] [--adjust-column] [*job_server_number* ...]

qstat -T [-P *privilege*] [-V] [-f] [-n] [-l] [-u *userlist*] [-F *Item_list*] [-o *Item_list*] [-O *Item_list*]
[--adjust-column] [*request_identifier* ...]

qstat --custom [-P *privilege*] [-V]

qstat --limit [-P *privilege*] [--group=*group_name*] [-V]

qstat --template [-P privilege] [-V] [-f] [-n] [-l] [--VE] [template_name ...]

qstat --venode [-P *privilege*] [-V] [*VI_hostname ...*]

qstat --cloud_template [-P privilege] [-V] [-f] [-n] [-l] [template_name ...]

DESCRIPTION

qstat command displays various information of NQSV system. qstat has the following options to specify output information.

-B	Batchserver information
-D	Scheduler information
-E	Execution host information
-G	Node group information
-J	Job information
-Q	Queue information
-R	Parametric request summary information
-S	Job server information
-T	Network request information
--custom	Custom resource information
--limit	Limit information per group and user
--template	Template information
--venode	VE node information
--cloud_template	Cloud bursting template information

Without any options decribed above, qstat displays the request information.

1. **Request information.**

Request information will be displayed if no display information option (B,D,E,G,J,Q,R,S,T,limit) is specified.

Specify a request identifier in *request_identifier* in the following form:

[*request_number*] [.*batch_server*] [@*batch_server_host*]

request_number : Request number

When the request is a parametric request, [] or [*n*] is displayed after request_number. *n* is a sub-request number.

batch_server : Batch Server Name

batch_server_host : Batch Server Host Name

If request_number is omitted, all requests owned by command executer will be the objects.

If batch_server is omitted, objects will be batch servers in *batch_server_host*.

If batch_server_host is omitted, objects will be batch server hosts in default.

Batch server hosts in default will be batch server hosts in /etc/opt/nec/nqsv/api_client.conf.

If not specified, local host will be the hosts.

If request_identifier is omitted, all requests owned by command executer in the batch server host in default will be the objects.

The following options can be specified:

-a

Displays the request is attached or not.

-c *cpumode*

Display CPU time according to *cpumode*.

0 Display present CPU time of use (CPU)

1 Display accumulated CPU time of use (ACCPU)

When -c option is not specified, qstat command displays present CPU time of use (CPU). Cpumode cannot be omitted when using this option. When -f option is specified, qstat command displays both values.

-e

Display the CPU time and Memory which is used on the VE. When -m or -c option is specified, the target CPU of -m or -c option is changed to the CPU on the VE.

-m

Display remaining elapsed time.

When -m option is not specified, qstat command displays elapsed time (Elapse). When -f option is specified, qstat command display this values.

-q *quelist*

Displays the requests submitted to the queue specified by *quelist.*

-s

For parametric requests, each sub-requests are displayed.

-u *userlist*

Targets the request which is owned by the user specified by *userlist*.

You can specify two or more login names in *userlist* if delimiting by "," without putting space.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

--group[=*group_name*]

Group name is displayed in summary information.

When group name is specified to *group_name* information of specified group is displayed.

--planned-start-time

Displays scheduled date and time to start execution of requests.

Time is output in the following format, which conforms to the extended ISO 8601.

The string "T" is a character indicating that the time value is to be continued.

```
yyyy-mm-ddThh:mm:ss
```

Of the options available for qstat, the following options can be used together, and all other options ignore --planned-start-time.

Options that can be used together:

[-P privilege] [-n] [-l] [-d] [-m] [-s] [-a] [-u userlist] [-q quelist]
[-c cpumode] [-o Item_list] [-O Item list] [--group[=group-name]]
[--adjust-column]

2. **Batch server information (-B option)**

Information of batch servers is displayed.
Specify an applicable batch server host name in *batch_server_host*.
If specification in *batch_server_host* is omitted, the batch server host in default will be the object.

The following option can be available.

-L

Display NQSV license setting information only.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

3. **Scheduler information (-D option)**

Scheduler information is displayed.
Specify the applicable scheduler in *scheduler_identifier* in the following form:

[*scheduler_ID*][@*batch_server_host*]

scheduler_ID	:	Scheduler ID
*batch_server_hos*t	:	Batch Server Host Name

If *scheduler_ID* is omitted, all scheduler residing in *batch_server_host* will be applicable.

If batch_server_host is omitted, the batch server host in default will be the host.

If *scheduler_identifier* is omitted, all scheduler on the default batch server host will be the object.

The following option can be specified:

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

4. **Execution host information (-E option)**

Information of execution hosts is displayed. Specify the hostname in *execution_host* in the following form:

[*execution_host_name*][@*batch_server_host*]

execution_host_name	:	Execution Host Name
batch_server_host	:	Batch Server Host Name

If *execution_host_name* is omitted, all execution hosts residing in *batch_server_host* will be the objects.

If batch_server_host is omitted, the batch server host in default will be the host.

The following options can be specified:

-g *node_group*

Displays the execution hosts which belongs to the node group specified by *node_group*.

-t

All execution hosts are displayed even if the job server is linked down. And some extra information is appended.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

5. **Node group information (-G option)**

-G option displays the node group information. To specify the target node group, specify the name of the node group by *node_group* in the following format.

node_group_name[@*batch_server_host*]

node_group_name Name of the node group *batch_server_host* Batch server host name

6. Job information (-J option)

Job information is displayed.

Specify the applicable job identifier in *job_identifier*.

The following options can be specified:

-c *cpumode*

Display CPU time according to *cpumode*.

0	Display present CPU time of use (CPU)
1	Display accumulated CPU time of use (ACCPU)

When -c option is not specified, qstat command displays present CPU time of use. Cpumode cannot be omitted when using this option.

When -f option is specified, qstat command displays both values.

-e

Display the CPU time and Memory usage of VE for the program in the job.

When -m option or -c option is specified with this option, the object CPU of -m option or -c option is changed to CPU on the VE.

-h *execution_host*

Specify execution hosts that will be the objects.

If an option is not specified, all execution hosts linked to the applicable batch server will be the objects. If this option is specified with -F, -F ehost must be specified.

-m

Display remaining CPU time.

Cannot specify this option and -c option at the same time.

When -f option is specified, qstat command displays this value.

-t

Display resource information per job in hybrid requests.

-u *userlist*

Target the request which is owned by the user specified by *userlist*.

You can specify two or more login names in *userlist* if delimiting by "," without putting space.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

7. Queue information (-Q option)

Displays queue information.

Specify applicable queues in *destination* in the following form:

[*queue_name*][@*batch_server_host*]

queue_name	:	Queue Name
batch_server_host	:	Batch Server Host Name

If *queue_name* is omitted, all queues residing in *batch_server_host* will be the objects.

If *batch_server_host* is omitted, the batch server host in default will be the object.

If *destination* is omitted, all queues in the batch server host in default will be the objects.

General user can display only queue with access privilege of the user.
In case Designated Group Execution Function of request is set to ON, it is checked whether group at execution of qstat or group which is specified by --group option has an access privilege of queue.
In case Designated Group Execution Function of request is set to OFF, this option is to be ignored.
The following options can be specified:

-e

Batch queues will be the objects.

-i

Interactive queues will be displayed.

-N

Network queues will be the objects.

-r

Routing queues will be the objects.

-t

Displays the number of all job servers bound to the queues including job servers which is linked down. (Without this option, the number of the jobservers is the count of the job servers which are linked up.)

--group=*group_name*

In case Designated Group Execution Function of request is set to ON, it is checked whether group which is specified by --group option has an access privilege of queue. In case Designated Group Execution Function of request is set to OFF, this option is to be ignored.

8. Parametric request summary information (-R option)

qstat -R option displays the summary information of the parametric requests. To display the target request, specify *parametric_request_identifier.*

The following options can be specified:

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

--group[=*group_name*]

Group name is displayed in summary information.
When group name is specified to *group_name* information of specified group is displayed.

9. Job server information (-S option)

Displays job server information.
Specify a job server name in *job_server* in the following form:

[*job_server_name*] [@*batch_server_host*]

job_server_name	:	Job Server Name
batch_server_host	:	Batch Server Host Name

If *job_server_name* is omitted, all job servers residing in *batch_server_host* will be the objects.

If *batch_server_host* is omitted, the batch server host in default will be the object.

The following options can be specified:

-g *node_group*

Displays the jobservers which belong to the specified node group.

-h *execution_host*

Specify execution hosts that will be the objects.
If an option is not specified, all execution hosts linked to the applicable batch server will be the objects.

-t

All job servers are displayed even if the ones are linked down.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

10. Network request information (-T option)

Network request information will be displayed.

Request_identifier specifies a parent request identifier of a network request.

The following options can be specified:

-u *userlist*

Target the request which is owned by the user specified by *userlist.*

You can specify two or more login names in *userlist* if delimiting by "," without putting space.

--adjust-column

When the information couldn't be indicated with established column width, the column width was expanded automatically according to the max length.

11. Custom resource information (--custom option)

Custom resource information will be displayed.

12. Limit Information per Group and User (--limit option)

In case that --limit is specified, information of limits which can be set by specifying group name and user name individually is displayed.

Group manager can display limit information of only the group of which the manager has privilege.

General user can display limit information of only user and group executing qstat command.

The following options can be specified.

--group=*group_name*

In case Designated Group Execution Function of request is set to ON, qstat command is executed by the group specified with *group_name.* In case Designated Group Execution Function of request is set to OFF, this option is to be ignored.

13. Template information (--template)

Template information will be displayed.

When "--VE" option is specified, the number of VEs and the number of HCA ports are displayed instead of the number of GPUs.

When "--cloud_template" option is specified, the information of the template for cloud bursting will be displayed.

14. VE node information

When "--venode" option is specified, the information of VEs incorporated
in VI(VH) is displayed. JSV on such VI have to be linked up.

15. Cloud template information (--cloud_template)

The information of the template for cloud bursting will be displayed.

The following options common to each display information option can be specified.

-d	Change display format of time to d+hh:mm:ss
-f	Display detail information
-l	Display all content without truncating to keep within specified length
-n	Suppress header output

-P *privilege*

Specify access privilege on executing command. The following *privilege* can be specified.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is
lowest.
With Special user privilege, request information of other users as well as information displayed with User privilege can be displayed.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V Display version of command and version of NQSV API in use and then exit.

-F *item*[,*item*,...]

Customize output format according to each item of display option such as B, D, E, G, J, Q, S, T and no option.

[No option for display information]:
rid reqnm own group quenm pri stt stall mem cpu acpu elaps relaps rflg hflg mflg jobs ehost sdate qdate rdate att pdate

[-B]: bsvhost mid umax gmax rrlm nrlm tot arr wai que run ext hld sus bstt

[-D]: schid schnm schost ques quenm

[-E]: ehost bsvhost osnm rel hwnm ldavg1 cpuavg1 ucpu fcpu umem1 fmem1 uswap1 fswap1 quenm ve

[-G]: ngrpnm ngrptype bsvhost comment jsvs quenm

[-J]: jno rid jid ejid mem cpu acpu rcpu jsvno ehost euser ecode vecpu avecpu rvecpu

[-J -t]: jno rid venum cpunum gpunum cputim memsz vmemsz

[-Q [-e]]: quenm schid jsvs stt1 stt2 pri qtot arr wai que prr run por ext hld hol sus mig stg ehost attbl

[-Q -r]: quenm stt1 stt2 pri rqrlm qtot arr wai que hld trs

[-Q -N]: quenm stghost stt1 stt2 pri nqrlm qtot wai que run

[-Q -i]: quenm schid jsvs stt1 stt2 pri qtot arr wai que prr run por ext hld sus ehost attbl

[-R]: rid reqnm own group quenm pri stt rflg hflg mflg jobs tot active done

[-S]: jsvno jsvnm bsvhost ehost link bind quenm jobs ldavg1 cpuavg1

[-T]: rid own pri stgdir stgno stgf stt

Each items mean following table.

Item	Meaning	Target option
acpu	accumulated CPU time from the starting the request	(no)
	accumulated CPU time from the starting the job	-J
active	number of the existing sub-requests	-R
arr	number of requests for arriving state	-B, -Q -e, -Q -r, -Q -i
att	the request is attached or not	(no)
attbl	possibility of qattach	-Q -e, -Q -i
avecpu	accumulated VE CPU time from the	-J

	starting the job	
bind	bind state	-S
bstt	batch server state	-B
bsvhost	batch server host name	-B, -E, -G, -S
comment	comment for the node group	-G
cpu	CPU time	(no), -J
cpuavg1	CPU average of most recent 1 minute	-E, -S
cpunum	limit of number of CPU per a job	-J -t
cputim	limit of CPU time per a job	-J -t
done	number of terminated sub-requests	-R
ecode	job exit code	-J
ehost	execution host name	(no), -B, -J, -Q -e, -Q -i, -S
ejid	execution job ID	-J
elaps	elapsed time	(no)
euser	job execution user name	-J
ext	number of requests for exiting state	-B, -Q -e, -Q -i
fcpu	number of free CPU	-E
fmem1	free physical memory size	-E
fswap1	free swap size	-E
gmax	limit value for input of requests per a group	-B
gpunum	limit of number of GPU per a job	-J -t
group	group name	(no), -R
hflg	holding possible or not	(no), -R
hld	number of batch requests for held state	-B, -Q -e, -Q -r, -Q -i
hol	number of requests for　holding state	-Q -e
hwnm	hardware name	-E
jid	job identifier	-J
jno	job number	-J, -J -t
jobs	number of jobs per a request	(no)
	number of jobs per a sub-request	-R
	number of jobs per a job server	-S
jsvnm	job server name	-S
jsvno	job server number	-J, -S
jsvs	number of job servers	-G, -Q -e, -Q -i
ldavg1	load average of most recent 1 minute	-E, -S

link	link state	-S
mem	memory size used by the request	(no)
	memory size used by the job	-J
memsz	limit of memory size per a job	-J -t
mflg	job migration possible or not	(no), -R
mid	machine ID	-B
mig	number of requests for migrating state	-Q -e
ngrpnm	name of the node group	-G
ngrptype	type of the node group	-G
nqrlm	maximum number of running network requests for each request	-Q -N
nrlm	network queue run limit	-B
osnm	name of operating system	-E
own	owner name	(no), -R, -T
por	number of requests for post-running state	-Q -e, -Q -i
pri	priority	(no), -R
	queue priority	-Q -e, -Q -r, -Q -N, -Q -i
	network request priority	-T
prr	number of requests for pre-running state	-Q -e, -Q -i
pdate	scheduled date and time to start execution	(no)
qdate	date and time when the request was submitted to the queue ※If the request was forwarded, the date and time when it was forwarded to the network queue	(no)
qtot	number of requests submitted in the queue	-Q -e, -Q -r, -Q -N, -Q -i
que	number of requests for queued state	-B, -Q -e, -Q -r, -Q -N, -Q -i
quenm	name of the queue resided by request	(no), -R
	list of bound the queue	-D, -E, -G, -S
	name of the queue	-Q -e, -Q -r, -Q -N, -Q -i
ques	number of bound the queue	-D
rcpu	remaining CPU time	-J

rdate	execution start date and time	(no)
rel	OS release number	-E
relaps	remaining elapsed time	(no)
reqnm	request name	(no), -R
rflg	re-execution possible or not	(no), -R
rid	request identifier	(no), -J, -J -t, -R
	parent request identifier	-T
rqrlm	maximum number of running requests for a routing queue	-Q -r
rrlm	routing queue run limit	-B
run	number of requests for running state	-B, -Q -e, -Q -r, -Q -N, -Q -i
rvecpu	remaining VE CPU time	-J
schid	scheduler identifier	-D, -Q -e, -Q -i
schnm	scheduler name	-D
schost	scheduler execution host	-D
sdate	date and time when the request was created	(no)
stall	stall status	(no)
stg	number of requests for staging state	-Q -e
stgdir	staging direction	-T
stgf	staging file	-T
stghost	host name at route-to destination of staging files	-Q -N
stgno	staging file number	-T
stt	request status	(no), -R
	network request status	-T
stt1	queue state (Submission factor)	-Q -e, -Q -r, -Q -N, -Q -i
stt2	queue state (Starting factor)	-Q -e, -Q -r, -Q -N, -Q -i
sus	number of requests for suspending, suspended and resuming state	-B, -Q -e, -Q -i
tot	number of requests managed by the batch server	-B
	total number of sub-requests	-R
trs	number of requests for transiting state	-Q -r
ucpu	number of used CPU	-E
umax	limit value for input of requests per a	-B

	user	
umem1	used physical memory size	-E
uswap1	used swap size	-E
ve	number of VE nodes	-E
vecpu	VE CPU time	-J
venum	limit of number of VE nodes per a job	-J -t
vmemsz	limit of virtual memory size per a job	-J -t
wai	number of requests for waiting state	-B, -Q -e, -Q -r, -Q -N, -Q -i

-o *item*[,*item*,...]

Sort by specified key, *item* in ascending order
If this option is specified more than once, the sorted results are further sorted according to the order of items specified. It is also possible to use in combination with -O option (in descending order).
The item to be specified is same as for -F.

-O *item*[,*item*,...]

Sort by specified key, *item* in descending order
If this option is specified more than once, the sorted results are further sorted according to the order of items specified. It is also possible to use in combination with -o option (in ascending order).
The item to be specified is same as for -F.

STANDARD OUTPUT

The following items are displayed in each list:

[Request Information]

- Request identifier (RequestID)
- Request name (ReqName)
- Owner (UserName)
- Group(GrpName) (When --group is Specified)
- Name of the queue (Queue)
- Priority (Pri)
- Request state (STT)

ARI	arriving
EXT	exiting
HLD	held
HOL	holding
MIG	migrating
POR	post-running
PRR	pre-running
QUE	queued
RUN	running
SUS	suspending, suspended, resuming
STG	staging
TRS	transferring
WAT	waiting

- Stall Status (S)

Y	Stall
-	Normal

- Memory size used by request (Memory)
- CPU time (Unit: second) (CPU) (When -c 0 is Specified)
- Accumulated CPU time (Unit: second) (ACCPU) (When -c 1 is Specified)
- Elapsed time (Unit: second) (Elapse)
- Remaining Elapsed time (Unit: second) (RElapse) (When -m is Specified)
- Re-execution Possible/Not Possible (R)

Y	Request with re-execution option
N	Request without re-execution option

- Hold Possible/Not Possible (H)

Y	Request with hold option
N	Request without hold option

- Job Migration Possible/Not Possible (M)

Y	Request with Job Migration option
N	Request without Job Migration option

- Number of jobs residing in request (Jobs)
- Presence of an attach connection(Att) (When -a is Specified)
- Scheduled start time (PlannedStartTime) (When --planned-start-time is specified)

[Request Information (Specify option -f)]

- Request identifier (Request ID)
- Request name (Request Name)
- Owner (User Name)

- Group (Group Name)
- Execution user ID (User ID)
- Group ID (Group ID)
- Submit user name (Submitted User) (When --user is Specified)
- Current state of request (Current State)

  (Stalled) displayed after state if a request is stalled state.
- State of request one before (Previous State)
- Time changed to current state (State Transition Time)
- Reason for change to current state (State Transition Reason)
- Error message if state is changed due to error (Error Message)
- Name of the queue resided by request (Queue)
- Job topology (Job Topology)
- Process Manager (Process Manager) (if batch request and Job topology is NEC MPI)
- Request priority (Request Priority)
- Output level of request log (Request Loglevel) (except interactive request)
- Re-execution possible/not possible (Rerunable)
- Holding possible/ not possible (Holdable)
- Client privilege during hold execution (Hold Type) (except interactive request)
- Job Migrating possible/ not possible (Migratable) (except interactive request)
- Client privilege at the suspended time (Suspend Type)
- Account code (Account Code)
- Name of standard output file (Stdout) (except interactive request)
- Name of standard output error file (Stderr) (except interactive request)
- Name of request log file (Reqlog) (except interactive request)
- Full path name of shell used (Shell)
- Forced shell (Restrict shell) (only for interactive request)
- E-mail address (Mail Address)
- Timing to send e-mail (Mail Option)
- Job Condition (Job Condition)
- Number of jobs residing in request (Number of Jobs)
- Date and time when request was created (Created Request Time)
- Date and time for start of scheduling (Entered Queue Time)
- Date and time that the request is planned to start (Planned Start Time)
- Date and time for execution of request (Execute Request Time)
- Date and time for start of request (Started Request Time)
- Date and time for end of request (Ended Request Time)
- Requested time to start the request execution (Requested Start Time)
- Requested time to end the request execution (Deadline Time) (except interactive

request)

- Umask value used octal (User Mask)
- Reservation ID (Reservation ID) (except interactive request)
- Request group number which belongs for request connection(Request Group Number) (except interactive request)
- Request ID of request submitted first for request connection (Lead Request) (except interactive request)
- Connected host of the interactive request (Interactive Host) (only for interactive request)
- Connecting port of the interactive request (Interactive Port) (only for interactive request)
- Idle timer of the interactive request (Idle Timer) (only for interactive request)
- Possibility of qattach (qattach command)
- The request is attached or not. (Attach)
- Cluster type selection. (Cluster Type Select)
- OpenStack template. (OpenStack Template) / Container template. (Container Template) / Cloud bursting template. (Cloud Template)

  CPU number used by the template (CPU Number)
  Memory size used by the template (Memory Size)
  GPU number used by the template (GPU Number)
  VE number used by the template (VE Number)
  HCA port number used by the template (HCA Number)
- UserPP script information. (UserPP Script)
- Exclusive execution. (Exclusive)
- Total number of VE nodes. (VE Node Number)
- Specified value of HCA. (HCA Number)
- Acceptance SIGTERM (Accept Sigterm)
- Cloud bursting availability status (Enable Cloud Bursting)
- User level check point script (Check Point Script) (except interactive request only when given check point script)
- Script for the user level restart (Restart Script) (except interactive request only when check point script is given)
- VE NUMA mode (VE NUMA Mode)
- Sending signal at exceeding warning value of the resource limitation (Warning Signal)
- Custom resource information. (Custom Resources)

  Custom resource name and it consumption.
- Execution Hosts information. (Execution Hosts(JSVNO))

Execution Host name and JSVNO.

- Resource Information (Resources Information)
  Memory size used by request (Memory)
  CPU time (Unit: second) (CPU Time)
  Accumulated CPU time (Unit: second) (Accumulated CPU Time)
  Elapsed time (Elapse)
  Remaining Elapsed time (Remaining Elapse)
  Virtual memory size used by request on Linux execution host (Virtual Memory)
  Assigned MIG Information (Multi Instance GPU Resources)
  Custom resource usage(custom resource name)
- Limit name and limit value information of logical host (Logical Host Resources)
  VE node number (VE Node Number)
  CPU number (CPU Number)
  GPU number (GPU Number)
  CPU time (CPU Time)
  Memory size (Memory Size)
  Virtual memory size (Virtual Memory Size)
  VE CPU time (VE CPU Time)
  VE memory size (VE Memory Size)
  Stdout file size (Stdout Size) (only batch request)
  Stderr file size (Stderr Size) (only batch request)
- Limit name and limit value information of VE node (VE Node Resources)
  VE CPU time (VE CPU Time)
  VE memory size (VE Memory Size)
- Resource limit name and limit value information (Resources Limits)
  Resource limit name and limit value for each request (Per_Req)
  Resource limit name and limit value for each job (Per_Job)
  Resource limit name and limit value for each process (Per_Prc)
- Kernel parameter name and value (Kernel Parameter)
- Scheduler Messages (Scheduler Message)
- User defined attributes (User Attributes)
- Staging file (Staging File)
  Stage-in file number (Stage-in File Number)
  Stage-out file number (Stage-out File Number)
  Place in transmitting origin (From)
  Place in transmitting *destination* (To)
  Status of staging (Status)

- Job Migration files (Migration File)

  (chkpnt_files) displayed if all of checkpoint files are targets of routing.

[Batch Server Host Information (Specify option -B)]

- Batch server host name (BatchServer)
- Machine ID (MachineID)
- Limit value for input of requests

UMAX	Per user
GMAX	Per group

- Run Limit

RRL	Routing queue run limit
NRL	Network queue run limit

- Number of requests managed by batch server (TOT)
- Number of requests for each state

ARR	arriving
WAI	waiting
QUE	queued
RUN	running
EXT	exiting
HLD	held
SUS	suspending

- State (Status)

[Batch Server Host Information (Specify option -B -f)]

- Batch server host name (Batch Server)
- NQSV version (NQSV Version)
- State (Batch Server State)
- Machine ID (Batch Server Machine ID)
- Log file path (Logfile Path)
- Log file output level (Logfile Level)
- Number of log files (Logfile Save Count)
- Log file size (Logfile MAX Size)
- Hostname of accounting server (Accounting Server Host Name)
- Port number of accounting server (Accounting Server Port Number)
- Directory of the job account files (Jacct directory)
- Setting of whether budget limitation is checked or not (Budget function)
- Request accounting and job accounting output flag (Request Accounting)
- Request accounting file path (Request Accounting File Path)

- Reservation accounting output flag (Reservation Accounting)
- Reservation accounting file path (Reservation Accounting File)
- Designated Group Execution Function of request(Specify Group for Request)
- Allow to use absolute path for staging file or not (Allow Absolute Exepath)
- Failure connected requests automatic deletion function (Auto Delete Failed Request)
- Number of requests managed by batch server (Total Request)
- Number of requests for each state (xxx Request)
- Heartbeat interval (Heart Beat Interval)
- Load information and HW resource loading information get interval (Load Interval)
- Job resource get interval (Get Resource Interval)
- Maximum sub-request entry for each parametric request (Max Subrequest Entry Limit)
- Routing queue run limit (Max Run Limit of Routing Queue)
- Routing Queue Retry Interval (Retry Interval of Routing Queue)
- Routing Queue Retry Span (Retry Span of Routing Queue)
- Network queue run limit (Max Run Limit of Network Queue)
- Network queue retry interval (Retry Interval of Network Queue)
- Network queue retry Span (Retry Span of Network Queue)
- Limit value on the number of requests input (Submit Number Limitation Value)
- Limit value on the number of users input (Submit User Number Limitation Value)
- Limit value on the number of groups input (Submit Group Number Limitation Value)
- Limit value which can be set by specifying group name or user name individually (The other Limitation Values are setting)
- License information (Use License)

  Number of licenses used and number of maximum licenses which can be used (License (xxx))

[Batch Server License Information (Specify option -B -L)]

- System name (Sysname)
- Product name (Product Name)
- Version (Ver)
- Expiration date (Expiration)
- Number of Licenses (Num)

[Scheduler Information (Specify option -D)]

- Scheduler ID (SCHID)
- Scheduler name (SCHName)
- Execution host (ExecutionHost)

- Number of bound execution queue (BindQueue)

[Scheduler Information (Specify option -D -f)]

- Scheduler ID (Scheduler ID)
- Scheduler name (Scheduler Name)
- Execution host (Execution Host)
- List of bound execution queue (Bind Queue list)
- Scheduler version (Scheduler Version)
- Scheduler Messages (Scheduler Message)

[Execution Host Information (Specify option -E)]

- Execution host (ExecutionHost)

  [B] is displayed at the head if the host is baremetal server.

  [C] is displayed at the head if the host is cloud instance.
- Batch server host name (BatchServer)
- Name of operating system (OS)
- OS release number (Release)
- Hardware name (Hardware)
- VE node number (VE)
- Load average of most recent 1 minute (Load)
- CPU average of most recent 1 minute (Cpu)

The following information is displayed with -t option.

- Job server number (JSVNO)
- Link status of the job server (JSV)
- Status of the execution host (STT)

ACT	Running
INA	Stopped

- (S) not use

[Execution Host Information (Specify option -E -f)]

- Execution host (Execution Host)

  [Baremetal] is displayed at the tail if the host is baremetal server.

  [Cloud] is displayed at the tail if the host is cloud instance.
- Batch serve host name (Batch Server)
- Name of operating system (Operating System)
- OS version number (Version)
- OS release number (Release)

- Hardware name (Hardware)
- OpenStack template name (OpenStack Template)
- Cloud bursting template name (Cloud Template)
- Defined resources for baremetal server (Defined Baremetal Resources)
  (Baremetal server only)
- VE node detail information (Vector Engine Information)　(VH only)

VE Node	VE node number
Cores	Core number
Memory	Memory size
Status	VE status
OS Status	VEOS status

- Execution host load information (Resource Information)

Memory	Information on physical memory size per execution host(unit:4KB)
Swap	Information on swap size per execution host(unit:4KB)
Number of Cpus	Information on the number of CPUs per execution host
Multi Instances GPU	Information on the MIG configuration per GPU (only Execution hosts with MIG)

- GPU Information (GPU Information) (only Execution host with GPU)
- Execution host Average Information (Average Information)

LOAD	Information on the load average per execution host
(Latest 1 minute)	Load average of most recent 1 minute
(Latest 5 minutes)	Load average of most recent 5 minutes
(Latest 15 minutes)	Load average of most recent 15 minutes
CPU	Information on the CPU activity ratio per execution host
(Latest 1 minute)	CPU average of most recent 1 minute
(Latest 5 minutes)	CPU average of most recent 5 minutes
(Latest 15 minutes)	CPU average of most recent 15 minutes

- CPUSET Information (CPUSET Information)
  (only socket scheduling feature enabled)

  Resource Sharing Group (Resource Sharing Group)

RSG Number	RSG number and CPUSET information
(none)	CPUSET is not supported on execution host.

- Socket resource usage (Socket Resource Usage)
  (only socket scheduling feature enabled)

  NUMA nodes (NUMA Nodes)

Socket	Socket number and socket resource information.
(none)	NUMA architecture is not supported on execution host.

- Device topology information (Device Topology)

(only HCA assignment feature and/or GPU-CPU Affinity feature enabled)

Socket	Information on the device connected to the socket.
PCIeSW	Information on the device connected to the PCIe switch.
(none)	Device resource is not defined on execution host.

The following information is displayed with -t option.

- Current state of the execution host (Current State)

Active	Running
Inactive	Stopped

- Time of the state transition (State Transition Time)
- Reason of the state transition (State Transition Reason)
- Job server number (Job Server Number)
- LINK status of the job server (LINK Batch Server)

UP	Job server is linked UP.
DOWN	Job server is linked DOWN.

- Node agent host (Node Agent)

[Node Group Information (Specify option -G)]

- Name of the node group (NodeGroup)
- Type of the node group (Type)
- Batch server host name (BatchServer)
- Comment for the node group (Comment)
- Number of Job servers (JSVs)
- Queue name that the node group is bound to (BindQueue)

[Node Group Information (Specify option -G -f)]

- Name of the node group (Node Group)
- Type of the node group (Type)
- Comment for the node group (Comment)
- List of the queues that the node group is bound to (Bind Queue list)
- List of the job servers included in the node group (Job Server number list)

The following is displayed only for the network topology node group:

- Switch layer level (Switch Layer)

The following is displayed only for the cloud bursting node group:

- Lock state (Lock State)
- Priority (Priority)
- Number of instances (Instances)

Display the number of cloud instances running in the node group in Live

Display the maximum number of cloud instances that can be launched in the node group in Max

- Network name (Network Name)
- Template (Template)

[Job Information (Specify option -J)]

- Job number (JNO)
- Request ID (RequestID)
- Execution Job ID (EJID)
- Memory size used by jobs (Memory)
- VE memory size used by jobs (VEMemory) (When -e is Specified)
- CPU time (Unit: second) (CPU) (When -c 0 is Specified)
- VE CPU time (Unit: second) (VECPU) (When -c 0 and -e are Specified)
- Accumulated CPU time (Unit: second) (ACCPU) (When -c 1 is Specified)
- Accumulated VE CPU time (Unit: second) (AVECPU) (When -c 1 and -e are Specified)
- Remaining CPU time (Unit: second) (RCPU) (When -m is Specified)
- Remaining VE CPU time (Unit: second) (RVECPU) (When -m and -e are Specified)
- Job server number (JSVNO)
- Execution host (ExecutionHost)

  * is displayed at the head if the job executed on VM.

  [B] is displayed at the head if the job executed on baremetal server.

  * is displayed at the head if the job executed on the container.

  [C] is displayed at the head if the job executed on the cloud instance.
- Job executing user (UserName)
- Job exit code used hexadecimal (Exit)

  0xf* : Exit code could not be collected due to JSV down

  others : Exit status returned by wait(2)

[Job Specific Resource Information (Specify option -J -t)]

- Request ID (Request ID)
- Job number (Batch Job Number)
- Per Logical Host Resource information (Resources Limits per Logical Host)

  Number of VE Node (Number of VE Nodes)

  Number of CPU (Number of CPU)

  Number of GPU (Number of GPU)

  CPU time (CPU Time)

  Memory size (Memory Size)

  Virtual memory size (Virtual Memory Size)

[Job Information (Specify option -J -f)]

- Request ID (Request ID)
- Job number (Batch Job Number)
- Execution Job ID (Execution Job ID)
- Job execution user (User Name)
- Job execution user ID (User ID)
- Job execution group ID (Group ID)
- Job server number (Job Server Number)
- Job server name (Job Server Name)
- Execution host (Execution Host)

  The host name of VM is displayed at the tail if the job executed on VM.

  [Baremetal] is displayed at the tail if the job executed on baremetal server.

  The host name of the container is displayed at the tail if the job executed on the container.

  [Cloud] is displayed at the tail if the job executed on cloud instance.
- Execution host (Vector Island) (When -e is Specified)
- Assigned VE node number (VE Node) (When -e is Specified)
- Job exit code used hexadecimal (Exit Code)

  0xf* : Exit code could not be collected due to JSV down

  others : Exit status returned by wait(2)
- Resource information (Resources Information)

  Memory size used by batch job (Memory)

  Memory cgroup size used by batch job (Memory Cgroup Resources)

  VE memory size used by batch job (VEMemory) (When -e is Specified)

  CPU time (Unit: second) (CPU Time)

  VE CPU time (Unit: second) (VECPU Time) (When -e is Specified)

  Accumulated CPU time (Unit: second) (Accumulated CPU Time)

  Accumulated VE CPU time (Unit: second) (Accumulated VECPU Time) (When -e is specified)

  Remaining CPU time (Unit: second) (Remaining CPU Time)

  Remaining VE CPU time (Unit: second) (Remaining VECPU Time) (When -e is specified)

  Virtual memory size used by batch job on Linux execution host (Virtual Memory)
- Assigned socket number　(Assigned Sockets)

[Job Specific Resource Information (Specify option -J -f -t)]

- Request ID (Request ID)

- Job number (Batch Job Number)
- Number of HCA (HCA Number)
- Reservation ID (Reservation ID)
- Per Logical Host Resource information (Resources Limits per Logical Host)

  Number of VE Node (Number of VE Node)

  Number of CPU (Number of CPU)

  Number of GPU (Number of GPU)

  CPU time (CPU Time)

  Memory size (Memory Size)

  Virtual memory size (Virtual Memory Size)

  VE CPU time (VE CPU Time)

  VE memory size (VE Memory Size)

  Stdout file size (Stdout Size) (only batch request)

  Stderr file size (Stderr Size) (only batch request)
- Per VE Node Resource information (Resources Limits per VE Node)

  VE CPU time (VE CPU Time)

  VE memory size (VE Memory Size)
- Per Process Resource information (Resources Limits per Process)

  CPU time (CPU Time)

  Number of Open File (Number Open File)

  Memory size (Memory Size)

  Virtual memory size (Virtual Memory Size)

  Data segment size (Data Segment Size)

  Stack segment size (Stack Segment Size)

  Core file size (Core File Size)

  Permanent file size (Permanent File Size)

  VE CPU time (VE CPU Time)

  VE memory size (VE Memory Size)

[Queue Information (Specify option -Q)]

- Queue Types

EXECUTION QUEUE	Batch queue
INTERACTIVE QUEUE	Interactive queue
ROUTING QUEUE	Routing queue
NETWORK QUEUE	Network queue

- Batch server host name (Batch Server Host)
- Queue name (QueueName)
- Scheduler ID (SCH)

- Number of job servers (JSVs)
- Route-to host of staging files (StagingMachine)
- Queue state (Submission factor) (ENA)

ENA	enabled
DIS	disabled

- Queue state (Starting factor) (STS)

ACT	active
INA	inactive

- Queue priority (PRI)
- Run limit for each queue (RLM)

  The run limit of the whole system is output to <TOTAL>
- Number of requests submitted in batch queue (TOT)
- Number of requests for each state

ARR	arriving
EXT	exiting
HLD	held
HOL	holding
MIG	migrating
POR	post-running
PRR	pre-running
QUE	queued
RUN	running
STG	staging
SUS	suspending, suspended, resuming
TRS	transiting
WAI	waiting
<TOTAL>	Total for Each State

[Queue Information (Specify by option -Q -f)]

- Queue type

  (Batch Queue, Interactive Queue, Routing Queue, Network Queue)
- Queue name
- Queue state (Submission and Starting factors) (Run State, Submit State)
- Scheduler ID (Scheduler ID)
- Scheduler Name (Scheduler Name)
- Bound job servers No. (Job Server Number)
- Bound node group name (Node Group Name)
- Access privilege check using supplementary group ID (Use Supplementary Groups ID

for Access Privileges Check)

ON	Supplementary group ID is used for access privileges check
OFF	Supplementary group ID is not used for access privileges check

- Access Limit Status

Access User List	Access limit status on users List of users enabled accessing will be displayed.
Noaccess User List	Access limit status on users List of users disabled accessing will be displayed.
Access Group List	Access limit status on groups List of groups enabled accessing will be displayed.
Noaccess Group List	Access limit status on groups List of groups that are disabled accessing will be displayed.

- Refuse submission (Refuse submission)

qsub ｜ submit	Refuse to submit via qsub or qlogin.
qmove	Refuse to submit via qmove.
local_routing	Refuse to routing via local routing queue.
remote_routing	Refuse to routing via remote routing queue.
(none)	Accept all via.

- Socket scheduling feature (NUMA Control)

ON	Use socket scheduling feature.
OFF	Do not use socket scheduling feature.

- Socket selecting policy (NUMA option)

Core bind policy (Core Bind Policy)

concentration	Socket concentration policy.
balance	Socket decentralization policy.

Memory allocation policy (Memory Allocation Policy)

membind	membind policy.
localalloc	localalloc policy.
Interleave	interleave policy.

- Submit CPU unit (Submit CPU Unit)
- Check the ratio of per_job CPU number and memory size (NUMA Unit Check)
- Hold restrictions (Hold Privilege) (except interactive queue)

User	Privilege as a general user or higher level is required to hold a request.
SpecialUser	Privilege as a special user or higher level is required to hold a request.
GroupManager	Privilege as a group manager or higher level is required to hold a

request.

Operator Privilege as an NQSV operator or higher level is required to hold a request.

Manager Privilege as an NQS manager or higher level is required to hold a request.

Scheduler Privilege as a scheduler or higher level is required to hold a request.

(none) No privilege is required to hold a request.

- Suspend restrictions (Suspend Privilege)

User Privilege as a general user or higher level is required to suspend a request.

SpecialUser Privilege as a special user or higher level is required to suspend a request.

GroupManager Privilege as a group manager or higher level is required to suspend a request.

Operator Privilege as an operator or higher level is required to suspend a request.

Manager Privilege as a manager or higher level is required to suspend a request.

Scheduler Privilege as a scheduler or higher level is required to suspend a request.

(none) No privilege is required to suspend a request

- Queue priority (Queue Priority)
- The number of requests which can be submitted to a queue (Submit Number Limit)
- The number of requests which one user can submit to a queue (Submit User Number Limit)
- The number of requests which one group can submit to a queue (Submit Group Number Limit)
- Maximum number of sub-requests in a parametric request (Subrequest Number Limit)
- Auto bind of Job server (Auto Bind JobServer)

ON Bind Automatically to the execution queue to which bound before on job server starting

OFF Not bind Automatically to the execution queue on job server starting

- Possibility to attach request (qattach command)
- Rerun availability of requests (Rerun default)

YES can rerun

NO cannot rerun

- Limit of rerun count of requests (Rerun Count Limit) (Only batch queue)

- Process manager setting of IntelMPI request (IntelMPI Process Manager)

  hydra Hydra process manager is used.

- Hook script function (Hook Function)
- Allowance for exclusive execution request (Exclusive Submit)
- Timeout of UserExit script (UserExit Timeout)
- Timeout of UserPP script (UserPP Timeout)
- Allowance for File Stage out (File Stageout)
- Delete a failed urgent request (Delete Failed Urgent Request)
- Use partial process swapping (Partial Process Swapping)
- GPU-CPU Affinity feature (GPU-CPU Affinity)

ON	Use GPU-CPU Affinity feature.
OFF	Do not use GPU-CPU Affinity feature.

- Number of CPU cores per GPU (GPU-CPU Affinity CPU per GPU)
- Process manager setting of NEC MPI request (NEC MPI Process Manager)

hydra	Hydra process manager is used.
mpd	MPD process manager is used.

- Minimum and Maximum value of job number per request. (Range of Jobs Limit per Batch/Interactive Request(min,max))
- Minimum and Maximum value of request priority. (Range of Request Priority) (except interactive queue)

  manager(min,max) Minimum and Maximum value of request priority which can be set in submitting a request as a manager.

  operator(min,max) Minimum and Maximum value of request priority which can be set in submitting a request as an operator.

  specialuser(min,max) Minimum and Maximum value of request priority which can be set in submitting a request as a special user.

  user(min,max) Minimum and Maximum value of request priority which can be set in submitting a request as a general user.

- Wait option for interactive request (Real Time Scheduling) (only for interactive queue)
- Idle timer for interactive request (Idle Timer) (only for interactive queue)
- Forced shell for interactive request (Restrict Shell) (only for interactive queue)
- Defined VE number (Defined VE Number)
- Possible range of the total VE nodes (Submit VE Node Range)
- Number of requests for each state

Total Request	Total number of requests submitted in queue
xxx Request	Number of requests in each state

- Queue list at request route-to *destination* (Destinations)
- Maximum number of running network requests for each request (Run Limit per Batch

Request)

- Host name at route-to *destination* of staging files (Staging Machine)
- Staging method for staging files (Staging Method)

  Internal   Standard staging method built in batch server

  External   User-defined external staging method
- Extended buffer size for the staging file transfer(Unit: KB) (Staging Extended Buffer Size)

  (none)   Using standard buffer for the staging file transfer
- User Exit Script (UserExit Script)

  Pre-running   At PRE-RUNNING status

  Post-running   At POST-RUNNING status
- Custom resource information (Custom Resources)

  Resource name

  Minimum and maximum limit for consumption (Range(min,max))

  Standard value (Std)

  Allow to use unused(0) or not (Permit Unused)

  Yes   Allow to use unused(0)   (default)

  No   Refuse to use unused(0)
- Range of the resource limit per logical host (Logical Host Resource Ranges)
- Range of the resource limit per VE node (VE Node Resource Ranges)
- Range of the HCA port number per logical host (Logical Host HCA Resource Ranges)
- Resource limit names and values (Resources Limits)

  Resource limit name and value for each request (Per_Req)

  Resource limit name and value for each job (Per_Job)

  Resource limit name and value for each process (Per_Prc)
- Kernel parameter name and value (Kernel Parameter)
- Limit value which can be set by specifying group name or user name individually (The other Limitation Values are setting)
- Scheduler Messages (Scheduler Message)

[Parametric Request Summary Information (Specify option -R)]

- Parametric request ID (RequestID)
- Request name (ReqName)
- Request owner (Username)
- Group(GrpName) (When --group is Specified)
- Queue name (Queue)
- Priority (Pri)
- State of the parametric request (STT)

ARI	arriving
EXT	exiting
HLD	held
QUE	queued
RUN	running
WAT	waiting

- Possibility of re-run (R)

Y	possible to re-run
N	impossible to re-run

- Possibility of Hold (H)

Y	possible to hold request
N	impossible to hold request

- Possibility of migration (M)

Y	possible to migrate job
N	impossible to migrate job

- Number of jobs per sub-request (Jobs)
- Total number of sub-requests (TOTAL)
- Number of existing sub-requests (ACTIVE)
- Number of terminated sub-requests (DONE)

[Parametric Request Information (Specify option -R -f)]

- Parametric Request identifier (Request ID)
- Request name (Request Name)
- Owner (User Name)
- Group (Group Name)
- Execution user ID (User ID)
- Group ID (Group ID)
- Current state of request (Current State)
- State of request one before (Previous State)
- Time changed to current state (State Transition Time)
- Reason for change to current state (State Transition Reason)
- Error message if state is changed due to error (Error Message)
- Name of the queue resided by request (Queue)
- Job topology (Job Topology)
- Process Manager (Process Manager) (if batch request and Job topology is NEC MPI)
- Request priority (Request Priority)
- Output level of request log (Request Loglevel)
- Re-execution possible/not possible (Rerunable)

- Holding possible/ not possible (Holdable)
- Client *privilege* during hold execution (Hold Type)
- Job Migrating possible/ not possible (Migratable)
- Client *privilege* at the suspended time (Suspend Type)
- Account code (Account Code)
- Name of standard output file (Stdout)
- Name of standard output error file (Stderr)
- Name of request log file (Reqlog)
- Full path name of shell used (Shell)
- E-mail address (Mail Address)
- Timing to send e-mail (Mail Option)
- Job Condition (Job Condition)
- Number of jobs per sub-request (Number of Jobs)
- Sub-request number information (Sub-request Numbers)
- Number of sub-requests (Number of Sub-requests)

  Total number of sub-requests (TOTAL)

  Number of existing sub-requests (ACTIVE)

  Number of terminated sub-requests (DONE)
- Date and time when request was created (Created Request Time)
- Date and time for start of scheduling (Entered Queue Time)
- Date and time that the request is planned to start (Planned Start Time)
- Date and time for execution of request (Execute Request Time)
- Date and time for start of request (Started Request Time)
- Date and time for end of request (Ended Request Time)
- Requested time to start the request execution (Requested Start Time)
- Requested time to end the request execution (Deadline Time)
- Umask value used octal (User Mask)
- Reservation ID (Reservation ID)
- Request group number which belongs for request connection(Request Group Number)
- Request ID of request submitted first for request connection (Lead Request)
- Possibility of qattach (qattach command)
- Cluster type selection. (Cluster Type Select)
- OpenStack template. (OpenStack Template) / Container template. (Container Template) / Cloud bursting template. (Cloud Template)

  CPU number used by the template (CPU Number)

  Memory size used by the template (Memory Size)

  GPU number used by the template (GPU Number)

  VE number used by the template (VE Number)

HCA port number used by the template (HCA Number)

- UserPP script information. (UserPP Script)
- Total number of VE nodes. (VE Node Number) (only when --venode is specified)
- Specified value of HCA. (HCA Number)
- Acceptance SIGTERM (Accept Sigterm)
- Cloud bursting availability status (Enable Cloud Bursting)
- User level check point script (Check Point Script) (only when given check point script)
- Script for the user level restart (Restart Script) (only when given check point script)
- VE NUMA mode (VE NUMA Mode)
- Sending signal at exceeding warning value of the resource limitation (Warning Signal)
- Limit name and limit value information of logical host (Logical Host Resources)

  VE node number (VE Node Number)

  CPU number (CPU Number)

  GPU number (GPU Number)

  CPU time (CPU Time)

  Memory size (Memory Size)

  Virtual memory size (Virtual Memory Size)

  VE CPU time (VE CPU Time)

  VE memory size (VE Memory Size)

  Stdout file size (Stdout Size) (only batch request)

  Stderr file size (Stderr Size) (only batch request)
- Limit name and limit value information of VE node(VE Node Resources)

  VE CPU time (VE CPU Time)

  VE memory size (VE Memory Size)
- Resource limit name and limit value information (Resources Limits)

  Resource limit name and limit value for each request (Per_Req)

  Resource limit name and limit value for each job (Per_Job)

  Resource limit name and limit value for each process (Per_Prc)
- Kernel parameter name and value (Kernel Parameter)
- User defined attributes (User Attributes)

[Job Server Information (Specify option -S)]

- Job server number (JSVNO)
- Job server name (JobServerName)
- Batch server host name (BatchServer)
- Execution host (ExecutionHost)

  [B] is displayed at the head if the host is baremetal server.

[C] is displayed at the head if the host is cloud instance.

- Link state (LINK)

UP	Batch server and job server are linked
DOWN	Batch server and job server are not linked

- Bind state (BIND)

Y	Scheduler and job server are bound
N	Scheduler and job server are not bound

(When -t option is specified, number of queues to which the job server is bound is displayed.)

- Queue name (Queue)
- Number of jobs (Jobs)
- Load average of most recent 1 minute (Load)
- CPU average of most recent 1 minute (Cpu)

[Job Server Information (Specify option -S -f)]

- Job server name (Job Server Name)
- Job server number (Job Server Number)
- Job server version (Job Server Version)
- Batch server host name (BatchServer)
- Execution host (Execution Host)

[Baremetal] is displayed at the tail if the host is baremetal server.

[Cloud] is displayed at the tail if the host is cloud instance.

- Link state (LINK Batch Server)

UP	Batch server and job server are linked
DOWN	Batch server and job server are not linked

- Bind state (BIND Queue)

BIND	Scheduler and job server are bound
UNBIND	Scheduler and job server are not bound

- Queue name (Queue)
- Enable/Disable of being used by JobManipulator (Assign JobManipulator license)

YES	Usable from JobManipulator
NO	Not usable from JobManipulator

- Network topology node group name (Network Topology Group)
- The name of cloud bursting node group to which the job server belongs (Cloud Group)
- OpenStack template name (OpenStack Template)
- Defined resources for baremetal server (Defined Baremetal Resources)

(Baremetal server only)

- Action for HCA failure check (HCA Failure Check)

OFF	Nothing to do for the node with failure
UNBIND	Unbind JSV on the node with failure
DOWN	Link down JSV on the node with failure

- Action for node health check (Health Check Action)

NOTHING	Nothing to do for the node with failure
UNBIND	Unbind JSV on the node with failure
DOWN	Link down JSV on the node with failure

- Execution host load information (Resource Information)

Memory	Information on physical memory size per execution host(unit:4KB)
Swap	Information on swap size per execution host(unit:4KB)
Number of Cpus	Information on the number of CPUs per execution host

- Execution host Average Information (Average Information)

LOAD	Information on the load average per execution host
(Latest 1 minute)	Load average of most recent 1 minute
(Latest 5 minutes)	Load average of most recent 5 minutes
(Latest 15 minutes)	Load average of most recent 15 minutes
CPU	Information on the CPU activity ratio per execution host
(Latest 1 minute)	CPU average of most recent 1 minute
(Latest 5 minutes)	CPU average of most recent 5 minutes
(Latest 15 minutes)	CPU average of most recent 15 minutes

- Information of Job Migration file transfer parameter (Migration_file Transfer Parameter Information)

Interface Hostname	Hostname of TCP/IP Network Interface on file routing
Socketbuffer	Size Socket buffer size
(OS default)	Default socket buffer size of OS
I/Obuffer Size	I/O buffer size

- Scheduler Messages (Scheduler Message)

[Network Request Information (Specify option -T)]

- Parent request identifier (RequestID)
- Owner (UserName)
- Network request priority (Pri)
- Staging direction (Dir)

IN	Stage-in
OUT	Stage-out

- Staging file number (STGNO)
- Staging file (StagingFile)

Dir = IN	Transmitting agency file name

Dir = OUT	Transmission place file name

- Staging method for staging files (Staging Method)

STAGE_IN	Transmitting agency file name
STAGE_OUT	Transmission place file name

- Network request status (STT)

QUE	queued
RUN	running
WAT	waiting

[Network Request Information (Specify option -T -f)]

- Parent request identifier (Request ID)
- Owner (User Name)
- Execution user ID (User ID)
- Group (Group ID)
- Network request priority (Network Request Priority)
- Staging direction (Staging Direction)

STAGE_IN	Stage-in
STAGE_OUT	Stage-out

- Staging file number (Staging Number)
- Staging file (Staging File)

STAGE_IN	Transmitting agency file name
STAGE_OUT	Transmission place file name

- Staging method (Staging Method)

Internal	NQSV internal staging
External	External staging defined by user

- Network request status (State)

[Custom resource Information (Specify option --custom)]

- Custom resource name (Custom Resource)
- Consumer (Consumer)
- Custom resource consume amount control information.

  Type (Type)

  Target (Target)

  Available resource limit (Available Resource Limit)

- Resource monitoring and control information

  Actual value collection and kind of the collected value (Check Mode)

  Job termination when the job exceed the custom resource limit (Terminate Job)

  The unit of the custom resource (Unit)

[Limit Information per Group and per User (Specify option --limit)]

- Batch server host name (BatchServer)
- Limit of request number to concurrently submit (Submit Number Limitation Value)
- Limit of user number to submit request(Submit User Number Limitation Value)
  Limit value by specifying user
- Limit of group number to submit request (Submit GroupNumber Limitation Value)
  Limit value by specifying group
- Queue type (Execution Queue, Interactive Queue, Routing Queue)
- Queue name
- Access Restriction State[Manager, Operator, Group Manager(only Group List)]

  Access User List — Access restriction state about user
  Display a list of users who are granted access privilege

  Noaccess User List — Access restriction state about user
  Display a list of users who are not granted access privilege

  Access Group List — Access restriction state about group
  Display a list of groups which are granted access privilege

  Noaccess Group List — Access restriction state about group
  Display a list of groups which are not granted access privilege

- Limit of request number to submit per queue (Submit Number Limit)
- Limit of request number for specified user to submit per queue (Submit User Number Limit)
  Limit value by specifying user
- Limit of request number for specified group to submit per queue (Submit Group Number Limit)
  Limit value by specifying group
- Maximum and minimum number of jobs per quest (Range of Jobs Limit per Batch/Interactive Request (min,max))
  Limit value by specifying user (User Limit)
  Limit value by specifying group (Group Limit)
- Resource limit name and limit information (Resource Limits)
  Elapse time limit per request ((Per-Req) Elapse Time Limit)
    Limit value by specifying user (User Limit)
    Limit value by specifying group (Group Limit)

[Template Information (Specify option --template)]

OpenStack Template:

- Template name (Template)
- Lock state (L)

-	unlocked
Y	locked

- OS image name (Image)
- Flavor name (Flavor)
- CPU Number (CPU)
- Memory size (Memory)
- GPU Number (GPU)
- Custom defined (Custom)
- Comment (Comment)

Container Template:

- Template name (Template)
- Lock status (L)

-	unlocked
Y	locked

- Image name (Image)
- CPU Number (CPU)
- Memory size (Memory)- GPU Number (GPU)
- Custom defined (Custom)
- Comment (Comment)

Cloud Template:

- Template name (Template)
- Lock status (L)

-	unlocked
Y	locked

- Image name (Image)
- CPU Number (CPU)
- Memory size (Memory)
- GPU Number (GPU)
- VE Number (VE)
- Custom defined (Custom)
- Comment (Comment)

[VE Template Information (Specify option --template --VE)]

Container Template:

- Template name (Template)

- Lock status (L)

-	unlocked
Y	locked

- Image name (Image)
- CPU Number (CPU)
- Memory size (Memory)
- VE Number (VE)
- HCA Port Number for MPI/IO/all (HCA)
- Custom defined (Custom)
- Comment (Comment)

[Template Information (Specify option --template -f)]

OpenStack Template:

- Template name (Template)
- Lock status (Lock State)
- OS image name (OS Image)
- Flavor name (Flavor)
- CPU Number (CPU Number)
- Memory size (Memory Size)
- GPU Number (GPU Number)
- Boot Timeout (Boot Timeout)
- Stop Timeout (Stop Timeout)
- Custom defined (Custom)
- Comment (Comment)
- Number of requests using template (Requests)

Container Template:

- Template name (Template)
- Lock status (Lock State)
- Image name (Image)
- CPU Number (CPU Number)
- Memory size (Memory Size)
- GPU Number (GPU Number)
- VE Number (VE Number)
- HCA port number (HCA Number)
- Boot Timeout (Boot Timeout)
- Stop Timeout (Stop Timeout)
- Custom defined (Custom)

- Comment (Comment)
- Number of requests using template (Requests)

Cloud Template:

- Template name (Template)
- Lock status (Lock State)
- Image name (Image)
- CPU Number (CPU Number)
- Memory size (Memory Size)
- GPU Number (GPU Number)
- VE Number (VE Number)
- Boot Timeout (Boot Timeout)
- Stop Timeout (Stop Timeout)
- Custom defined (Custom)
- Comment (Comment)

[VE node information(Specify option --venode)]

- Execution host name of VI (VectorIsland)
- VE node number (VE No)
- Number of core of VE (Cores)
- Memory size of VE (Memory)
- Status of VE (Status)
- Status of VEOS (OS Status)

[Cloud Template Information (Specify option --cloud_template)]

Cloud Template:

- Template name (Template)
- Lock status (L)

-	unlocked
Y	locked

- Image name (Image)
- CPU Number (CPU)
- Memory size (Memory)
- GPU Number (GPU)
- VE Number (VE)
- Custom defined (Custom)
- Comment (Comment)

[Cloud Template Information (Specify option --cloud_template -f)]

Cloud Template:

- Template name (Template)
- Lock status (Lock State)
- Image name (Image)
- CPU Number (CPU Number)
- Memory size (Memory Size)
- GPU Number (GPU Number)
- VE Number (VE Number)
- Boot Timeout (Boot Timeout)
- Stop Timeout (Stop Timeout)
- Custom defined (Custom)
- Comment (Comment)

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

NOTE

When the adjusting column width function (--adjust-column option) is used, it may take time to display a lot of information.

SEE ALSO

qsub(1)

NOTE

When the adjusting column width function (--adjust-column option) is used, it sometimes takes time before indicate a lot of information. (It takes about 5 seconds for 1000 lines output.)

1.16. qsub(1)

NAME

qsub -- Submit Batch Request

SYNOPSIS

qsub [*options*] [*script-file* ...]

DESCRIPTION

qsub submits batch requests to the NQSV.
Specify the name of a shell script file to be executed as a batch request in *script-file*. Failing to specify this script file will read a script from standard input.
The script will be spooled immediately as a file. The submit batch request will not be affected even if the script file is altered later.

Correctly submitting a batch request will display the request ID of the batch request. Request ID is created in the form of "seqno. host-name", "seqno" is a sequence number allocated to a request by the NQSV and "host-name" is the name of the batch server machine. The ID character string is used to uniquely identify a request.

You can specify two or more *script-file*. In this case, the Request connection function of NQSV associates the requests and controls an execution order.
Request connection has two connection types, serial and parallel.
There is no limit to the number of connected requests. But command line is limit to 4095 characters. Therefore connected requests are limited to the range 4095 characters that can be specified.

Serial type:

The next request starts to execute after finishing of preceding request execution. In qsub command, specify the script files delimited by space as *script1* [*script2* ...] .

Parallel type:

All connected requests start execution at the same time. In qsub command, specify the script file delimited by ":" as *script1*[:*script2*:...] .

By Request connection function, when all the connected requests are submitted, they will be in the HELD state of PRIV_MGR privilege, and won't be scheduling.

When all requests finished submitting, the 1st request is released.
Therefore, if the error(e.g. option analysis error) occurred while submitting the requests, the submitted requests remain in the HELD state. But there is a feature that batch server removes left requests automatically. please refer to section 5.1.5 in NQSV User's Guide [Management] for details.

Specify various options, which specify operations of Command qsub and characteristics of batch requests, in *options*. In the Request connection, the option specified for *options* is effective to all requests and the embedded option specified in the script file is effective only to the request.
The following options can be specified in *options*:

-a *date_time*

Set the batch request scheduling start time. The form for *date_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The each two digits in *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-A *account_string*

Set the account code of a batch request to *account_string*. The maximum number of characters which can be specified for *account_string* is 15 bytes. Command qsub sets a value specified in *account_string* as an account code attribute.

When the qsub command is executed with no -A option, the account code name which is described in the $HOME/.acct is used as an account code. The format of $HOME/.acct is as following.

- The character string from the first character to the first space or tab or return is recognized as an account code name.

- The length of account code name specified in $HOME/.acct should not exceed 15 bytes.

When this option is not specified and there is no $HOME/.acct, account code will not to be specified.
When account code is specified in either the option or $HOME/.acct, qsub does not check if it is an available account code.

-b *batch_job_count*

Specify the number of batch jobs to be executed. If this option is not specified, the number of batch jobs to be executed will be 1.

-B ["]*job_condition*[","*job_condition*"...]

Specify the conditions to select job server when the scheduler allocates job servers to batch job (Job condition).
Please refer to JobManipulator's manual for details of scheduling by this specification.
The form of *job_condition* is as follows.

[*job_number*:]*condition*

- *job_number* is the job number of the request (0 or more).

- When the *job_number* of *job_condition* is omitted or ALL is specified for *job_number*, job_condition is effective to all jobs.

- The following can be specified for *job_number*.

(1) Single number	0:
(2) All jobs specified by "ALL"	ALL:
(3) Two or more numbers using delimiters ","	0,2,5:
(4) Consecutive numbers using "-"	0-4:
(5) Combination of (3) and (4)	0,2,4-6:

- *job_number* is followed by condition sentence(*condition*) of scheduler ,separated by ":".

- The strings which do not apply to the specified conditions of *job_number* is treated as *condition*, and *job_number* is treated as ALL:.

- When -B option and -b option are specified at the same time, the number of jobs becomes a number specified by -b option.
 In the numbers of jobs specified by -b option, the job which is not specified with *job_number* doesn't have a condition.
 If *job_number* is larger than the number of jobs specified by -b option, the syntax error will occur.

- Available *condition* items of JobManipulator are as follows.

HW (Hardware)
Hardware name of the execution host on which the job is executed.

JSV (Job Server Number)
The number of the Job Server that executes the job.

NGRP(Name of Node Group)
The name of the Node Group that includes the execution host of the job.

A maximum of 255 bytes can be specified in *job_condition*.

-C *directive_prefix*

Specify an embedded option in the script file. In case option -C is omitted, the value of Environment Variable PBS_DPREFIX will be used. In the absence of definition of the environment variable, PBS will be used.

-d [*hostname*:]*path_name*

Directs the log output of a batch request to a file specified by *path_name*. A host can be specified by specifying hostname: as *path_name*. When the *hostname* is omitted, the name of the host which submits the batch request is used.
The following conversion-character can be specified as *path_name*.

%r	Converted to Request ID.
%s	Converted to the sequence number in Request ID.
%m	Converted to the Machine ID in Request ID.

%%   Converted to "%".

When not specified this option, the request log does not transmit to the client host.

-D *log_level*

Specify the log level of a batch request.

When not specified this option, the request log is not output.

The following can be specified in *log_level*.

0   The request log is not output. (default).

1   The logs about the batch request are output.

2   The logs about the batch request and the batch job are output.

-e [*hostname*:]*path_name*

Directs standard error output of a batch request to a file or a path specified in *path_name*.

A host can be specified by specifying *hostname*: as *path_name*.

When the *hostname* is omitted, the name of the host which submits the batch request is used.

The following conversion-character can be specified as *path_name*.

%f   Converted to a rank information of MPI program.

The format of the rank information is as follows.

applnum.universe.rank

applnum is an application number, *universe* is a universe number and *rank* is a rank number.

%f can be used only when -f option is specified with qsub command and NEC MPI program is executed in a necmpi request.

%m   Converted to the Machine ID in Request ID.

%r   Converted to Request ID.

%s   Converted to the sequence number in Request ID.

%t   Converted to sub-request number for parametric requests.

%[0*n*]j   Converted to Job number.

n indicates the digit number of the job number.

%j   : 0,1,2...

%03j   : 000,001,002...

%%   Converted to "%".

The form of file names in default for standard error output is as follows:

request_name.e%s

For a parametric request, the standard error files are created for each sub-request. The file names are followed by "."(period) + subrequest numbers.

-f

Enables to create output of necmpi type request for each MPI rank. By specifying this option, the output of the request cannot be created by job unit. -f is available only for NEC MPI request.

-G ["]*migration_file_path*[","*migration_file_path*"...]

User files that is routed to a destination host on doing Job migration can be specified by *migration_file_path* as an absolute path. If *migration_file_path* ends with '/', the specification is recognized as an directory and all of files under the directory are targets of routing. When "chkpnt_files" is specified to *migration_file_path*, all of checkpoint files are targets of routing.
The following conversion-character can be specified as *migration_file_path*.

%m	Converted to the Machine ID in Request ID.
%P	Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)
%r	Converted to Request ID.
%s	Converted to the sequence number in Request ID.
%t	Converted to Sub-request number of the parametric request.
%m	Converted to the Machine ID in Request ID.
%%	Converted to "%".

-h

Holds a batch request after submitting it.

-H y | n

Specify hold enable/disable. "y" will switch to a holdable request and "n", an unholdable request. If this option is not specified, requests will be holdable requests.

-I "*stage_in*"[,"*stage_in*"...]

Transmit the files which relate to the job execution from the client host to the execution host before the job execution.
The form of *stage_in* is as follows.

[*client_host*:]*file_path*, [*job_number*:]*executionhost_file_path*

client_host

Specify the client host name which has staging-file. If *client_host* is omitted, the submission host is used as *client_host*.

file_path

Specify the path name of a staging-file on the client host.
If file_path does not begin with "/", the file is treated as the relative path from the current working directory at the request submission.
The following conversion-character can be specified as *file_path*.

%H	Converted to the Hardware name of Execution Host.
%m	Converted to the Machine ID in Request ID.
%n	Numeric string specified. For details of the form, please refer to the following "Numeric string specification".
%P	Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)
%r	Converted to Request ID.
%R	Converted to the OS release number of Execution
%s	Converted to the sequence number in Request ID.
%S	Converted to the OS name of Execution Host.
%t	Converted to sub-request number for parametric requests.
%[0*n*]j	Converted to Job number. *n* indicates the digit number of the job number. %j : 0,1,2... %03j : 000,001,002...
%[0*n*]t	Converted to Sub-request number of a parametric request. n indicates the digit number of the Sub-request number.
%(*START-END*)date	If the request is a parametric request, this specification is

converted to the date string of *YYYYMMDD* format for each sub-request.
Please refer to the "DATE PARAMETER" described below.

%% Converted to "%".

job_number

Specify the job number to which staging-file relates.
The following can be specified for *job_number*.

(1) Single number	0:
(2) All jobs specified by "ALL".	ALL:
(3) Two or more numbers using delimiters ","	0,2,5:
(4) Consecutive numbers using "-"	0-4:
(5) Combination of (3) and (4)	0,2,4-6:
(6) Job number specified by %j	%j:
(7) Numeric string specified by %n	%{0,2,5}n:

If *job_number* is omitted or ALL is specified for *job_number*, files are staged to all jobs.
If *job_number* is larger than the number of jobs specified by -b option, the syntax error will occur.

executionhost_file_path

Specify the relative path name from environment variable STGDIR on the execution host to set staging-file. You can use ${STGDIR}/*executionhost_file_path* in the job when you use staging-file.
The following conversion-character can be specified as *executionhost_file_path*.

%m Converted to the Machine ID in Request ID.

%n Numeric string specified. For details of the form, please refer to the following "Numeric string specification".

%P Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)

%r Converted to Request ID.

%s Converted to the sequence number in Request ID.

%t Converted to sub-request number for parametric requests.

%[0*n*]j	Converted to Job number. *n* indicates the digit number of the job number. %j : 0,1,2... %03j : 000,001,002...
%%	Converted to "%".

You can specify the directory by adding "/" at the end of the path name.
When the directory is specified for *file_path*, all the regular files which exist Under the directory are targeted. However, if the file is a symbolic link, link-target file will be staging. And if the file is a hard link, the file will be unlinked.
When the directory is specified for *executionhost_file_path*, the staging-file is copied under the directory as a same file name as the original file.
When the directory is specified for *file_path*, the file cannot be specified for *executionhost_file_path*.

The SETUID/SETGID bit is not preserved on security.

-j *join_list*

Merges standard output and standard error output of batch requests. Merged results will be output to standard error or standard output specified by *join_list*.
The following can be specified for *join_list*.

e

Outputs merged results of standard output and standard error output of batch requests to standard error.

o

Outputs merged results of standard output and standard error output of batch requests to standard output.

-J y | n

Specify migration enable/disable. "y" will switch to a migratable request and "n", an unmigratable request.
If this option is not specified, requests will be migratable requests.

-l *resource_list*[,*resource_list*,...]

Set maximum and warning values for resource limits with requests, jobs and processes.
The resource_list are as follows. Enclose with double quotes (") when both

maximum and warning values of a limit are specified. Warning value is set to the same value as the maximum value if warning value is omitted.

Options for resource limits per request

elapstim_req=["]*max_limit*[,*warn_limit*"]

Limit by elapsed time after start of execution

Options for resource limits per job/logical host

cpunum_job=*max_limit*

Limit on the number of CPUs that can be used

In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

cputim_job=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

gpunum_job=*max_limit*

Limit on the number of GPUs that can be used

memsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VH.

socknum_job=*max_limit*

Limit of socket number

In case of SX-Aurora TSUBASA, it is the number of sockets used on VH.

This resource can be specified when the queue's numa control feature is enabled and submit_cpu_unit is set to "any" or "socket".

It cannot be specified with cpunum_job.

vmemsz_job=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

The above are legacy options for setting maximum and warning values for resource limits per job/logical host, and these options will not be expanded in the future.

When setting maximum and warning values for resource limits per job/logical host, it is recommended to use the "Options for resource limits per logical host" (ex.: --cpunum-lhost) described below.

Options for resource limits per process

coresz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum core file size that can be created

cputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

datasz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum data segment size that can be used

In case of SX-Aurora TSUBASA, it is the data size used on VH.

filenum_prc=*max_limit*

Limit on the number of files that can be opened simultaneously

filesz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum file size that can be created

stacksz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum stack size that can be used

In case of SX-Aurora TSUBASA, it is the stack size used on VH.

vecputim_prc=["]*max_limit*[,*warn_limit*"]

Limit by VE CPU time

In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

vememsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum VE memory size that can be used

In case of SX-Aurora TSUBASA, it is the memory size used on VE.

vmemsz_prc=["]*max_limit*[,*warn_limit*"]

Limit on maximum virtual memory size that can be used

In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

For a parametric request, the resource limits are set to each sub-request.

Limit values are sometimes converted into values that are convenient to the machine.

-m *mail_options*

Sends e-mail to report the batch request state changes.
The following can be specified in *mail_options*:

a

E-mail is sent when a batch request is abnormally terminated. For the Parametric request, e-mail is sent when each sub-request is abnormally terminated.

b

E-mail is sent when a batch request is started.

d

E-mail is sent when each sub-request (of the parametric request) is terminated.

e

E-mail is sent when a batch request is terminated. For the parametric request, e-mail is sent when all sub-requests are terminated.

n

E-mail is not sent. An error will result if specified simultaneously with other *mail_options*.

s

E-mail is sent when each sub-request (of the parametric request) is started.

-M *mail_list*

Specify the user list for sending e-mails. The form of the *mail_list* is as follows:

mail_address[,*mail_address*, ...]

The maximum length of the *mail_list* that can be specified is 1023 bytes.
If this option is not specified, *mail_list* is set up by the submitting user at the qsub host.

-N *name*

Specify a batch request name. A maximum of 63 bytes can be specified in *name*.
Failing to specify this option will take the name of the script file assigned to the command line. "STDIN" will be set if a script is not specified

-o [*hostname*:]*path_name*

Directs standard output of a batch request to a file or a path specified in *path_name*. A host can be specified by specifying *hostname*: as *path_name*.
When the *hostname* is omitted, it will be the name of the host which submits the batch request.
The following conversion-character can be specified as *path_name*.

%f Converted to a rank information of MPI program. The format of the rank information is as follows.

applnum.universe.rank

applnum is an application number, *universe* is a universe number and *rank* is a rank number.
%f can be used only when -f option is specified with qsub command and NEC MPI program is executed in a necmpi request.

%m Converted to the Machine ID in Request ID.

%P Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)

%r Converted to Request ID.

%P Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)

%s Converted to the sequence number in Request ID.

%t Converted to sub-request number for parametric requests.

%[0*n*]j Converted to Job number.

n indicates the digit number of the job number.

%j	: 0,1,2...
%03j	: 000,001,002...

%% Converted to "%".

The form of file names in default for standard output is as follows:

request_name.o%s

For a parametric request, the standard output files are created for each sub-request. The file names are followed by "."(period) + subrequest numbers.

-O "*stage_out*"[,"*stage_out*"...]

Transmit the files which is the output of the job from the execution host to the client host after the job execution.
The form of *stage_out* is as follows.

[*client_host*:]*file_path*,[*job_number*:]*executionhost_file_path*

client_host

Specify the client host name which has staging-file. If *client_host* is omitted, the submit host is used as a *client_host*.

file_path

Specify the path name to which the staging file is stored on the client host. If the path name does not begin with "/", it is treated as the relative path name from the current working directory at the request submission.
The following conversion-character can be specified as *file_path*.

%H	Converted to the Hardware name of Execution Host.
%P	Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)
%m	Converted to the Machine ID in Request ID.
%n	Numeric string specified. For details of the form, please refer to the following "Numeric string specification".
%r	Converted to Request ID.
%R	Converted to the OS release number of Execution Host.
%s	Converted to the sequence number in Request ID.

%S Converted to the OS name of Execution Host.

%t Converted to sub-request number for parametric requests.

%[0*n*]j Converted to Job number.
n indicates the digit number of the job number.

%j : 0,1,2...
%03j : 000,001,002...

%[0*n*]t Converted to Sub-request number of a parametric request.
n indicates the digit number of the Sub-request number.

%(*START-END*)date
If the request is a parametric request, this specification is converted to the date string of *YYYYMMDD* format for each sub-request.
Please refer to the "DATE PARAMETER" described below.

%% Converted to "%".

job_number

Specify the job number to which staging-file relates.
The following can be specified for *job_number*.

(1) Single number	0:
(2) All jobs specified by "ALL".	ALL:
(3) Two or more numbers using delimiters","	0,2,5:
(4) Consecutive numbers using "-"	0-4:
(5) Combination of (3) and (4)	0,2,4-6:
(6) Job number specified by %j	%j:
(7) Numeric string specified by %n	%{0,2,5}n:

If *job_number* is omitted or ALL is specified for job_number, files are staged from all jobs.
If *job_number* is larger than the number of jobs specified by -b option, the syntax error will occur.

executionhost_file_path

Specify the relative path name from environment variable STGDIR on the execution host to set staging-file.
You can use ${STGDIR}/*executionhost_file_path* in the job when you use staging-file.
The following conversion-character can be specified as *executionhost_file_path*.

%m	Converted to the Machine ID in Request ID.
%n	Numeric string specified. For details of the form, please refer to the following "Numeric string specification".
%P	Converted to Request ID. (Sub-request number is expanded with '_' instead of '[' and ']'.)
%r	Converted to Request ID.
%s	Converted to the sequence number in Request ID.
%t	Converted to sub-request number for parametric requests.
%[0*n*]j	Converted to Job number. *n* indicates the digit number of the job number. %j : 0,1,2... %03j : 000,001,002...
%%	Converted to "%".

You can specify the directory by adding "/" at the end of the path name.
When the directory is specified for *executionhost_file_path*, all the regular files which exist under the directory are targeted. However, if the file is symbolic link, link-target file will be staging. And if the file is a hard link, the file will be unlinked.
When the directory is specified for file_path, the staging-file is copied under the directory as the same file name as the source file.
When the directory is specified for *executionhost_file_path*, the file cannot be specified for *file_path*.
The SETUID/SETGID bit is not preserved on security.

-p *priority*

Specify batch request priority. *priority* must be an integer of [-1024...1023].

-P *privilege*

Specify access privilege for command execution.
The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager *privilege* is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-q *destination*

Specify a batch queue for queuing batch requests. In case this option is not specified, the default batch queue (specified by Environment Variable PBS_QUEUE) will be used.

-r y | n

Specify rerunning enable/disable. "y" will switch to a rerunable request and "n", a request that cannot be rerun. If this option is not specified, requests will be rerunable requests.

-R

Displays NQSV command version and NQSV/API version.

-s *request_start_time*

This is the dedicated option for using execution start time specifying feature of JobManipulator.
A batch request will be started to run at the time specified in *request_start_time*.
Please refer [JobManipulator] for details.
The form for *request_start_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The each two digits of *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.

-S *path_name*

Specifies the shell to execute the shell script for batch requests.
The shell given by "path_name" is executed as a login shell in a job.
Environment variables exported by /etc/profile, etc., which are loaded at login shell startup, are carried over to the job script.

When this option is specified, the contents in the job script is redirected to the specified shell. Therefore, the #! at the beginning of the job script may not be interpreted.
(It depends on the specification of the specified shell,
For example, if you specify /bin/bash with the -S option, #! is not interpreted.)
Please check the specifications of the shell you want to use.

-t *subreqno*

Submits a parametric request which spawns sub-requests with sub-request numbers specified by *subreqno*.
The format of *subreqno* is as follows.

N	Sub-request number=*N*
N-M	Sub-request numbers are the sequential numbers between *N* and *M*. (*N* must be smaller than *M*.)
N-M:S	Sub-request numbers are the numbers that the start index is *N*, the ending index is *M* and the stepping factor is *S*. (*N* must be smaller than *M*.)

-T *batch_job_topology*

Specify the mode of batch jobs to be executed. The following can be specified in *batch_job_topology*.

distrib

Batch jobs to be executed are distributed jobs. Batch jobs with number specified by option -b, will be created and executed.
Each batch jobs will execute the same script image.

intmpi

Batch jobs to be executed will be intmpi jobs.
This option has to be specified when executing IntelMPI programs in the batch job.

mvapich

Batch jobs to be executed will be mvapich jobs.
This option has to be specified when executing mvapich programs in the batch job.

necmpi

Batch jobs to be executed will be necmpi jobs. This option or necmpi_hydra have to be specified when executing NEC MPI programs in the batch request. In this case a process manager is used the one set in the submitted queue.

openmpi

Batch jobs to be executed will be openmpi jobs.
This option has to be specified when executing OpenMPI programs in the batch job.

pltmpi

Batch jobs to be executed will be pltmpi jobs.
This option has to be specified when executing Platform MPI programs in the batch job.

necmpi_hydra

Batch jobs to be executed will be necmpi jobs. This option or necmpi have to be specified when executing NEC MPI programs in the batch request. Unlike necmpi, a process manager always uses the hydra method.

If this option is not specified, the mode of batch jobs to be executed will be "distrib".

-U *user_attribute_list*

Specify the optional attribute string to the batch request.
The format of *user_attribute_list* is as follows.

user_attribute[=*value*] [,*user_attribute*[=*value*], ...]

The total character string length of *user_attribute* and *value* must be 4000 bytes or less.
If the existing name is specified for *user_attribute*, the value will be overwritten.
Multiple *value* must be enclosed in double quotes (").

user_attribute="value1,value2"

If *value* is omitted, value will be null character.
The string specified by this option has no influence on the behavior of the request.

-v *variable_list*

Specify the environment variables that are used in executing batch requests.
The form of *variable_list* is as follows:

variable[=*value*] [,*variable*[=*value*], ...]

The total length for character strings *variable* and *value* must be 4000 bytes or less.
If the existing variable name is specified for *variable*, the value will be overwritten.
Multiple *value* must be enclosed in double quotes (").

variable="value1,value2"

The following environment variables are set by default.
The value can be altered by this option except for PBS_ENVIRONMENT, PBS_JOBID, PBS_SUBREQNO, PBS_JOBNAME and PBS_NODEFILE.

PBS_ENVIRONMENT

Sets "PBS_BATCH" to indicate that the requests is batch processing.

PBS_JOBID

Sets a batch job identifier of the batch request.

PBS_SUBREQNO

Sets a sub-request number of the parametric request.

PBS_JOBNAME

Sets a batch request name.

PBS_NODEFILE

Sets a path to the file of execution host list of all jobs in the request.

PBS_O_HOME

Sets Environment Variable "HOME" in a client host.

PBS_O_HOST

Sets a client host name.

PBS_O_LANG

Sets Environment Variable "LANG" in a client host.

PBS_O_LOGNAME

Sets Environment Variable "LOGNAME" in a client host.

PBS_O_MAIL

Sets Environment Variable "MAIL" in a client host.

PBS_O_PATH

Sets Environment Variable "PATH" in a client host.

PBS_O_SHELL

Sets Environment Variable "SHELL" in a client host.

PBS_O_TZ

Sets Environment Variable "TZ" in a client host.

PBS_O_WORKDIR

Sets a directory for work in a client host.

-V

Exports all environment variables in submitting batch request to the environment for batch request execution.

-w *script_variable_list*

Specifies the variable to be used in an embedded option line in a script. This variable is not set as an environment variable at the batch request execution. The format of *script_variable_list* is as follows.

variable[=*value*] [,*variable*[=*value*], ...]

The total length for character strings *variable* and *value* must be 4000 bytes or less.
If the existing environment variable name is specified for *variable*, the value will be overwritten. (Please also refer EMBEDDED OPTIONS described below.)
Multiple *value* must be enclosed in double quotes (").

variable="value1,value2"

-y *reservation_id*

This is the dedicated option for using advance reservation feature of JobManipulator. A batch request will be executed using resource reserved section specified in *reservation_id*.
Please refer [JobManipulator] for details.
This option cannot be specified for parametric request.

-Y *deadline_time*

This is the dedicated option for using *deadline_time* feature of JobManipulator.
A batch request will be ended to run at the time specified in *deadline_time*.
Please refer [JobManipulator] for details.
The form for *deadline_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The each two digits of *CC*,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* are expressed as follows:

CC	First two digits of year
YY	Last two digits of year
MM	Month (01-12)
DD	Date (01-31)
hh	Hour (00-23)
mm	Minute (00-59)
SS	Second (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.
This option cannot be specified for parametric request.

-z

No message displayed when the batch request has been submitted correctly.
If this option is not specified, the request ID of the batch request and the queue
Name to which the request was submitted will be displayed in standard output.

-Z

Only the request ID is displayed when the batch request has been submitted correctly. Other message will not be displayed.

--accept-sigterm{=[*yes*|*no*]}

Specifies whether to enable or disable SIGTERM acceptance in scripts executed in the request. If this option is set to yes, SIGTERM is enabled for the script.
If the optional argument is omitted, it is assumed to be yes.
If this option is not specified, the default behavior is no (disabled).

--after *request_identifier* [,*request_identifier* ...]

Specifies the execution order of the submitted requests. The request that is submitted with this option will be scheduled to start execution after termination of the request specified by *request_identifier.*
By specifying two or more *request_identifiers* separated by comma, the submitted request waits all the requests.
As *request_identifier,* request ID of normal request, parametric request and its sub-request can be specified.
In an NQSV workflow script, request name can be used as *request_identifier.*
There is no limit to the number of connected requests. But command line is limit to 4095 characters. Therefore connected requests are limited to the range 4095 characters that can be specified.

--cancel-after

If a request terminated abnormally, following requests are canceled automatically.

--custom *cr_name*=*n*[,*cr_name*=*n...*]

Specify the consumption of custom resource specified by *cr_name.*
Multiple custom resources can be specified with delimiting by the comma.
Integer value of 1 to 2147483647 or unused(0) can be specified for *n.*
Submitting with unused(0) is allowed for only allowed queue.

--enable-cloud-bursting{=[*yes*|*no*]}

Specifies whether to burst and execute request to cloud computing resources. If this option is set to *yes*, bursting will be performed. If this option is set to *no*, bursting will not be performed. If this option is not specified, the default behavior is *no*.

--exclusive [= *host*]

Specify the request to execute as 1 logical host (job) per 1 host (exclusive execution).
Specify the parameter "*host*". This "*host*" parameter can omit.
This option can't be used with template (--template) option of provisioning function. It will be error when it used at same time.

--group=*group_name*

Specify the group name by which a request is executed. Specify group name to *group_name*.
When this option is not specified, a request is executed by the group on executing request submission command.
In case the Designated Group Execution Function of request is set to OFF, this option is to be ignored.

--job-separator

Works as a delimiter line between jobs. It can be abbreviated as "---".

--mig=*gi-name*[:*ci-slice-count*[,*ci-slice-count*...]]

Specify the multi instance GPU (MIG).
Specify GPU instance (GI) profile name to *gi-name*. qstat -Ef shows GI profiles to be able to specify.
Specify *ci-slice-count* if specified GI is further divided into compute instances (CI).
If no *ci-slice-count* is specified, a CI of the same size as the GI profile is created.
[Specification Example]
--mig=4g.20gb:1,1,2
In this case the GI profile as 4g.20gb is used and the GI is divided into 3 CIs.
If multiple --mig options are specified, multiple GIs can assign to the request. If both --mig and --gpunum-lhost are specified, only --mig is applied and --gpunum-lhost is ignored.

--parallel

The requests submitted with a --parallel option are scheduled to start simultaneously.
If two or more job scripts follows the other qsub options, the job scripts are submitted as separate requests and they are scheduled to start in parallel.
When this option is specified, the Request connection function (both serial type and

parallel type) cannot be available.

There is no limit to the number of connected requests. But command line is limit to 4095 characters. Therefore connected requests are limited to the range 4095 characters that can be specified.

--restart-script={no | job | *script*}

Specify the script for the user level restart.

If "no" is specified, the request finishes.

If "job" is specified, the request reruns.

Specify *script* as absolute path. If there is no file that is specified by *script* , qsub doesn't error.

If --restart-script option is specified without specifying --use-custom-suspend=chkpnt:*script*, qsub command fails.

Default is "job".

--template=*template_name*

Specify the template of job environment with provisioning.

cpunum_job, memsz_job and gpunum_job of -l option cannot be specified with this option.

The number of CPUs, memory size, and the number of GPUs defined in the template *template_name* are applied to the request.

--use-custom-suspend={no | chkpnt:*script*}

Specify the script for user level check point with chkpnt:*script* (the keyword chkpnt: cannot be omitted). If "no" is specified, no user level check point is taken. *script* is specified by absolute path. In this case, the child notation (~) indicating the home directory cannot be used. Even if there is no file indicating by *script*, qsub does not cause an error.

If this option is not specified, the default behavior is no.

--use-hca=*num*

--use-hca=[{io | mpi | all}:*num*][,{io | mpi | all}:*num...*]

It is possible to specify the port number of HCA according to the kind of HCA (for ScaTeFS direct IO, for MPI and both for ScaTeFS direct IO and for MPI).

For it,　you have to specify "io", "mpi" or "all" with colon before the "num".

If only "num" is specified, it is assumed that "all" is specified.

This option is effective in the request which use VE.

--user = *user_name*

Submit a request as *user_name* owner. Specify user name for *user_name*.

This option requires group manager privilege. The user name that specified to *user_name* must belongs to the group as primary group that is managed by this group manager privilege.

The user name that is mapped to another user name as local user or remote user in user mapping function cannot be specified to *user_name.*

If this option is not specified, the owner of the request is user who execute the qsub command.

--userpp-script=*location*:*path*[,*location*:*path*]

Execute the UserPP script specified by *path* at the *location.*

Following value can be specified for *location*.

prr	Execute location is PRE-RUNNING.
por	Execute location is POST-RUNNING.

--venode=*total_venum*

VE nodes which is specified by total_venum are assigned to a request. The number of logical host is calculated automatically using submitted queue's default number of VEs. You must specify a value 1 or greater. If "--venum-lhost" is specified the value which is specified by this option is used instead of the queue's default number of VEs. This option and the specification of the number of job (number of logical host) by -b option cannot be specified simultaneously.

--venuma={on | off}

Specifies the NUMA mode of the VE node on which the request is executed.

For more information about partitioning and NUMA modes, see " SX-Aurora TSUBASA VEOS NUMA Mode Guide for Partitioning Mode".

on	Set VE NUMA mode to ON.
off	Set VE NUMA mode to OFF.

--warning-signal=elapstim:*signal*

Specifies a sending signal at exceeding the warning value of the limit by elapsed time after start of execution. *signal* can be specified in the following form:

Signal name: (Example: SIGHUP)

Signal name with "SIG" removed: (Example: HUP)

Signal number: (Example: 1) For detailed information. refer to signal(7).

Default is SIGTERM if this option is omitted.

Options for resource limits per logical host

You can specify maximum or warning value for resource limits per logical host using following options. The logical hosts are generated according to designation of "-b job_count" at submitting of a request. Warning value is set to the same value as the maximum value if warning value is omitted.

You can also set the maximum and warning value for resource limits per logical host with -l option(ex.: -l cpunum_job), but since -l is a legacy setting and will not be expanded in the future, it is recommended to use the following options.

Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--cpunum-lhost=*max_limit*

Specify the limit on the number of CPUs per logical host. This option corresponds to -l cpunum_job=max_limit (Limit on the number of CPUs that can be used(Job). In case of SX-Aurora TSUBASA, it is the number of CPUs used on VH.

--cputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of CPU time per logical host. This option corresponds to -l cputim_job=max_limit Limit of CPU time (Job)). In case of SX-Aurora TSUBASA, it is the CPU time used on VH.

--gpunum-lhost=*max_limit*

Specify the limit on the number of GPUs that can be used per logical host. This option corresponds to -l gpunum_job=max_limit (Limit on the number of GPUs that can be used (Job)).

--memsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum memory size that can be used per logical host. This option corresponds to -l memsz_job=max_limit (Limit on maximum memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the memory size used on VH.

--stderrsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit of stderr file size per logical host.

--stdoutsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit of stdout file size per logical host.

--vecputim-lhost=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per logical host. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per logical host. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

--venum-lhost =*max_limit*

Specify the limit on number of VE node per job.

--vmemsz-lhost=*max_limit*[,*warn_limit*]

Specify the limit on maximum virtual memory size per logical host. This option corresponds to -l vmemsz_job=max_limit (Limit on maximum virtual memory size that can be used (Job)). In case of SX-Aurora TSUBASA, it is the virtual memory size used on VH.

Each VE process requires 32 TB and several hundred megabytes of virtual memory on the VH side.

Options for resource limits per venode

You can specify maximum or warning value for resource limits per VE node using following options. Warning value is set to the same value as the maximum value if warning value is omitted.

Please refer **FORM OF RESOURCE LIMIT VALUE** for detail.

--vecputim-venode=*max_limit*[,*warn_limit*]

Specify the limit of VE CPU time per VE node. In case of SX-Aurora TSUBASA, it is the CPU time used on VE.

--vememsz-venode=*max_limit*[,*warn_limit*]

Specify the limit on maximum VE memory size per VE node. In case of SX-Aurora TSUBASA, it is the memory size used on VE.

FORM OF RESOURCE LIMIT VALUE

Time Limit

Limits on time are specified in the following form:

[[*hours*:]*minutes*:]*seconds*[.*fraction*]

Specify the character string "unlimited" for unlimited.
Maximum limit of default is 2^31-1 seconds.

[Specification Example]

1234:58:21.29	1234 hours 58 minutes and 21.29 seconds
59:01	59 minutes and 1 second
12345	12345 seconds
121.1	121.1 seconds

Size Limit

Specify size limits in the following form:

integer[.*fraction*][*units*]

The following units can be specified in "units".

b	Byte
kb	Kilobyte (1024 b)
mb	Megabyte (1024 kb)
gb	Gigabyte (1024 mb)
tb	Terabyte (1024 gb)
pb	Petabyte (1024 tb)
eb	Exabyte (1024 pb)

Failing to specify in units will be construed as having specified byte. Specify the character string "unlimited" for unlimited. Maximum limit of default is 2^31-1 bytes.

[Specification Example]

1234	1234Bytes
1234kb	1234Kilobytes
1234.5gb	1234.5Gigabytes

Numeric string specification (%n)

A numeric string (%n) can be specified for a path name of the client host, job number and a path name of the execution host of a -I or -O option. The format is as follows.

%[0<n>]<l>n

<l> : {<w> [, <w> ...]}

<w> : <e> | (<e>) | (<e>)*<n> | (<e>)@<n>

<e> : <n> [, <n> ...] | <n>-<n> | <d>

<d> : N

<n> : Non-negative value

N : A list element is ignored.

- : Contiguous numeric string

Example:

0-2 : 0,1,2

2-0 : 2,1,0

1-1 : 1

1-z : Error

* : Entire numeric string is repeated.

Example:

(1)*3 : 1,1,1

(0-2)*3 : 0,1,2,0,1,2,0,1,2

(2-0)*3 : 2,1,0,2,1,0,2,1,0

(0-2)*z : Error

The repeat count must be 1 or more; otherwise, an error occurs.

@ : Elements of a numeric string are repeated.

Example:

(1)@3 : 1,1,1

(0-2)@3 : 0,0,0,1,1,1,2,2,2

(2-0)@3 : 2,2,2,1,1,1,0,0,0

(0-2)@z : Error

The repeat count must be 1 or more; otherwise, an error occurs.

Two or more numeric strings can be specified. If the number of elements in the specified numeric strings are equivalent to each other, the smallest number of elements is assigned to all the numeric strings.
The numeric string specification needs { } like %{(0-2)*2}n

DATE PARAMETER

By specifying the range of date in the form of %(*YYYYMMDD-YYYYMMDD*) date in path name of a staging file of client host with -I, -O option, a day during the specified consecutive days is corresponded to each sub-request of parametric request and the date is converted to date string in the form of *YYYYMMDD.*

The ways to correspond date to a sub-request are as follows.

(1) For consecutive days specified in the form of *YYYYMMDD-YYYYMMDD,* number starting from 0 is assigned to date of the consecutive days from the beginning in order.

(2) As for number assigned at (1) above, each date is corresponded to sub-request whose sub-request number of parametric request is same as assigned number of date.

(Example)

In case of specifying by qsub with
-t 2-4 -I "/input/data_%(20150101-20150131)date,infile", sub-requests to be executed are one with sub-request number, 2, 3 and 4. The staging file is corresponded to each sub-request as follows.

Staging File	Assigned Number	Sub-request number to use
/input/data_20150101	0	None
/input/data_20150102	1	None
/input/data_20150103	2	2
/input/data_20150104	3	3
/input/data_20150105	4	4
:	:	None
/input/data_20150131	30	None

Date parameter is only used on submitting parametric requests. Specified range (*START-END*) of date needs to meet the condition, START<=END.
The date specified as *START* and *END* must be after Jan. 1st of 1970.
In case of shell command line, parentheses (and) to specify range of date need to be escaped.

EMBEDDED OPTION

Options specified in command lines of qsub command can be described in a script file as embedded options. The line of embedded options in the job script file begins with "#PBS", and specify the option starting with a "-" character after "#PBS" with more than one space characters.
The string "PBS" to identify embedded options from other script line can be change by environment variable "PBS_DPREFIX" or by qsub -C option. (The -C option specified as an embedded option is ignored.)
If the same options is specified both in embedded options and in command line, the

options in the command lines is prior than embedded options.
The embedded option can be described before any line except a comment line, line started with ":" and blank line appears.

Environment variables and script variables specified by the -v or -w option can be referenced in an embedded option line as ${variable}.
If the same variable names exist, they are referenced in the following order.

1. Variable (specified by -w) to be used in an embedded option line in a script
2. Environment variable (specified by -v) specified by qsub command option
3. Environment variable

The following is examples of embedded options in scripts:

```
#!/bin/sh
#PBS -N Sample_JOB
#PBS -w ELAPS=600
#PBS -m be
#PBS -q batch1 #default
#PBS -l elapstim_req=${ELAPS}
```

ENVIRONMENT VARIABLES

PBS_QUEUE

Default batch queue name

PBS_DPREFIX

Embedded option in script file

STGDIR

This directory is the starting point when indicating the "staging-file" on the execution host.

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qdel(1), qstat(1)

NOTES

1. To specify a special character in a shell at a command line, escape.
2. To specify a character used as a delimiter of options (such as " in -I option) as a normal character, escape.
3. A slash (/), null character, and newline character cannot be specified in a file name.
4. In case there is no corresponding MPI execution environment on execution host, MPI request fails at PRE-RUNNING state.
5. When the batch request is executed, the job server sets the following environment variables. Therefore, the environment variables which specified by -V or -v option are invalid.
    - ENVIRONMENT
    - PATH
    - HOME
    - SHELL
    - USER
    - LOGNAME
    - QUEUENAME
    - STGDIR
    - PBS_JOBID
    - TZ (if the environment variable TZ is set in the job server)

1.17. qwait(1)

NAME

qwait -- Wait for the state transition of Request execution

SYNOPSIS

qwait [-P *privilege*] [-V] [-t *timeout*] [-w *wait_state*] [-f] *request_identifier*

DESCRIPTION

Wait until a request specified by *request_identifier* comes to a state specified by *wait_state*. If *wait_state* is omitted, qwait waits until the specified request comes to the EXITING state or jobs end. When activated, qwait sleeps and exits by displaying a message upon the specified status of request execution.

To *request_identifier*, request ID of a parametric request and sub-request ID can be specified. Termination codes and output of qwait for the request including sub-request of a parametric request are shown below.

0 exited+Exit code of request

Request has been transited to specified state and ended normally.

1 killed by signal+signo

Request has killed by the signal signo.

2 deleted in the XXXXXX state

Request has been deleted by delete request.

3 error

Request was deleted by the NQSV error.

4 time out

qwait terminated because of time out.

5 not found

The specified request does not exist.

6 qwait error

qwait aborted. This also includes an error of NQSV API such that qwait lost the connection to the batch server.

7 rerun

Request has been deleted by rerun request.

8 exited by system failure

Request has been exited by hardware failure or execution host down.

9 <*exceeded resource limit name*> + limit exceeded

Request has been terminated due to the resource limit being exceeded.

10 check point suspend normally

Request has been terminated normally by the user level check point script.

11 check point suspend abnormally

Request has been terminated abnormally by the user level check point script.

In the past, the exit code 1 was output when the resource limit (CPU usage time, etc.) for each logical host was exceeded or when the Elapse time limit was exceeded. This exit code is the same as when the request is terminated by sending a signal by a user or **qsig**(1), and it is not possible to determine whether the request is terminated due to exceeding the resource limit. Therefore, a new exit code 9 is provided, and this exit code is output when the resource limit is exceeded. The table below shows exceeded resource limit name in **qwait**(1) output.

The resource limit name	Exceeded resource limit name string in qwait output
The limit of CPU time per logical host	CPU time
The limit of maximum memory size per logical host	MEM size
The limit on maximum virtual memory size per logical host	VMEM size
The limit of VE CPU time per logical host	VE CPU time
The limit on maximum VE memory size per logical host	VE MEM size
The limit by elapsed time per logical host	ELAPSE time
The limit by Stdout file size per logical host	STDOUT size limit exceeded
The limit by Stderr file size per logical host	STDERR size limit exceeded

The limit by Stdout file size per logical host and The limit by Stderr file size per logical host	STDOUT&STDERR size limit exceeded

The following table shows a relation of termination codes and wait states of qwait.

Wait states	Return codes that qwait can return
exiting	0,1,2,3,4,5,6,7,8,9,10,11
exited	0,1,2,3,4,5,6,8,9,10,11

And the termination codes and output for parametric requests are as follows.

0 exited (*N* subrequests)

All sub-requests have been transited to specified state and ended normally (*N* is the number of sub-requests).

2 deleted in the XXXXXX state

Parametric request has been deleted by delete request (Deletion of each sub-request is not included).

3 error

Parametric request was deleted by the NQSV error.

4 time out

qwait terminated because of time out.

5 not found

The specified parametric request does not exist.

6 qwait error

qwait aborted. This also includes an error of NQSV API such that qwait lost the connection to the batch server.

7 rerun

Parametric request has been deleted by rerun request.

9 <*exceeded resource limit name*> + limit exceeded

Parametric request has been terminated due to the resource limit being exceeded. Note: <*exceeded resource limit name*> is the same as for a normal request.

The following options can be specified:

-f

Wait for state transition even when a request is not submitted yet.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager *privilege* is highest and User *privilege* is lowest. When -P option is not specified, access *privilege* for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-t *timeout*

Time specified in *timeout* will be set as time-out time in a unit of seconds.

-V

Display only the version of the command and NQSV API.

-w *wait_state*

Wait till a request comes to the state specified in *wait_state*. The following can be specified in *wait_state*.

exiting

Wait till a request comes to the EXITING state or jobs end.

exited

Wait till a request comes to the EXITED state.

When -w option is not specified, it means that *wait_state* is exiting.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

qsub(1)

1.18. qwait2(1)

NAME

qwait2 -- Wait for the end of multiple requests

SYNOPSIS

qwait2 [-P *privilege*] [-V] [-t *timeout*] *requests*

DESCRIPTION

qwait2 waits till a request specified by *requests* ends or is deleted. And it waits till the specified request comes to the EXITED state.
The request identifier or the request name is specified in *requests*. Only when executing qwait2 in a workflow script, the request name can be specified. You can specify two or more requests as follows in *requests*.

req,req,... Specify the request identifier or the request name delimited by comma. qwait2 waits till all requests ended. (AND)

req:req:... Specify the request identifier or the request name delimited by semicolon. qwait2 waits till one of requests ended.(OR)

When both of a comma and a semicolon are specified, qwait2 gives priority to a comma.

The request identifier of a parametric request and the sub-request identifier can be specified in *req* too. When the request identifier of a parametric request was specified, qwait2 waits till all sub-requests ended.
The request name of a parametric request can be specified in *req* too.

qwait2 displays the exit state of the specified request to stdout. The exit states are as follows.

n (decimal number)

Request has ended. It executed exit(2) or was killed by signal.
qwait2 displays an exit code of shell which was executing a job script of the request by a decimal number.
When a parametric request is specified, it displays an exit code of the first sub-request.

D

Request has been deleted by qdel(1). And a ELAPSE time limit was exceeded, so it was deleted.

F

Request has ended by H/W breakdown.

C

Request has been terminated normally by the user level check point script.

E

Request has been terminated abnormally by the user level check point script.

-

Request has not ended.

When waiting for more than one request, the exit states of the request are divided at space by a specified order.
Execute the following command image for one of an end of 1.bsv and 2.bsv or an end of 3.bsv to wait for the early one.

```
$ qwait2 1.bsv,2.bsv:3.bsv
256 - D
```

In above example, 3.bsv has ended, so qwait2 ended.
The meaning of the above result is as follows.

1.bsv has ended. The exit code is 256.
2.bsv has not ended.
3.bsv was deleted by qdel(1).

If qwait2 was executed in a workflow script, and a specified request had ended, qwait2 displays the exit state. In an above example, when qwait2 was executed in a workflow script even if 1.bsv has ended already, the same results as the above are displayed.
When executing qwait2 outside a workflow script or waiting for a request except the own workflow script in a workflow, if a specified request has ended it will be an error because a request doesn't exist.

When a batch server is restarted, qwait2 waits for an end of a request.

The following options can be specified:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege

g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV. If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-t *timeout*

Time specified in *timeout* will be set as time-out time in a unit of seconds. When specified time passes, qwait2 ends.
When -t option is not specified, time-out time is set by Environment Variable NQSV_QWAIT_TIMEOUT or NQSII_QWAIT_TIMEOUT. If NQSV_QWAIT_TIMEOUT or NQSII_QWAIT_TIMEOUT is not set, time-out time is 1 hour.
When time-out time has passed, qwait2 displays the exit state of the request.
Therefore when 0 was specified by *timeout*, it displays the current state of the request.

-V Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

NQSV_QWAIT_TIMEOUT or NQSII_QWAIT_TIMEOUT

Time-out time

SEE ALSO

qdel(1), qwait(1)

EXIT STATUS

The exit status of the qwait2 command is as follows.

0 qwait2 waited until a specified request ended or was deleted, and it has ended.

1 qwait2 has ended by an error. The request specified in *requests* doesn't exist or the option is wrong.

2 qwait2 has ended because of time-out.

3 qwait2 has ended by other error.

When qwait2 ended by an error, it doesn't display the exit state of the waiting request.

1.19. scacctjob/racctjob(1)

NAME

scacctjob,racctjob -- display the job accounting data

SYNOPSIS

/opt/nec/nqsv/bin/scacctjob [*options*]

/opt/nec/nqsv/bin/racctjob acct-server [*options*]

DESCRIPTION

scacctjob displays lists of the NQSV job accounting information.

racctjob accesses the accounting server and displays the job accounting information. The 1st argument must be the host name of the accounting server and the scacctjob's options can be specified after that.

The following are output by default.

JOB ID	Job ID
REQUEST ID	Request ID
REQUEST NAME	Request name
USER NAME	Submit user name
HOST NAME	Execution host name
QUEUE NAME	Submit queue name
QUEUED TIME	Submit time
START TIME	Start time
END TIME	End time
CPU(SECS)	CPU consumption time
REAL(SECS)	Real time of the job

The following can also be output by specifying the corresponding options.

GROUP NAME	Group name
ACCOUNT CODE	Account code
REQUEST PRTY	Request priority
NICE	Nice value
KCORE MIN	Total memory consumption (KB * MIN)
MEAN SIZE(K)	Average memory consumption

MAXMEM SIZE(K)	Max. memory consumption
V KCORE MIN	Total virtual memory consumption (KB * MIN)
V MEAN SIZE(K)	Average virtual memory consumption
V MAXMEM SIZE(K)	Max. virtual memory consumption
REQELAPS TIME(S)	Elapse Time Limit Value
REQCPU TIME(S)	CPU Time Limit Value for Job
REQCPU NUM	Number of CPU Limit Value for Job
REQMEM SIZE(K)	Memory Size Limit Value for Job
REQGPU NUM	Number of GPU Limit value for Job
EXIT STAT	Exit status
QUE TYPE	queue type
WAIT TIME	the time from scheduling start time until actual start time
REQVE NUM	Number of requested VE nodes for the Job (*1)
RSVVE NUM	Number of reserved VE nodes for the Job (*1)
VE CPU(SECS)	CPU consumption time on VE nodes (*1)
VE KCORE MIN	Total memory consumption on VE nodes (KB * MIN) (*1)
VE MEAN SIZE(K)	Average memory consumption on VE nodes (*1)
VE MAXMEM SIZE(K)	Max. memory consumption on VE nodes (*1)
VE REQ NODELIST	List of assigned VE nodes for the Job (*1)
VE USE NODELIST	List of used VE nodes for the Job (*1)
VE RSV NODELIST	List of reserved VE nodes for the Job (*1)
RESERVATION ID	Resource Reservation Section ID for Job
crname (*2)	Requested amount of custom resource. There is a valid value when consumption unit of the custom resource is a job.
crname TOTAL	Total usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment" or "integrate".
crname MEAN	Average usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment".
crname MAX	Maximum usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment".
UNIT	Unit of Custom Resource.
HW FAILURE	Hardware trouble flag.
VE TIME SLICE	Timeslice value on VE nodes (*3)
VE MFLOPS	MFLOPS value on VE nodes (*3)
VE MFLOPS(REAL)	MFLOPS(REAL) value on VE nodes (*3)
VE FPEC(K)	Floating point data element count on VE nodes (*3)
VE SYSTEM CALL	Number of system calls on VE nodes (*3)
VE-VH DATA TRANS	Data transfer amount between VE-VH (*3)

NUMBER OF VE PROCESS	Number of VE processes (*3)
VE INSTRCT(K)	Number of execution instructions on VE nodes (*3)
VEC-INST(K)	Number of vector instructions on VE nodes (*3)
VEC-ELMT(K)	Vector elements count on VE nodes (*3)
VEC-EXE(SECS)	Vector execution time on VE nodes (*3)
VEC-INT LOAD(K)	Vector load element count on VE nodes (*3)
VE L1 CACHE MISS TIME(SECS)	L1 cache miss time on VE nodes (*3)
VE CPU PORT CONFLICT(SECS)	Port conflict time on VE nodes (*3)
VEC ARITH EXEC TIME(SECS)	Vector arithmetic execution time on VE nodes (*3)
VEC LOAD EXEC TIME(SECS)	Vector load execution time on VE nodes (*3)
VE POWER THROTTLING(SECS)	Power throttling time on VE nodes (*3)
VE THERMAL THROTTLING(SECS)	Thermal throttling on VE nodes (*3)
VE MAX NTASK	Maximum number of threads on VE nodes (*3)
VEC-OPERAND RATIO	The ratio of vector operations to all operations on VE nodes (*3)
NUMA NODE	The number of NUMA node on VE nodes (*3)

*1 These items are available only for the environment whose execution host is SX-Aurora TSUBASA system.

*2 *crname* is the Custom Resource name.

*3 These items are only available if the host of execution is the SX-Aurora TSUBASA system and VE process account is enabled.

The following options can also be specified.

-N Shows the scacctjob version information. When this option is given, other options are disregarded.

-P *priv* Specify access privilege for the command execution.

One of the following characters can be specified as *priv*.

m Manager privilege
o Operator privilege
g Group manager privilege
s Special user privilege
u General User privilege

When -P option is not specified, access privilege for this command execution is set by environment variable NQSV_PRIV and NQSII_PRIV. If also those environment variable are not set, the command runs with the user privilege.

Those environment variable can take the following string values.

Note that those environment variable cannot be used for racctjob.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	General User privilege

-Y Displays the group name.

-X Displays the account code.

-t Displays CPU consumption time, divided into user mode and system mode.

-k Displays total memory consumption "KCORE MIN".

-m Displays average memory consumption (in kilobytes).

-M Displays maximum memory consumption (in kilobytes).

--vmem Displays total virtual memory consumption "KCORE MIN", average virtual memory consumption (in kilobytes), and maximum virtual memory consumption (in kilobytes).

-T Show the information of Elapse Time Limit Value (Second).

-q Show the information of CPU Time Limit Value for Job (Second), Number of CPU Limit Value for Job, and Memory Size Limit Value for Job (Kilobytes).

-G Show the information of Number of GPU Limit Value for Job

-p Displays the request priority, and nice value.

-h Shows the exit status.

-2 Adds the date to the time display.

-3 Displays the human readable format.

-K Show the type of queue.

-w Show WAIT TIME. There is wait time from submit time of the request to start time. Wait time is calculated from submit time if qmove transfers the request to another queue. Wait time is calculated from the time of qsub -a if qsub submits the request with -a.

-V Shows Number of requested VE nodes for the Job, Number of reserved VE nodes for the Job, CPU consumption time on VE nodes(Second), Total memory consumption on VE nodes (KB * MIN), Average memory consumption on VE nodes (KB) and Max. memory consumption on VE nodes (KB). This option is available only for the environment whose execution host is SX-Aurora TSUBASA system.

-Z Shows List of assigned VE nodes for the Job, List of used VE nodes for the Job and List of reserved VE nodes for the Job. This option is available only for the environment whose execution host is SX-Aurora TSUBASA system.

-F Shows requested amount of each custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -B option.

-B *crname1[,crname2...]*

Shows requested amount of the specified custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -F option.

-O Shows total usage, average usage, and maximum usage of each custom resources. "UNUSED" mean the custom resource was not used. This option cannot be specified with -j option.

-j *crname1[,crname2...]*

Shows total usage, average usage, and maximum usage of the specified custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -O option.

--veps

Shows CPU consumption time (user) (seconds), total memory consumption (kilobytes/min), and average memory consumption (kilobytes) on VE nodes. This option is not available at the same time as the -V option. The value displayed in this option is calculated based on the process account output by the VEOS.

--veps-perform

Shows the time slice, MFLOPS, and MFLOPS(REAL) values on VE nodes.

--veps-prog

Shows the number of floating-point data execution elements (K), the number of system calls, the amount of data transferred between VE-VH, and the number of execution processes on VE nodes.

--veps-inst

Shows the number of execution instructions (K), vector instruction execution (K), vector element (K), vector instruction execution time (seconds), and vector load execution element (K) on the VE node.

--veps-time

Shows the L1 cache miss time (seconds), CPU port contention time (seconds), vector arithmetic execution time (seconds), vector load execution time (seconds), HW stop time (seconds) due to power throttling, and HW stop time (seconds) due to thermal throttling on VE nodes.

--veps-task

Shows the maximum number of threads on VE nodes.

--veps-opratio

Shows the ratio of vector operations to all operations on VE nodes.

--veps-numanoode

Shows the number of NUMA node on VE nodes.

-R Outputs information for one job on multiple lines.

-v Does not display a heading line.

-r Shows resource reservation section ID of job. "-" is displayed when resource reservation section ID is not specified.

-u *user*

Shows information of a job belonging to the specified user. A user ID or login name can be specified to the argument *user*.

-g *group*

Shows information of a job executed by the specified group.

-x *acct-code*

Shows information of a job executed by the specified account code.

-I *request-ID*

Displays information for the job with the specified request ID.

-s *time*

Shows information of the jobs that have submitted at or after the specified time. Time specification format is as follows.

hour[:*min*[:*sec*[, *MM*/*DD*[/*YY*]]]]
(*MM*: month, *DD*: day, *YY*: last two digits of year)

-e *time*

Shows information of the jobs that have submitted at or before the specified time.

-S *time*

Shows information of the jobs that have started at or after the specified time.

-E *time*

Shows information of the jobs that have terminated at or before the specified time.

-A *time*

Displays information for jobs that have terminated on or after the specified time.

-b *time*

Displays information for jobs that have started at or before the specified time.

-n *request-name*

Displays information for jobs with the same request name as the specified one.

-Q *queue*

Displays information for jobs registered in the specified queue.

-W *type*

If the "master" is specified to type, only jobs with job number 0 are displayed. If "slave" is specified, jobs with job number 1 or higher are displayed.

-l *hostname*

Displays jobs executed by the specified host name.

--hw-failure

Displays hardware trouble flag.

Hardware trouble: 1

Failure was detected in the node health check at the end of the job: 20

Both of the above occurred: 21

--long-request-name

Displays all output information without cutting off even if exceeding the fixed length.

-o *sortname*

Sort by specified parameter. *sortname* can take the following string values.

JOB_ID	Sort by JOB ID
REQUEST_ID	Sort by REQUEST-ID
NAME	Sort by REQUEST-NAME
USER_NAME	Sort by USER NAME
GROUP_NAME	Sort by GROUP NAME
ACCOUNT_CODE	Sort by ACCOUNT CODE
HOST_NAME	Sort by HOST NAME
QUEUE_NAME	Sort by QUEUE NAME
QUEUED_TIME	Sort by QUEUED TIME
START_TIME	Sort by START TIME
END_TIME	Sort by END TIME
CPU	Sort by CPU (SECS)
CPU_SYS	Sort by CPU (SECS) SYS
CPU_USER	Sort by CPU (SECS) USER
REAL	Sort by REAL (SECS)
REQUEST_PRTY	Sort by REQUEST PRTY
NICE	Sort by NICE

REQELAPS_TIME	Sort by REQELAPS TIME(S)
REQCPU_TIME	Sort by REQCPU TIME(S)
REQCPU_NUM	Sort by REQCPU NUM
REQMEM_SIZE	Sort by REQMEM SIZE(K)
REQGPU_NUM	Sort by REQGPU_NUM
EXIT_STAT	Sort by EXIT STAT
KCORE_MIN	Sort by KCORE MIN
MEAN_SIZE	Sort by MEAN SIZE(K)
MAXMEM_SIZE	Sort by MAXMEM SIZE(K)
V_KCORE_MIN	Sort by V KCORE MIN
V_MEAN_SIZE	Sort by V MEAN SIZE(K)
V_MAXMEM_SIZE	Sort by V MAXMEM SIZE(K)
QUE_TYPE	Sort by QUE TYPE
WAIT_TIME	Sort by WAIT TIME(SEC)
REQVE_NUM	Sort by REQVE NUM (*1)
RSVVE_NUM	Sort by RSVVE NUM (*1)
VE_CPU	Sort by VE CPU(S) (*1)(*2)
VE_KCORE_MIN	Sort by VE KCORE MIN (*1)(*2)
VE_MEAN_SIZE	Sort by VE MEAN SIZE(K) (*1)(*2)
VE_MAXMEM_SIZE	Sort by VE MAXMEM SIZE(K) (*1)
RES_ID	Sort by RESERVATION_ID
HW_FAILURE	Sort by HW FAILURE
VE_CPU_PS	Sort by VE CPU(SECS) (*1)(*3)
VE_KCORE_MIN_PS	Sort by VE KCORE MIN (*1)(*3)
VE_MEAN_SIZE_PS	Sort by VE MEAN SIZE(K) (*1)(*3)
VE_TIME_SLICE	Sort by VE TIME SLICE (*1)
VE_MFLOPS	Sort by VE MFLOPS (*1)
VE_MFLOPS_REAL	Sort by VE MFLOPS(REAL) (*1)
VE_FPEC	Sort by VE FPEC(K) (*1)
VE_SYSTEM_CALL	Sort by VE SYSTEM CALL (*1)
VE_VH_DATA_TRANS	Sort by VE-VH DATA TRANS (*1)
VE_PROC_NUM	Sort by NUMBER OF VE PROCESS (*1)
VE_INSTRCT	Sort by VE INSTRCT(K) (*1)
VE_INST	Sort by VE-INST(K) (*1)
VEC_ELMT	Sort by VEC-ELMT(K) (*1)
VEC_EXE	Sort by VEC-EXE(SECS) (*1)
VE_INST_LOAD	Sort by VEC-INST LOAD(K) (*1)
VE_L1_CACHE_MISS_TIME	Sort by VE L1 CACHE MISS TIME(SECS) (*1)

VE_CPU_PORT_CONFLICT	Sort by VE CPU PORT CONFLICT(SECS) (*1)
VEC_ARITH_EXEC_TIME	Sort by VEC ARITH EXEC TIME(SECS) (*1)
VEC_LOAD_EXEC_TIME	Sort by VEC LOAD EXEC TIME(SECS) (*1)
VE_PORT_THROTTLING	Sort by VE POWER THROTTLING(SECS) (*1)
VE_THERMAL_THROTTLING	Sort by VE THERMAL THROTTLING(SECS) (*1)
VE_MAX_NTASK	Sort by VE MAX NTASK (*1)
VEC_OP_RATIO	Sort by VEC-OPERAND RATIO (*1)

(*1) These items are available only for the environment whose execution host is SX-Aurora TSUBASA system.
(*2) These items are sorted by the values of VE CPU (SECS), VE KCORE MIN, and VE MEAN SIZE(K) displayed with the -V option.
(*3) These items are sorted by the values of VE CPU (SECS), VE KCORE MIN, and VE MEAN SIZE(K) displayed with the --veps option.

-d *database[,database...]*

Specify the database file to output. This is a legacy option. Use for NQSV R1.02 and earlier databases. Use the --db-dir option for NQSV R1.04 or later.

--db-dir *database-dir[,database-dir...]*

Specify the directory to output the database. If this option and -d option is not specified, the default database (/var/opt/nec/nqsv/asv/jacct and jacct_cr) is assigned.

SEE ALSO

scacctreq(1-N), scacctrsv(1-N), racctreq(1-N), racctrsv(1-N).

1.20. scacctreq/racctreq(1)

NAME

scacctreq, racctreq -- display the request accounting data

SYNOPSIS

/opt/nec/nqsv/bin/scacctreq [*options*]
/opt/nec/nqsv/bin/racctreq acct-server [*options*]

DESCRIPTION

scacctreq displays lists of the NQSV request accounting information.

racctreq accesses the accounting server and displays the request accounting information. The 1st argument must be the host name of the accounting server and the scacctreq's options can be specified after that.

The following are output by default.

REQUEST-ID	Request ID
REQUEST-NAME	Request name
USER NAME	Submit user name
QUEUE NAME	Submit queue name
QUEUED TIME	Submit time
START TIME	Start time
END TIME	End time
CPU(SECS)	CPU consumption time
REAL TIME	real time of request
STATUS	Exit status (simple format)

The following can also be output by specifying the corresponding options.

PRERUN COUNT	Rerun count of the parent of parametric request (Only for sub-request of the parametric request)
RERUN COUNT	Rerun count
GROUP NAME	Group name
ACCOUNT CODE	Account code
REQUEST PRTY	Request priority
NICE	Nice value
KCORE MIN	Total memory consumption (KB * MIN)
MEAN SIZE(K)	Average memory consumption
MAXMEM SIZE(K)	Max. memory consumption
V KCORE MIN	Total virtual memory consumption (KB * MIN)
V MEAN SIZE(K)	Average virtual memory consumption
V MAXMEM SIZE(K)	Max. virtual memory consumption
REQELAPS TIME(S)	Elapse Time Limit Value
REQCPU TIME(S)	CPU Time Limit Value per Job. For hybrid requests, the average value of the CPU usage time limit for each job in the request
REQCPU NUM	Number of CPU Limit Value per Job. For hybrid requests, the average value of the number of CPUs that can be executed simultaneously for each job in the request

REQMEM SIZE(K)	Memory Size Limit Value per Job. For hybrid requests, the average memory size limit for each job in the request
REQGPU NUM	Number of GPU Limit Value per Job. For hybrid requests, the average value of number of GPU Limit for each job in the request
EXIT STAT	Exit status
QUE TYPE	queue type
JOBS	number of jobs
SUBREQ	number of subrequests
EHOST NUM	Number of execution hosts
TEMPLATE NAME	Template name (*2)
REQVE NUM	Number of requested VE nodes (*1)
RSVVE NUM	Number of reserved VE nodes (*1)
VE CPU(S)	CPU consumption time on VE nodes [SEC] (*1)
VE KCORE MIN(K)	Total memory consumption on VE nodes [KB * MIN] (*1)
VE MEAN SIZE(K)	Average memory consumption on VE nodes [KB] (*1)
VE MAXMEM SIZE(K)	Max. memory consumption on VE nodes [KB] (*1)
REQCPUTIME SUM(S)	Total CPU usage time limit for jobs in the request
REQCPUNUM SUM	Total number of CPUs that can execute concurrent jobs
REQMEMSIZE SUM(M)	Total memory size limit for jobs in the request
REQGPUNUM SUM	Total number of GPUs that can be executed concurrently for jobs in the request
crname (*3)	Requested amount of custom resource. Requested amount of all requests, when consumption unit of the custom resource is a request, Requested amount for each job, when consumption unit of the custom resource is a job.
crname TOTAL	Total usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment" or "integrate".
crname MEAN	Average usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment".
crname MAX	Maximum usage of the custom resource. There is a valid value when the Custom Resource check_mode is "moment".
UNIT	Unit of Custom Resource.
HW FAILURE	Hardware trouble flag.
VE TIME SLICE	Timeslice value on VE nodes (*4)
VE MFLOPS	MFLOPS value on VE nodes (*4)
VE MFLOPS(REAL)	MFLOPS(REAL) value on VE nodes (*4)
VE FPEC(K)	Floating point data element count on VE nodes (*4)
VE SYSTEM CALL	Number of system calls on VE nodes (*4)

VE-VH DATA TRANS	Data transfer amount between VE-VH (*4)
NUMBER OF VE PROCESS	Number of VE processes (*4)
VE INSTRCT(K)	Number of execution instructions on VE nodes (*4)
VEC-INST(K)	Number of vector instructions on VE nodes (*4)
VEC-ELMT(K)	Vector elements count on VE nodes (*4)
VEC-EXE(SECS)	Vector execution time on VE nodes (*4)
VEC-INT LOAD(K)	Vector load element count on VE nodes (*4)
VE L1 CACHE MISS TIME(SECS)	L1 cache miss time on VE nodes (*4)
VE CPU PORT CONFLICT(SECS)	Port conflict time on VE nodes (*4)
VEC ARITH EXEC TIME(SECS)	Vector arithmetic execution time on VE nodes (*4)
VEC LOAD EXEC TIME(SECS)	Vector load execution time on VE nodes (*4)
VE POWER THROTTLING(SECS)	Power throttling time on VE nodes (*4)
VE THERMAL THROTTLING(SECS)	Thermal throttling on VE nodes (*4)
VE MAX NTASK	Maximum number of threads on VE nodes (*4)
VEC-OPERAND RATIO	The ratio of vector operations to all operations on VE nodes (*4)

*1 These items are available only for the environment whose execution host is SX-Aurora TSUBASA system.
*2 These items are not available if the execution host is SX-Aurora TSUBASA system.
*3 *crname* is the Custom Resource name.
*4 These items are available only if the execution host is on the SX-Aurora TSUBASA system and VEOS process account feature is enabled.

The following options can also be specified.

-N Shows the scacctreq version information. When this option is specified, other options are disregarded.

-P *priv* Specify access privilege for the command execution.
One of the following characters can be specified as *priv*.
m Manager privilege
o Operator privilege
g Group manager privilege
s Special user privilege
u General User privilege

When -P option is not specified, access privilege for this command execution is set by environment variable NQSV_PRIV and NQSII_PRIV. If also those environment

variable are not set, the command runs with the user privilege.
Those environment variable can take the following string values.
Note that those environment variable cannot be used for racctreq.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	General User privilege

-r Displays the rerun count of the request and the rerun count of the parent request. The rerun count of each request or sub-request is displayed in RERUN COUNT. And the rerun count of the parametric request is appeared in PRERUN COUNT.

-Y Displays the group name.

-X Displays the account code.

-t Displays CPU consumption time, divided into user mode and system mode.

-k Displays total memory consumption "KCORE MIN".

-m Displays average memory consumption (in kilobytes).

-M Displays maximum memory consumption (in kilobytes).

--vmem Displays total virtual memory consumption "KCORE MIN", average virtual memory consumption (in kilobytes), and maximum virtual memory consumption (in kilobytes).

-T Shows the information of Elapse Time Limit Value (Second).

-q Shows the information of CPU Time Limit Value Per Job (Second), Number of CPU Limit Value Per Job, and Memory Size Limit Value Per Job (Kilobytes) , Number of GPU Limit Value Per Job. Shows the average value of each job for each resource request in the case of a hybrid request.

-z Shows the total information of CPU Time Limit Value Per Job (Second), Total

Number of CPU Limit Value Per Job, and Total Memory Size Limit Value Per Job (Kilobytes), Total Number of GPU Limit Value Per Job.

-p Displays the request priority, and nice value.

-h Shows the exit status in more detail.

-2 Adds the date to the time display.

-3 Displays the human readable format.

-K Shows the type of queue.

-C Shows the number of execution nodes, jobs and subrequests.

-L Shows the template name.

-V Shows Number of requested VE nodes, Number of reserved VE nodes, CPU consumption time on VE nodes (Second), Total memory consumption on VE nodes (KB * MIN), Average memory consumption on VE nodes (KB) and Max. memory consumption on VE nodes (KB). This option is available only for the environment whose execution host is SX-Aurora TSUBASA system.

-F Shows requested amount of each custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -G option.

-G *crname1[,crname2...]*

Shows requested amount of the specified custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -F option.

-O Shows total usage, average usage, and maximum usage of each custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -j option.

-j *crname1[,crname2...]*

Shows total usage, average usage, and maximum usage of the specified custom resource. "UNUSED" mean the custom resource was not used. This option cannot be specified with -O option.

--veps

Shows CPU consumption time (user) (seconds), total memory consumption (kilobytes/min), and average memory consumption (kilobytes) on VE nodes. This option is not available at the same time as the -V option. The value displayed in this option is calculated based on the process account output by the VEOS.

--veps-perform

Shows the time slice, MFLOPS, and MFLOPS(REAL) values on VE nodes.

--veps-prog

Shows the number of floating-point data execution elements (K), the number of system calls, the amount of data transferred between VE-VH, and the number of execution processes on VE nodes.

--veps-inst

Shows the number of execution instructions (K), vector instruction execution (K), vector element (K), vector instruction execution time (seconds), and vector load execution element (K) on the VE node.

--veps-time

Shows the L1 cache miss time (seconds), CPU port contention time (seconds), vector arithmetic execution time (seconds), vector load execution time (seconds), HW stop time (seconds) due to power throttling, and HW stop time (seconds) due to thermal throttling on VE nodes.

--veps-task

Shows the maximum number of threads on VE nodes.

--veps-opratio

Shows the ratio of vector operations to all operations on VE nodes.

-R Outputs information for one request on multiple lines.

-v Does not display a heading line.

-u *user*

Shows information of a request belonging to the specified user. A user ID or login name can be specified to the argument *user*.

-g *group*

Shows information of a request executed by the specified group.

-x *acct-code*

Shows information of a request executed by the specified account code.

-I *request-ID*

Displays information for the request with the specified request ID.

-s *time*

Shows information of the requests that have submitted at or after the specified time. Time specification format is as follows.

hour[:*min*[:*sec*[, *MM*/*DD*[/*YY*]]]]
(*MM*: month, *DD*: day, *YY*: last two digits of year)

-e *time*

Shows information of the requests that have submitted at or before the specified time.

-S *time*

Shows information of the requests that have started at or after the specified time.

-E *time*

Shows information of the requests that have terminated at or before the specified time.

-A *time*

Displays information for requests that have terminated on or after the specified time.

-b *time*

Displays information for requests that have started at or before the specified time.

-n *request-name*

Displays information for requests with the same request name as the specified one.

-Q *queue*

Displays information for requests registered in the specified queue.

-l *templatename*

Displays information for requests which have specified templatename.

-c *crname*

Displays information for requests that used the specified custom resource.

--hw-failure

Displays hardware trouble flag.

Hardware trouble: 1

Failure was detected in the node health check at the end of the job: 20

Both of the above occurred: 21

--long-request-name

Displays all output information without cutting off even if exceeding the fixed length.

-o *sortname*

Sort by specified parameter. When sorting by a custom resource, specify a "CR_" followed by a custom resource name as *sortname*. When sorting by a custom resource, -c option is ignored. Only the declaration values of custom resources can be sorted.

sortname can take the following string values.

REQUEST_ID	Sort by REQUEST ID
REQUEST_NAME	Sort by REQUEST NAME
USER_NAME	Sort by USER NAME
GROUP_NAME	Sort by GROUP NAME
ACCOUNT_CODE	Sort by ACCOUNT CODE
QUEUE_NAME	Sort by QUEUE NAME
QUEUED_TIME	Sort by QUEUED TIME
START_TIME	Sort by START TIME
END_TIME	Sort by END TIME
CPU	Sort by CPU (SECS)
CPU_SYS	Sort by CPU (SECS) SYS
CPU_USER	Sort by CPU (SECS) USER

REAL	Sort by REAL (SECS)
REQUEST_PRTY	Sort by REQUEST PRTY
NICE	Sort by NICE
REQELAPS_TIME	Sort by REQELAPS TIME(S)
REQCPU_TIME	Sort by REQCPU TIME(S)
REQCPU_NUM	Sort by REQCPU NUM
REQMEM_SIZE	Sort by REQMEM SIZE(K)
REQGPU_NUM	Sort by REQGPU NUM
EXIT_STAT	Sort by STATUS or EXIT STAT
KCORE_MIN	Sort by KCORE MIN
MEAN_SIZE	Sort by MEAN SIZE(K)
MAXMEM_SIZE	Sort by MAXMEM SIZE(K)
V_KCORE_MIN	Sort by V KCORE MIN
V_MEAN_SIZE	Sort by V MEAN SIZE(K)
V_MAXMEM_SIZE	Sort by V MAXMEM SIZE(K)
QUE_TYPE	Sort by QUE TYPE
EHOST_NUM	Sort by EHOST_NUM
NUM_JOBS	Sort by JOBS
NUM_SUBREQ	Sort by SUBREQ
RERUN_COUNT	Sort by RERUN COUNT
PRERUN_COUNT	Sort by PRERUN COUNT
TEMPLATE_NAME	Sort by TEMPLATE NAME
CR_*crname*	Sort by *crname* (custom resource)
REQVE_NUM	Sort by REQVE NUM (*1)
RSVVE_NUM	Sort by RSVVE NUM (*1)
VE_CPU	Sort by VE CPU(S) (*1)(*3)
VE_KCORE_MIN	Sort by VE KCORE MIN (*1)(*3)
VE_MEAN_SIZE	Sort by VE MEAN SIZE(K) (*1)(*3)
VE_MAXMEM_SIZE	Sort by VE MAXMEM SIZE(K) (*1)
REQCPUTIM_SUM	Sort by REQCPUTIME SUM (S)
REQCPUNUM_SUM	Sort by REQCPUNUM SUM
REQMEMSIZE_SUM	Sort by REQMEMSIZE SUM (M)
REQGPUNUM_SUM	Sort by REQGPUNUM SUM
HW_FAILURE	Sort by HW FAILURE
VE_CPU_PS	Sort by VE CPU(SECS) (*1)(*4)
VE_KCORE_MIN_PS	Sort by VE KCORE MIN (*1)(*4)
VE_MEAN_SIZE_PS	Sort by VE MEAN SIZE(K) (*1)(*4)
VE_TIME_SLICE	Sort by VE TIME SLICE (*1)

VE_MFLOPS	Sort by VE MFLOPS (*1)
VE_MFLOPS_REAL	Sort by VE MFLOPS(REAL) (*1)
VE_FPEC	Sort by VE FPEC(K) (*1)
VE_SYSTEM_CALL	Sort by VE SYSTEM CALL (*1)
VE_VH_DATA_TRANS	Sort by VE-VH DATA TRANS (*1)
VE_PROC_NUM	Sort by NUMBER OF VE PROCESS (*1)
VE_INSTRCT	Sort by VE INSTRCT(K) (*1)
VE_INST	Sort by VE-INST(K) (*1)
VEC_ELMT	Sort by VEC-ELMT(K) (*1)
VEC_EXE	Sort by VEC-EXE(SECS) (*1)
VE_INST_LOAD	Sort by VEC-INST LOAD(K) (*1)
VE_L1_CACHE_MISS_TIME	Sort by VE L1 CACHE MISS TIME(SECS) (*1)
VE_CPU_PORT_CONFLICT	Sort by VE CPU PORT CONFLICT(SECS) (*1)
VEC_ARITH_EXEC_TIME	Sort by VEC ARITH EXEC TIME(SECS) (*1)
VEC_LOAD_EXEC_TIME	Sort by VEC LOAD EXEC TIME(SECS) (*1)
VE_PORT_THROTTLING	Sort by VE POWER THROTTLING(SECS) (*1)
VE_THERMAL_THROTTLING	Sort by VE THERMAL THROTTLING(SECS) (*1)
VE_MAX_NTASK	Sort by VE MAX NTASK (*1)
VEC_OP_RATIO	Sort by VEC-OPERAND RATIO (*1)

*1 These items are available only for the environment whose execution host is SX-Aurora TSUBASA system.

*2 These items are not available if the execution host is SX-Aurora TSUBASA system.

*3 These items are sorted by the values of VE CPU (SECS), VE KCORE MIN, and VE MEAN SIZE(K) displayed with the -V option.

*4 These items are sorted by the values of VE CPU (SECS), VE KCORE MIN, and VE MEAN SIZE(K) displayed with the --veps option.

-d *database*

Specify the directory name of the target request accounting database. The database can be specified from older versions of NQSV. If this option is not specified, the default database (/var/opt/nec/nqsv/asv/racct, racct_tally, racct_cr, and racct_cr_tally) is used.

SEE ALSO

scacctjob(1-N), scacctrsv(1-N), racctjob(1-N), racctrsv(1-N).

1.21. scacctrsv/racctrsv(1)

NAME

scacctrsv/racctrsv -- display the reservation accounting data

SYNOPSIS

/opt/nec/nqsv/bin/scacctrsv [*options*]
/opt/nec/nqsv/bin/racctrsv acct-server [*options*]

DESCRIPTION

scacctrsv displays lists of the resource reservation accounting information.

racctrsv accesses the accounting server and displays the resource reservation accounting information. The 1st argument must be the host name of the accounting server and the scacctrsv's options can be specified after that.

The following are output by default.

RESERVATION ID	Resource reservation ID
RESERVATION TIME	
START	Planned start time
END	Planned end time
EHOST NUM	Number of hosts
CPU NUM	Number of CPUs per node
QUEUE	Specified queue
STATUS	Exit status

The following can also be output by specifying the corresponding options.

BATCH_SERVER	Batch server host name
USER NAME	User name
GROUP NAME	Group name
ACCOUNT CODE	Account code
CREATE TIME	Reservation creation time
ACTUAL RESERVED TIME	
START	Actual start time of the reservation
END	Actual end time of the reservation
TEMPLATE NAME	Template name

The following options can also be specified.

-N Shows the scacctrsv version information. This option is given, other options are disregarded.

-P *priv* Specify access privilege for the command execution.
One of the following characters can be specified as *priv*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	General User privilege

When -P option is not specified, access privilege for this command execution is set by environment variable NQSV_PRIV and NQSII_PRIV. If also those environment variable are not set, the command runs with the user privilege.
Those environment variable can take the following string values.
Note that those environment variable cannot be used for racctrsv.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	General User privilege

-U Displays the user name.

-Y Displays the group name.

-X Displays the account code.

-B Displays the batch server host name.

-C Displays the reservation creation time.

-F Displays the actual start time and the actual end time.

-L Displays the template name.

-R Outputs information for one reservation on multiple lines.

-v Does not display a heading line.

-h *bsv* Shows information of the reservation accounting belonging to the specified batch server.

-u *user*

Shows information of the reservation accounting belonging to the specified user. A user ID or login name can be specified to the argument *user*.

-g *group*

Shows information of the reservation accounting created for the specified group.

-x *acct-code*

Shows information of the reservation accounting belonging to the specified account code.

-I *RES-ID*

Displays information for the reservation accounting with the specified reservation ID.

-s *time*

Shows information of the reservation accounting that was created at or after the specified time. Time specification format is as follows.

hour[:*min*[:*sec*[, *MM*/*DD*[/*YY*]]]]
(*MM*: month, *DD*: day, *YY*: last two digits of year)

-e *time*

Shows information of the reservation accounting that was created at or before the specified time.

-S *time*

Shows information of the reservation accounting that was planned to start at or after the specified. The information is selected by the actual start time when -f option is specified.

-E *time*

Shows information of the reservation accounting that was planned to terminate at or before the specified. The information is selected by the actual termination time when -f option is specified.

-A *time*

Shows information of the reservation accounting that was planned to terminate at or after the specified. The information is selected by the actual termination time when -f option is specified.

-b *time*

Shows information of the reservation accounting that was planned to start at or before the specified. The information is selected by the actual start time when -f option is specified.

-Q *queue*

Shows information of the reservation accounting registered in the specified queue.

-l *templatename*

Shows information of the reservation accounting which have specified templatename.

-o *sortname*

Sort by specified parameter. *sortname* can take the following string values.

RES_ID	Sort by RESERVATION ID
BSV	Sort by BATCH_SERVER HOST
USER_NAME	Sort by USER NAME
GROUP_NAME	Sort by GROUP NAME
ACCOUNT_CODE	Sort by ACCOUNT CODE
CREATE_TIME	Sort by CREATE TIME
RES_START_TIME	Sort by RESERVATION TIME START
RES_END_TIME	Sort by RESERVATION TIME END
START_TIME	Sort by ACTUAL RESERVED TIME START
END_TIME	Sort by ACTUAL RESERVED TIME END
NODE_NUM	Sort by NODE NUM

CPU_NUM	Sort by CPU NUM
QUEUE	Sort by QUEUE
STATUS	Sort by STATUS
TEMPLATE_NAME	Sort by TEMPLATE NAME

-d *database*

Specify the directory name that the target data base exist. If this option is not specified, the default database (/var/opt/nec/nqsv/asv/rsvacct) is assigned.

SEE ALSO

scacctjob(1-N), scacctreq(1-N), racctjob(1-N), racctreq(1-N).

1.22. sstat(1)

This command is for administrators to refer to the scheduling-related setting information set by NQSV/JobManipulator. To check the detailed information of a request, please use qstat(1).

NAME

sstat -- Display JobManipulator Status

SYNOPSIS

sstat [-**R**] [-**P** *privilege*] [-**V**] [-**f**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**u** *userlist*] [-**q** *queuelist*] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**d**] [-**l**] [-**n**] [--**group**[=*group_name*]] [([-**q** *queuelist*] [*batch_request_identifier ...*]) | [-**p** *parametric_request_number*]]

sstat -**A** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**i** *interval*] [-**b** *begin_time*] [-**e** *end_time*] [-**n**]

sstat -**B** [-**P** *privilege*] [-**V**] [-**f**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**d**] [-**l**] [-**n**] [--**group**[=*group_name*]] [--template]

sstat -**C** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**l**] [-**n**] [*complex_name ...*]

sstat -**D** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**f**] [-**n**]

sstat -**E** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**a**] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**l**] [-**n**] [-**g** *node_group*] [*execution_host ...*]

sstat -**E** --**eco-status** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**f**] [-**l**] [-**n**] [-**g** *node_group*] [*execution_host ...*]

sstat -**E** --**hw-failure** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**a**] [-**l**] [-**n**]

sstat -**E** -f [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**a**] [-**l**] [-**n**]

sstat -**J** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*] [-**l**] [-**n**] [--**dynamic**] [*jobserver_identifier ...*]

sstat -**M** [-**P** *privilege*] [-**V**] [-**h** *server_host*] [-**s** *scheduler_id*]

sstat -**Q** [-**P** *privilege*] [-**V**] [-**f** [-**j**] [-**a**] [--**limit**]] [-**h** *server_host*] [-**s** *scheduler_id*] [-**e**] [-**g**] [-**i**] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**d**] [-**n**] [*queue ...*]

sstat -**S** [-**P** *privilege*] [-**V**] [-**f** [--**limit**]] [-**h** *server_host*] [-**s** *scheduler_id*] [-**F** *Item list*] [-**o** *Item list*] [-**O** *Item list*] [-**d**] [-**l**] [-**n**]

sstat --help

sstat --version

DESCRIPTION

1. If the option **-R** is specified or if none of the options **-A**, **-B**, **-C**, **-D, -E**, **-G**, **-J**, **-S**, **-M**, and **-Q** is specified:

   Scheduling information of a batch request will be displayed.
   Specify a request identifier in *batch_request_identifier* in the following form:

   request_number[.*batch_server*]

   request_number

   Sequence number of the request ID. For a subrequest, specify [*n*] following the sequence number. Specify the subrequest number in *n*.

   batch_server

   The name of batch server

   If *request_number* is omitted, all batch requests owned by a command executer will be the objects.
   If *batch_server* is omitted, objects will be NQSV batch servers used by JobManipulator.

   If *batch_request_identifier* is omitted, all batch requests owned by a command executer will be the objects.

   The following options can be specified:

   -f

   Displays detailed information.

   -n

   Do not output headers.

-p *parametric_request_number*

Targets will be the subrequests of the parametric request specified by *parametric_request_number*, which have been in JobManipulator.

-q *queuelist*

Targets will be the batch request which belongs to the execution queue specified by *queuelist*.

It is possible to specify two or more queue names in *queuelist* if delimited by commas.**-n**

Do not output headers.

-u *userlist*

Targets will be the batch request which is owned by the user specified by *userlist*. It is possible to specify two or more login names in *userlist* if delimited by commas.

--group[=*group_name*]

Display group name. If the group name is specified, only the requests of the group are displayed.

2. If the option **-A** is specified:
   Assign map information of a batch request is displayed.

   The following options can be specified:

-b *begin_time*

Specifies the beginning time of assign map displayed to *begin_time*. The form for *begin_time* is as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The two digits are expressed as follows:

CC: First two digits of year

YY: Last two digits of year

MM: Month (01-12)

DD: Date (01-31)

hh: Hours (00-23)

mm: Minutes (00-59)

SS: Seconds (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.
If this option is omitted, the current time will be used.
The beginning time is rounded down to the nearest minute.

-e *end_time*

Specify the end time of assign map displayed to *end_time*. The form for *end_time* is the same as *begin_time*.
If this option is omitted, it will be the latest time of the planned end time of requests which are assigned on the map.

-i *interval*

Specify the interval for displaying assign map information to *interval*. The unit is a minute.
If this option is omitted, assign map information is displayed in every 1-minute interval.

-n

Do not output headers.

3. If the option **-B** is specified:
Information about the Resource Reservation Section is displayed:

The following option can be specified.

-f

Display detailed information.

-n

Do not output headers.

--**group**[=*group_name*]

Display the group name. If the group name is specified, only the Resource Reservation Section of the group is displayed.

--template

Only information of the Resource Reservation Section specifying template are displayed.

4. If the option **-C** is specified:
   Display the information of complex queue controlled by JobManipulator.

   For *complex_name*, specify the name of the complex queue whose information is to be displayed. If *complex_name* is omitted, information of all complex queues is displayed.

   The following option can be specified.

   -n

   Do not output headers.

5. If the option **-D** is specified:
   Information about the power-saving schedule is displayed:

   The following option can be specified.

   -f

   Display detailed information.

   -n

   Do not output headers.

6. If the option **-E** is specified:
   Information about the execution hosts used by JobManipulator is displayed.
   For *execution_host*, specify the name of the execution host whose information is to be displayed. If *execution_host* is omitted, information to be displayed differs depending on whether the **-a** option is specified or not. For details, see the following description.

   The following option can be specified:

-a

Display information of the attached execution hosts. If **-a** is omitted, information of the execution hosts that were bound with the queue is displayed.

-g *node_group*

For node_group, specify a node group name. Information of the execution hosts in the specified node group is displayed.

-n

Do not output headers.

--eco-status

Display the eco-state of the execution hosts. If **-f** is also specified, the detailed information is displayed.

--hw-failure

Display the execution host with a HW failure and the execution hosts excluded from scheduling due to the HW failure.

7. If the option **-J** is specified:
   Display the job server information for JobManipulator.
   The number of target job server is specified with *jobserver_identifier*.
   When *jobserver_identifier* is omitted, all information on job servers from 0 to 10239 are displayed.

   The following option can be specified.

   -n

   Do not output headers.

8. If the option **-M** is specified:
   Information about the scheduling priority is displayed.

9. If the option **-Q** is specified:
   Information about the execution queues managed by JobManipulator is displayed.
   General user can display only queue with access privilege of the user

   The following option can be specified:

   -e

   Display the information of execution queues.

   -f

   Display the detailed information.

   If **-j** is specified, the job server assign priority for the queue is displayed. If **-a** is also specified the job server assign priorities of all attached hosts are displayed. If **-a** is omitted, the job server assign priorities of only the bound job servers are displayed.

   If **--limit** is specified extra, the request run limit, cpu run limit and VE run limit for each user/group in the queue is displayed.

   -i

   Display the information of interactive queues.

   -n

   Do not output headers.

   queue

   Specify the name of the queue to be displayed.

10. If the option **-S** is specified:
    Information of the JobManipulator server is displayed.

    The following options can be specified:

    -f

    Displays detailed information.

    If **--limit** is specified extra, the request run limit, cpu run limit and VE run limits for each user/group in the scheduler is displayed.

    -n

    Do not output headers.

11. If the option **--help** is specified:

    Display only the usage of sstat.

12. If the option **--version** is specified:
    Display only the version of the command and JobManipulator API.

The following options can be specified in common:

-d

Changes display format of time data to **d+hh:mm:ss**.

-F *item*[,*item*,...]

Customizes output of this command according to the items for each option (R, Q, S, E, B, C, no option) which indicates information to be displayed.

[Item list]

Item list:

[-R or No display option]: rid reqnm own quenm stt pri priasg priesc sdate edate elapse cpus mem

[-Q]: quenm type qrlm qurlm qualm qtot exc que asg run ext hld sud

[-S]: jmhost grlm gurlm gualm tot exc que asg run ext hld sud sstt

[-E]: ehost cpuratio memratio

[-B]: resid resstime resetime reshosts rescpus quenm resnm

[-C]: cquenm crlm curlm cualm type qtot exc que asg run ext hld sud

-h *server_host*

Specify the JobManipulator server host. If this option is omitted, the default JobManipulator server host will be set. (Host name specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf) If a default host is not specified, a local host will be assumed.

-l

Displays all output information without cutting off even if exceeding the fixed length.

-o *item*[,*item*,...]

Information will be sorted in ascending order with the key specified in *item*.
If this option is specified more than once, the sorted results are further sorted according to the order of items specified.
It is also possible to use in combination with the **-O** option (descending sort).

-O *item*[,*item*,...]

Information will be sorted in descending order with the key specified in *item*.
If this option is specified more than once, the sorted results are further sorted according to the order of items specified.
It is also possible to use in combination with the **-o** option (ascending sort).

-P privilege

Specify access privilege for command execution. The following can be specified in *privilege*.

m	(Manager privilege)
o	(Operator privilege)
g	(Group Manager privilege)
s	(Special user privilege)
u	(User privilege)

Among access privileges, Manager privilege is the highest and User privilege is the lowest.

The information on other user's requests can be referred to by using Special user privilege further to user privilege. The user with group manager privilege can refer to the information on the requests of his/her managed group.

When the **-P** option is not specified, access privilege for command execution is set by Environment Variable JM_PRIV. If this variable is not set, user privilege is set up.

-s *scheduler_id*

Specify the scheduler ID of JobManipulator. If this option is omitted, the scheduler ID will be the value of Environment Variable JM_SCHED_ID. If this variable is not set, the scheduler ID specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf is used. If none of the above is not specified, 0 will be the scheduler ID.

-V

Display only the version of the command and JobManipulator API.

Standard Output

The following items are displayed in each list:

Batch Request Scheduling Information

- Batch request identifier (**RequestID**)
- Batch request name (**ReqName**)
- Owner (**UserName**)
- Group name (**GrpName**)(When **--group** is specified.)
- Name of batch queue resided by batch request (**Queue**)
- Priority (**Pri**)

  Assign / Escalation

- Batch request status (**STT**)

ASG	ASSIGNED
CHK	CHKPNTING
EXC	EXCLUDE
EXD	EXITED
EXT	EXITING
HLD	HELD
POR	POST-RUNNING
PRR	PRE-RUNNING
QUE	QUEUED
RES	RESUMING
RUN	RUNNING
STD	STAGED

STG	STAGING
SUS	SUSPENDING
SUD	SUSPENDED
WAT	WAITING

QUE in JobManipulator means the state before staging.

- Scheduled start time (**PlannedStartTime**)

Batch Request Scheduling Information (with the **-f** option specified)

- Batch request identifier (**Request ID**)
- Batch request name (**Request Name**)
- Owner (**User Name**)
- Execution user ID (**User ID**)
- Group name (**Group Name**)
- Group (**Group ID**)
- Current state of batch request (**Current State**)
- Previous state of batch request (**Previous State**)
- Time changed to current state (**State Transition Time**)
- Reason for change to current state (**State Transition Reason**)
- Error message if state is changed due to error (**Error Message**)
- Reason for being suspended (**Suspend Reason**)

INTERRUPT This request is suspended due to interruption by an urgent or special request.

SMGR_SUSPEND This request is suspended by the **smgr** command.

SMGR_RESUME Resumption request is sent to the suspended request by the **smgr** command.

(none) This request is not in suspended state.

- Name of batch queue resided by batch request (**Queue**)
- Scheduling priority at assign(**Scheduling Priority (Assign)**)

Value of each elements

User share	**(User Share)**
User rank	**(User Rank)**
Request priority	**(Request Priority)**
Elapse time	**(Elapse Time)**

The number of CPU	**(CPU Number)**
The number of VE	**(VE Number)**
Memory size	**(Memory Size)**
The number of jobs	**(Job Number)**
Wait time for execution after submission to the queue	**(Run Wait Time)**
Wait time for execution after becoming assignable	**(Run Wait Time(ASS))**
Wait time for restart	**(Restart Wait Time)**
Based up for a request suspended by urgent request	**(Baseup Interrupted)**
Baseup for a rescheduled request	**(Baseup Reschedule)**
Baseup user defined	**(Baseup User Definition)**
Usage data of request priority	**(Pastusage Request Priority)**
Usage data of elapse time	**(Pastusage Elapse Time)**
Usage data of number of CPU	**(Pastusage CPU Number)**
Usage data of number of VE	**(Pastusage VE Number)**
Usage data of memory size	**(Pastusage Memory Size)**

- Scheduling priority at escalation(**Scheduling Priority (Escalation)**)

  Value of each elements

User share	**(User Share)**
User rank	**(User Rank)**
Request priority	**(Request Priority)**
Elapse time	**(Elapse Time)**
The number of CPU	**(CPU Number)**
The number of VE	**(VE Number)**
Memory size	**(Memory Size)**
The number of jobs	**(Job Number)**
Wait time for execution after submission to the queue	**(Run Wait Time)**
Wait time for execution after becoming assignable	**(Run Wait Time(ASS))**
Wait time for restart	**(Restart Wait Time)**
Based up for a request suspended by urgent request	**(Baseup Interrupted)**
Baseup for a rescheduled request	**(Baseup Reschedule)**
Baseup user defined	**(Baseup User Definition)**

Usage data of request priority	(**Pastusage Request Priority**)
Usage data of elapse time	(**Pastusage Elapse Time**)
Usage data of number of CPU	(**Pastusage CPU Number**)
Usage data of number of VE	(**Pastusage VE Number**)
Usage data of memory size	(**Pastusage Memory Size**)

- Scheduled start time (**Planned Start Time**)
- Scheduled end time (**Planned End Time**)
- Deadline time (**Deadline Time**)
- Deadline scheduling (**Deadline Scheduling**)

  ON Deadline request

  OFF Non-deadline request

- Elapse margin (**Elapse Margin**)
- Cloud bursting priority (**Cloud Bursting Priority**)
- Job server No. executing each job (**Job Server a Job belongs to (Job No. : JSV No.)**)
- Resource limit name and limit value information (**Resources Limits**)

  Elapsed time limit (**Elapse Time**)

  CPU number limit for each job (**CPU Number**)

  Memory size limit for each job (**Memory Size**)

- Scheduler message(**Scheduler Message**)
- Escalation limit reason(**Escalation Limit Reason**)

  'Min Forward Time' is over

  Side Escalation Difference Limit

  'Max Side Escalation Jobs' and 'No Escalation Period' is over.

  Side Escalation Number of Jobs Limit and Side Escalation Start Time Limit

  'Min Forward Time', 'Max Side Escalation Jobs' and 'No Escalation Period' is over.

  both conditions above-mentioned

JobManipulator Server Information (with the **-S** option specified)

- JobManipulator server host name (**JMServerHost**)
- Request run limit in the scheduler (**GRL**)

  ULIM Unlimited

- Request run limit per user in the scheduler (**URL**)

ULIM	Unlimited

- Assign limit per user in the scheduler (**UAL**)

ULIM	Unlimited

- The number of batch requests managed by JobManipulator (**TOT**)
- The number of batch requests for each state

ASG	ASSIGNED
EXC	EXCLUDE
EXT	EXITING
HLD	HELD
QUE	QUEUED
RUN	RUNNING
SUD	SUSPENDED

- State (**Status**)

Active	Active
No-Scheduling	Inactive

JobManipulator Server Host Information (with the **-S**,**-f** option specified)

- JobManipulator server host name (**JobManipulator ServerHost**)
- JobManipulator server version (**JobManipulator Version**)
- JobManipulator server state (**Status**)

Active	Active
No-Scheduling	Inactive

- Scheduler ID (**Scheduler ID**)
- Scheduling interval (**Schedule Interval**)
- Scheduling enable time (**Schedule Time**)
- Elapse Unlimited Scheduling (**Use Elapse Unlimited**)

ON	Unlimited
OFF	Elapse limit

- Request assign policy (**Request Assign Policy**)

Resource balance	Resource balanced assignment

CPU concentration CPU concentrated assignment

- Assign policy priority(**Assign Policy Priority**)

Network Topology	The assignment which is considered about network topology
Staging Job	Preferential assignment policy of the node without any staging job whose scheduled start time has been canceled.

- Request run limit in the scheduler (**Global Run Limit**)

  ULIMITED Unlimited

- Request run limit per group in the scheduler (**Group Run Limit**)

  ULIMITED Unlimited

- Request run limit for each group in the scheduler (**Group Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- CPU run limit per group in the scheduler (**Group CPU Run Limit**)

  ULIMITED Unlimited

- CPU run limit for each group in the scheduler (**Group CPU Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- VE run limit per group in the scheduler (**Group VE Run Limit**)

  ULIMITED Unlimited

- VE run limit for each group in the scheduler (**Group VE Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- Request run limit per user in the scheduler (**User Run Limit**)

  ULIMITED Unlimited

- Request run limit for each user in the scheduler (**User Run Limit of Each user**) (When **--limit** is specified.)

  ULIMITED Unlimited

- CPU run limit per user in the scheduler (**User CPU Run Limit**)

ULIMITED Unlimited

- CPU run limit for each user in the scheduler (**User CPU Run Limit of Each user**) (When **--limit** is specified.)

  ULIMITED Unlimited

- VE run limit per user in the scheduler (**User VE Run Limit**)

  ULIMITED Unlimited

- VE run limit for each user in the scheduler (**User VE Run Limit of Each user**) (When **--limit** is specified.)

  ULIMITED Unlimited

- Assign limit per user (**User Assign Limit**)

  ULIMITED Unlimited

- Half-life decay (**Half-Reduce Period**)
- Standard CPU number (**Standard CPU Number**)
- Standard VE number (**Standard VE Number**)
- Standard memory size (**Standard Memory Size**)
- Standard Job number (**Standard Job Number**)
- Escalation (**Escalation**)

Forward_Escalation	Escalation without change of the node
All_Escalation	Escalation in all node
OFF	Disable escalation

- Escalation Interval (**Escalation Interval**)
- Action for non-reservable request (**Treat unbookable request**)

delete	Deletes request
after	Assigns the earliest and executable time after specified time

- Method for interrupted requests (**Interruption method**)

suspend	Suspends the request
rerun	Rerun the request

- Block of assignment by urgent request (**Interrupt assign block**)

  ON Prohibits the assignment of newly submitted lower-level type requests to the execution host to which an urgent request waiting to be executed is assigned.

  OFF Allows the assignment of newly submitted lower-level type requests to the execution host to which an urgent request waiting to be executed is assigned.

- Where high-priority requests interrupt to (**Interrupt to where**)

next_run	Head of assigned requests waiting for execution
current	Current time

- Auto Delete Resource Reservation function (**Auto Delete Resource Reservation**)

ON	Auto Delete Resource Reservation function is valid.
OFF	Auto Delete Resource Reservation function is invalid.

- Forced re-scheduling function (**Forced Re-Scheduling**)

ON	The stalled , non-bursted running request is rescheduled
OFF	The stalled, non-bursted running request is not rescheduled

- Forced re-scheduling function of bursted running request (**Forced Re-Scheduling(Cloud)**)

ON	The stalled , bursted running request is rescheduled
OFF	The stalled, bursted running request is not rescheduled

- Dynamic DC Control function (**Dynamic DC Control**)

ON	Dynamic DC Control is started.
OFF	Dynamic DC Control is stopped.

- The maximum number of operation nodes (**Max Operation Hosts**)
- Peak cut degree of urgency (**Peak Cut Urgency**)

right_now	A node is powered off immediately.
wait_run	A node is powered off after the running request is finished.

- The minimum idle time (**Minimum Idle Time**)
- The threshold of estimated DC-OFF time (**Estimated DC-OFF Time**)
- The DC-OFF limit (**DC-OFF Limit**)
- Overtake limit control (**Use Overtake Priority**)
    - Execution queue = normal type (**normal**)

ON	Limit for overtake is valid
OFF	Limit for overtake is invalid

    - Execution queue = special type (**special**)

ON	Limit for overtake is valid
OFF	Limit for overtake is invalid

- Priority to limit overtake (**Overtake Priority**)
    - Execution queue = normal type (**normal**)

    - Execution queue = special type (**special**)
- Overtaking-related settings(**Overtake**)
    - Overtaking assignment function for small requests(**Allow Small Request**)

      ON Enable overtaking assignment function for small requests.

      OFF Disable overtaking assignment function for small requests.

- Map maintenance time after HW failure (**Keep Forward Schedule**)
- Stage-in margin (**Stage-in Margin**)
    - Adjusting time of estimated stage-in time(**Additional Margin for Escalation**)
    - Threshold of stage-in(**Stage-in Threshold**)
    - First stage-in time(**First Stage-in Time**)
- Waiting time of rescheduling of starting of virtual machine (VM), baremetal server or container which fails in starting by the provisioning environment(**Provisioning Start Retry Time)**
- Scheduling method(**Scheduling Method**)
    - Setting of the feature of setting of scheduling method (**VE Degradation**)

      Continue schedule with the change in the number of VE node

      Exclude exclude VI with degraded VE from the targets of scheduling

      Auto Exclude VI with degraded VE from the targets of scheduling if 1 or more VE degrade. Include VI whose all degraded VE recover to the targets of scheduling.

    - Setting of the request assignment mode(**Assign Mode**)

      Continue The scheduling process is not terminated at a certain time but continues.

      Reset The scheduling process will be terminated at a certain time.

    - Setting of the caching of non-schedulable requests (**Non Scheduled Request Cache**)

      ON Enable

      OFF Disable

- Immediate scheduling of requests(**Realtime Scheduling**)

  ON Enable Realtime Scheduling

  OFF Disable Realtime Scheduling

- Mode of Immediate scheduling of requests(**Realtime Scheduling Mode**)

  Default Immediate scheduling of requests is not done in the time zone for event processing.

  Always Always perform immediate scheduling of requests.

- HCA device group topology scheduling (**Device Group Topology**)

ON	Enable HCA Device Group Topology Scheduling
OFF	Disable HCA Device Group Topology Scheduling

- Maximum number of requests to burst into cloud resources at one time (**Cloud Bursting Request Number**)
- Cloud bursting priority weight (**Cloud Bursting Priority Weight**)

Resource	Weight for the amount of requested resources
Wait Time	Weight for execution wait time
Scheduling Priority	Weight for scheduling priority
Custom Resource	Weight for custom resources

- The number information of batch requests managed by JobManipulator (**Request Statistical Information**)
    - The number of batch requests managed by JobManipulator (**Total Request**)
    - The number of batch requests for each state

ASSIGNED	(**Assigned Request**)
EXITED	(**Exited Request**)
EXITING	(**Exiting Request**)
HELD	(**Held Request**)
POST-RUNNING	(**Post-running Request**)
PRE-RUNNING	(**Pre-running Request**)
QUEUED	(**Queued Request**)
RESUMING	(**Resuming Request**)
RUNNING	(**Running Request**)
STAGING	(**Staging Request**)
STAGED	(**Staged Request**)
SUSPENDED	(**Suspended Request**)
SUSPENDING	(**Suspending Request**)
WAITING	(**Waiting Request**)

- Logfile information (**Logfile Information**)
    - Logfile path name (**Logfile Path**)
    - Logfile level (**Logfile level**)
    - Maximum logfile size (**Logfile MAX Size**)

    - Logfile save count (**Logfile Save Count**)
- Job servers bound to JobManipulator (**Bind JobServer**)
    - Job servers which use licenses (**Use Licenses**)
    - Job servers which do not use licenses (**Unuse Licenses**)

Scheduling Priority(with the **-M** option specified)

- Timing of calculation for scheduling priority

  When assigns request (**Assign**)

  When escalates request (**Escalation**)
- Weight coefficient (**Scheduling Weight**)
    - User share value (**User Share**)
    - User rank value (**User Rank**)
    - Request priority (**Request Priority**)
    - Elapse time (**Elapse Time**)
    - The number of CPUs (**CPU Number**)
    - The number of VEs (**VE Number**)
    - Memory size (**Memory Size**)
    - The number of jobs (**Job Number**)
    - Wait time for execution after submission to the queue(**Run Wait Time**)
    - Wait time for execution after becoming assignable(**Run Wait Time(ASS)**)
    - Wait time for restart (**Restart Wait Time**)
- Base-up (**Baseup**)
    - Request suspended by urgent request (**Interrupted**)
    - Request to be re-scheduled (**Reschedule**)
- PastUsage (**PastUsage**)
    - Request priority (**Request Priority**)
    - Elapse time (**Elapse Time**)
    - The number of CPUs (**CPU Number**)
    - The number of VEs (**VE Number**)
    - Memory size (**Memory Size**)
- PastUsage weight coefficient for User rank value (**PastUsage(User Rank)**)
    - Request priority (**Request Priority**)
    - Elapse time (**Elapse Time**)
    - The number of CPUs (**CPU Number**)
    - The number of VEs (**VE Number**)

- Memory size (**Memory Size**)

Queue Information(with the **-Q** option specified)

- Execution queue name (**QueueName**)
- Queue type (**Type**) (This is not displayed for an interactive queue.)
- Request run limit in the queue (**RL**)

  ULIM Unlimited

- Request run limit per user in the queue (**URL**)

  ULIM Unlimited

- Request assign limit per user in the queue (**UAL**)

  ULIM Unlimited

- The number of batch requests per queue (**TOT**)
- The number of batch requests per scheduling status

ASSIGNED	(**ASG**)
EXCLUDE	(**EXC**)
EXITING	(**EXT**)
HELD	(**HLD**)
QUEUED	(**QUE**)
RUNNING	(**RUN**)
SUSPENDED	(**SUD**)

Queue Information (with the **-Q**,**-f** option specified)

- Execution queue name (**Queue Name**)
- Queue type (**Queue Type**) (This is not displayed for an interactive queue.)
- Map width (**Schedule Time**)
- Request assign policy (**Request Assign Policy**)

Resource balance	Resource balanced assignment
CPU concentration	CPU concentrated assignment
DEFAULT	According to the setting of the scheduler

- Where high-priority requests interrupt to. **Interrupt to where**)
  (This is not displayed for an interactive queue.)

  next_run Head of assigned requests waiting for execution

  current Current time

  DEFAULT According to the setting of the scheduler

- Request run limit in the queue (**Run Limit**)

  UNLIMITED Unlimited

- Request run limit per group in the queue (**Group Run Limit**)

  ULIMITED Unlimited

- Request run limit for each group in the queue (**Group Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- CPU run limit per group in the queue (**Group CPU Run Limit**)

  ULIMITED Unlimited

- CPU run limit for each group in the queue (**Group CPU Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- VE run limit per group in the queue (**Group VE Run Limit**)

  ULIMITED Unlimited

- VE run limit for each group in the queue (**Group VE Run Limit of Each Group**) (When **--limit** is specified.)

  ULIMITED Unlimited

- Request run limit per user in the queue (**User Run Limit**)

  UNLIMITED Unlimited

- Request run limit for each user in the queue (**User Run Limit of Each User**) (When **--limit** is specified.)

  ULIMITED Unlimited

- CPU run limit per user in the queue (**User CPU Run Limit**)

  UNLIMITED Unlimited

- CPU run limit for each user in the queue (**User CPU Run Limit of Each User**) (When **--limit** is specified.)

ULIMITED Unlimited

- VE run limit per user in the queue (**User VE Run Limit**)

  UNLIMITED Unlimited

- VE run limit for each user in the queue (**User VE Run Limit of Each User**) (When **--limit** is specified.)

  ULIMITED Unlimited

- Request assign limit per user in the queue (**User Assign Limit**)

  UNLIMITED Unlimited

- Elapse margin (**Elapse Margin**)
- Cloud bursting scheduler map occupancy threshold (**Cloud Bursting Occupancy Ratio**)
- Current cloud bursting scheduler map occupancy threshold (**Current Cloud Bursting Occupancy Ratio**)
- Current cloud bursting availability status (**Current Cloud Bursting Status**)

  ENABLE Currently the cloud bursting is enabled

  DISABLE Currently the cloud bursting is disable

- Request cloud bursting priority display availability information (**Display Cloud Bursting Priority**)

  ON Request cloud bursting priority is displayed

  OFF Request cloud bursting priority is not displayed

- Conditions of selecting target requests of escalation. (**Escalation Limit**)
  - Maximum of the number of jobs with Side Escalation (**Max Side Escalation Jobs**)

    DISABLE disabled

  - No escalation period of time from current time (**No Escalation Period**)

    DISABLE disabled

  - Minimum difference between before escalation and after escalation (**Min Forward Time**)

    DISABLE disabled

- Wait time of rescheduling (**Retry Time**)
  - When Stage-in processing failed (**Staging**)
  - When PRE-RUNNING processing failed (**Pre-running**)
- Wait time of stage-out processing (**Wait_Stageout**)
- Deadline control information (**Deadline Control**) (This is not displayed for an interactive queue.)

    - Deadline scheduling (**Deadline Scheduling**)

      ON Deadline scheduling is enabled.

      OFF Deadline scheduling is disabled.

    - Minimum reduce rate (**Minimum Reduce Rate**)
    - Reduce rate when requests terminates on time (**Ontime Reduce Rate**)
    - Maximum reduce rate (**Maximum Reduce Rate**)
    - Reduce rate decreases until the end time (**End Time of Rate Decrease**)
    - Reduce rate decreases from the start time (**Start Time of Rate Decrease**)
    - Reduce rate increases until the end time (**End Time of Rate Increase**)
    - Reduce rate increases from the start time (**Start Time of Rate Increase**)
- Minimum number of operation nodes for the queue (**Min Operation Hosts**)
- Reservation Margin (**Reservation Margin**)
    - PRE-MARGIN (**Pre-margin**)
    - POST-MARGIN (**Post-margin**)
- The minimum network topology node group selection (**Network Topology Control**) (If the minimum network topology node group selection is enabled)
    - Enable/Disable setting (Network Topology Minimum Scheduling)

      ON Enable the minimum network topology node group selection.
    - Number of execution hosts per minimum network topology node group (Hosts per group)

      JobManipulator calculates automatically. "(Default)" is displayed after the value.
- Schedule Type (**Schedule Type**)

  Backfill Enable Backfill Scheduling

  FIFO Enable FIFO Scheduling
- JSV assign priority information (**JSV Assign Priority**) (If the **-j** option specified) (This is not displayed for an interactive queue.)
- Dynamic JSV Priority information(**JSV Dynamic Assign Priority**) (If the **-j** option specified) (This is not displayed for an interactive queue.)
- Dynamic JSV Priority Control (**Dynamic JSV Priority Control**) (This is not displayed for an interactive queue.)

  ON Enable

  OFF Disable
- Dynamic JSV Priority item value for priority calculation (**Dynamic JSV Priority Item**) (This is not displayed for an interactive queue.)
- The number of batch requests controlled by JobManipulator (**Request Statistical Information**)
    - The number of batch requests controlled by JobManipulator (**Total Request**)

- The number of batch requests per scheduling status

- Elapse margin (**Elapse Margin**)
- The number of batch requests per execution queue

ASSIGNED	**(Assigned Request)**
EXCLUDE	**(Exclude Request)**
EXITED	**(Exited Request)**
EXITING	**(Exiting Request)**
HELD	**(Held Request)**
POST-RUNNING	**(Post-running Request)**
PRE-RUNNING	**(Pre-running Request)**
QUEUED	**(Queued Request)**
RESUMING	**(Resuming Request)**
RUNNING	**(Running Request)**
STAGED	**(Staged Request)**
STAGING	**(Staging Request)**
SUSPENDED	**(Suspended Request)**
SUSPENDING	**(Suspending Request)**
WAITING	**(Waiting Request)**

Execution Host Information (with the **-E** option specified)

- Execution host name (**ExecutionHost**)
- Limit of the usable ratio of CPUs (**CPURatio**)
- Limit of the usable ratio of memory size (**MemRatio**)

Execution Host Information (with the **-E --eco-status** option specified)

- Execution host name (**ExecutionHost**)
- Status of power-saving (**EcoStaus**)

EXCLUDED	Excluded from the targets of power-saving control
PEAKCUT	Peak Cut

-	Other than those above

- State transition time of power-saving (**StateTransitionTime**)
- DC-OFF times of current day (**OFF(D)**)
- Accumulative DC-OFF times (**ACCUM**)

Execution Host Information (with the **-E --eco-status -f** option specified)

- Execution host name (**Execution Host**)
- Status of power-saving control (**Eco Status**)
- State transition time of power-saving control (**State Transition Time**)
- The reason why the node has been excluded (**Exclude Reason**)

ACTIVE_LINKDOWN	The node is started but the Job Server is LINKDOWN
HW_FAILURE	HW failure
START_FAIL	Failed to start
Cloud Bursting Node	Node starting with cloud bursting function

- DC-OFF times of current day (**DC-OFF Times (Day)**)
- Accumulative DC-OFF times (**DC-OFF Times (ACCUM)**)

Execution Host Information (with the **-E --hw-failure** option specified)

- Execution host name (**ExecutionHost**)
- Status of HW failure (**Status**)

ABNORMAL STOP	A failure informed by the Node Agent
DEGRADED	VE degraded but included in the target of scheduling
EXCLUDED	Excluded from the targets of scheduling due to the failure of stopping of virtual machine (VM), baremetal server or container under environment of provisioning. or excluded from the targets of scheduling by VE nodes degradation

Execution Host Information (with the **-E -f** option specified)

- Execution host name (**Execution Host**)
- The limit of simultaneous usable number of CPUs on the execution host(**CPU Number Ratio**)
- The limit of simultaneous usable number of CPUs for each RSG when the resources are divided by RSG(**CPU Number Ratio of RSG**)
- The limit of simultaneous usable memory size on the execution host(**Memory Size Ratio**)
- The limit of simultaneous usable memory size for each RSG when resources are divided by RSG (**Memory Size Ratio of RSG**)
- Link state (**LINK Batch Server**)

UP	Linkup
DOWN	Linkdown

- Bind state (**BIND Queue**)

BIND	Bind
UNBIND	Unbind

- The eco-state of the execution hosts (**Eco Status**)
    - Status of power-saving control (**Eco Status**)
    - State transition time of power-saving control(**State Transition Time**)
    - The reason why the node has been excluded(**Exclude Reason**)

ACTIVE_LINKDOWN	The node is started but the Job Server is LINKDOWN
HW_FAILURE	HW failure
START_FAIL	Failed to start
Cloud Bursting Node	Node starting with cloud bursting function

    - DC-OFF times of current day (**DC-OFF Times (Day)**)
    - Accumulative DC-OFF times(**DC-OFF Times(ACCUM)**)

- Status of HW failure of execution host (**Hardware Failure**)
    - Status of HW failure(**Status**)

ABNORMAL STOP	A failure informed by the Node Agent
DEGRADED	VE degraded
EXCLUDED	Excluded from the targets of scheduling due to the failure of stopping of virtual machine (VM), baremetal server or container under environment of provisioning.

or exclude from the targets of scheduling by feature of excluding VI from the targets of scheduling

  - Exclude reason (**Exclude Reason**)

BAREMETAL_ABNORMAL_LINKDOWN	The Job Server that is started on baremetal server of provisioning environment is LINKDOWN incorrectly
BAREMETAL_UNKNOWN_TEMPLATE	Running under unknown template information on provisioning environment
BAREMETAL_DELAY_STAGEIN	Stage-in is delaying on provisioning environment
BAREMETAL_DELAY_STAGEOUT	Stage-out is delaying on provisioning environment
STOPFAIL_PROVISIONING	Failed to stop on provisioning environment
VE_DEGRADATION	VE degraded

  - The information VE node degraded or not (**VE Degradation**)

YES	Degraded
(without display)	Not degraded

- Scheduling state of execution host (**Scheduling**)
  - Scheduling target or not (**Status**)

Available	Target of scheduling
Unavailable	Not target of scheduling

  - Exclusion reason of the scheduling target (**Reason**)

NQS API Error	Failure of getting resource information by an error of API
No License	There are no license

- Waiting time of retry for starting of provisioning (**Provisioning Waiting Retry**)
  - template name = the ending time of the retry

Map Information (with the **-A** option specified)

- Job server number (**JSV**)
- Job ID (**Job-ID**)
- CPU run limit (**CPU**)
- Memory size limit (**Memory**)

Resource Reservation Section Information (with the **-B** option specified)

- Resource Reservation Section ID (**RES ID**)
- Start time of Resource Reservation Section (**Start Time**)
- Period of Resource Reservation Section (**End Time**)
- The number of execution hosts used in Resource Reservation Section (**NodeNum**)
- The number of CPUs per execution hosts in Resource Reservation Section (**CPUNum**)
- Execution queue (**Queue**)
  When the Resource Reservation Section is created by specifying the execution queue.
- Group Name (**GrpName**)
  When **--group** is specified.

Resource Reservation Section Information (with the **-B --template** option specified) (for Resource Reservation Section specifying template)

- Resource Reservation Section ID (**RES ID**)
- Start time of Resource Reservation Section (**Start Time**)
- Period of Resource Reservation Section (**End Time**)
- The template name specified at creation of reservation (**Template**)
- The number of machines specified at creation of reservation (**MacNum**)
- Execution queue (**Queue**)
- Group Name (**GrpName**)
  When **--group** is specified.

Resource Reservation Section Information (with the **-B**,**-f** option specified)

- Resource Reservation Section ID (**Resource Reservation ID**)
- Resource Reservation Section Name (**Resource Reservation Name**)

- Group Name (**Group Name**)
- Execution queue (**Queue Name**)
  When reserved section is created by specifying the execution queue.
- Start time of Resource Reservation Section (**Reserve Start Time**)
- Period of Resource Reservation Section (**Reserve End Time**)
- The number of execution hosts used in Resource Reservation Section (**Execution Host Number**)
- The number of CPUs per execution hosts in Resource Reservation Section (**Reserve CPU Number by Host**)
- Execution host name used in Resource Reservation Section (**Reserved Hosts (HOST_NAME : STATUS)**)
- Requests uses this reservation area (**Requests uses this reservation area**)

Resource Reservation Section Information (with the **-B --template**, **-f** option specified) (for resource reservation section specifying template)

- Resource Reservation Section ID (**Resource Reservation ID**)
- Resource Reservation Section Name (**Resource Reservation Name**)
- Group Name (**Group Name**)
- Execution queue (**Queue Name**)
- Start time of Resource Reservation Section (**Reserve Start Time**)
- Period of Resource Reservation Section (**Reserve End Time**)
- The template name specified at creation of reservation (**Reserve Template**)
- The number of machines specified at creation of reservation (**Reserve Machine Number**)
- Execution host name used in Resource Reservation Section (**Reserved Machines (HOST_NAME : STATUS)**)
- Requests uses this reservation area (**Requests uses this reservation area**)

Complex Queue Information (with the **-C** option specified)

- Complex queue name (**QueueName**)
- Belonging queue(**QueueName**)
- Belonging queue type(**Type**)
- Request run limit in a complex queue (**RL**)

  ULIM Unlimited

- User run limit in a complex queue (**URL**)

ULIM Unlimited

- User assign limit in a complex queue (**UAL**)

  ULIM Unlimited

- The number of batch requests per queue and complex queue (**TOT**)
- The number of batch requests per scheduling status

ASSIGNED	(**ASG**)
EXCLUDE	(**EXC**)
EXITING	(**EXT**)
HELD	(**HLD**)
QUEUED	(**QUE**)
RUNNING	(**RUN**)
SUSPENDED	(**SUD**)

Power-saving Schedule Information (with the **-D** option specified)

- Power-saving Schedule ID (**EcoID**)
- Start time of power-saving schedule (**EcoStartTime**)
- End time of power-saving schedule (**EcoEndTime**)
- Host name of the target execution host in power-saving schedule (**ExecutionHost**)

Power-saving Schedule Information (with the **-D**,**-f** option specified)

- Power-saving Schedule ID (**Eco Schedule ID**)
- Start time of power-saving schedule (**Scheduled Start Time**)
- End time of power-saving schedule (**Scheduled End Time**)
- The number of the target execution hosts in power-saving schedule (**Number of Scheduled Hosts**)
- Host name of the target execution host in power-saving schedule (**Scheduled Hosts**)

Job Server Information (with the **-J** option is specified)

- The number of job server(**JSVNO**)

- Queue Name (**Queue**)
- JSV Assign Priority (**Priority**)
- Dynamic JSV Priority (**Dynamic Priority**) (with the **--dynamic** option specified)

ENVIRONMENT VARIABLES

JM_PRIV

Default access privilege for command execution

JM_PRIV_MGR	(Manager privilege)
JM_PRIV_OPE	(Operator privilege)
JM_PRIV_GMGR	(Group manager privilege)
JM_PRIV_SPU	(Special user privilege)
JM_PRIV_USR	(User privilege)

JM_SCHED_ID

Default scheduler ID of JobManipulator

FILE

/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf

1.23. wdel(1)

NAME

wdel -- Delete Workflow

SYNOPSIS

wdel [-P *privilege*] [-V] *wfl-id* ...

DESCRIPTION

wdel deletes workflows specified by *wfl-id*. Two or more workflow identifiers can be specified in *wfl-id* by delimiting at space.
wdel deletes specified workflows and all requests which are submitting and executing in a workflow.

The following options can be specified:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege
o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV. If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege

PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

wstart(1)

1.24. wstart(1)

NAME

wstart -- Execute Workflow

SYNOPSIS

wstart [-V] [--cancel-after] [*wfl-script*]

DESCRIPTION

wstart executes the shell script file specified in *wfl-script* as a workflow. If *wfl-script* isn't specified, it reads a workflow script from stdin.

wstart displays the workflow identifier got from a batch server to stdout.

wstart executes a workflow script as a child process. And wstart waits for an end of Executing Process of the workflow script. A workflow of a NQSV request can be described using qsub(1), qwait(1) and qwait2(1) in the workflow script.
The following is examples of a workflow script:

```
qsub -q batch1 -N REQ-A ./job_A.sh
qsub -q batch2 -N REQ-B ./job_B.sh
qsub -q batch2 -N REQ-C --after REQ-A,REQ-B ./job_C.sh
qwait2 REQ-C
```

Stdin, stdout and stderr of the workflow script are input and output at the terminal where wstart was executed.
The workflow which is executing can be deleted by wdel(1) or it can be deleted by pushing down ctrl-C on the terminal where wstart was executed.
In this case, all requests submitted in the workflow are deleted, and wstart ends.
When the workflow is deleted by wdel(1) and wstart has ended, wstart displays the message to stderr.

A workflow of the NQSV request executed by wstart exists in the NQSV system, until wstart ends and all requests submitted in the workflow script end. Therefore it can be displayed by the wstat(1) command.

The following options can be specified:

-V Display only the version of the command and NQSV API.

--cancel-after

Specify "--cancel-after" option to qsub(1) command all together in workflow script.

SEE ALSO

qsub(1), qwait(1), qwait2(1), wdel(1), wstat(1)

1.25. wstat(1)

NAME

wstat -- Display Workflow

SYNOPSIS

wstat [-P *privilege*] [-V] [*wfl-id* ...]

DESCRIPTION

wstat displays the information of the workflow.

When *wfl-id* is omitted, all workflows owned by command executer will be the objects.

The following items are displayed :

WFL-ID	Workflow identifier
Request	Number of requests submitted in the workflow. The request which has ended is also included in that. For a parametric request, it is the number of the sub-requests.
Owner	Owner name of the workflow
Stat	State of the workflow

When *wfl-id* is specified, wstat displays the information of the request which was submitted in the specified workflow. Two or more workflow identifiers can be specified in *wfl-id* by delimiting at space.

The following items are displayed :

WFL-ID	Workflow identifier
RequestID	Request identifier
ReqName	Request name
Stat	State of the request
Exit	Request exit code used hexadecimal When the request has not ended, '-' is displayed.

The following options can be specified:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	Manager privilege

o	Operator privilege
g	Group manager privilege
s	Special user privilege
u	User privilege

Among access privileges, Manager privilege is highest and User privilege is lowest. When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV. If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V Display only the version of the command and NQSV API.

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege
PRIV_OPE	Operator privilege
PRIV_GMGR	Group manager privilege
PRIV_SPU	Special user privilege
PRIV_USR	User privilege

SEE ALSO

wstart(1)

2. Administrator's Commands

With NQSV-ResourceManager package nmapmgr is installed under /opt/nec/nqsv/bin and nqs_ntfr is installed under /opt/nec/nqsv/sbin. Others are installed under /opt/nec/nqsv/bin with NQSV-Client package.

Commands	Contents
budgetedit(1M)	Displays and Edits Budget
nmapmgr(1M)	Management of NQSV Machine ID Database
nqs_ntfr(1M)	Notify BSV of failures of execution host
qmgr(1M)	Configuration and Operation Management of NQSV
subedit(1M)	Displays and Edits the Accounting Rate
sceditacct(1M)	Edit Accounting Database
scsumacct(1M)	Displays Daily and Monthly Accounting Information
scturnacct(1M)	Saves Each Type of Accounting Data
smgr(1M)	Configuration and Operation Management of JobManipulator
sushare(1M)	User Share Management Tool

2.1. budgetedit(1M)

NAME

budgetedit -- displays and edits budget

SYNOPSIS

/opt/nec/nqsv/bin/budgetedit {add|upd} {-u|-g|-a} *name*:*initial*[/*remain*/*estimate*]

/opt/nec/nqsv/bin/budgetedit del {-u|-g|-a} *name*

/opt/nec/nqsv/bin/budgetedit -N

DESCRIPTION

The budgetedit command displays, registers, updates, and deletes budget setting information and result information of the budget management items.

If the command is executed with no options, all the registered budget settings and results are displayed.

```
====================================================
USER         REMAIN        ESTIMATE          INITIAL
====================================================
usr1         11223.41        1200.00        12245.00
usr2       1395382.88           0.00      1399445.00
usr3        126555.98        3000.00       126555.98

====================================================
GROUP        REMAIN        ESTIMATE          INITIAL
====================================================
grp5         11223.41        1200.00        12245.00
grp4       1395382.88           0.00      1399445.00

====================================================
ACCOUNT      REMAIN        ESTIMATE          INITIAL
====================================================
acct1        11223.41        1200.00        12245.00
acct3      1395382.88           0.00      1399445.00
```

OPTIONS

The budgetedit command specifies the operation, application range, target and budget settings as follows.

```
    # budgetedit add -u usr1:192345.87/1229445.98/0.00
```

add Adds a new budget setting. If the specified target already exists, an error message is displayed.

upd Updates the budget setting. If the specified target does not exist, an error message is displayed. If you want to increase or decrease the budget, please specify only *initial*. The balance is automatically updated. If you want to modify the budget data directly for maintenance, please specify all of *initial/remain/estimate*.

You can specify the amount in one of the following ways.

- Specified by absolute value
  Change the budgeted amount in absolute value. Please specify without a sign. Mixing unsigned and signed values causes a specification error.

```
# budgetedit add -u usr1:9999/8888/7777
```

- Specified by relative value
  Specify the amount of money to be added or subtracted from the budgeted amount in relative value. A sign is required for the amount you specify. You cannot mix negative and positive signs. If you do not need to change the value, please specify 0.

```
# budgetedit add -u usr1:+1000/+1000/+1000
```

del Deletes the budget setting. If the specified target does not exist, an error message is displayed.

-N Displays the budgetedit version information. This option takes priority even when other options are specified, and they are disregarded.

-u Sets a budget setting for each user.

-g Sets a budget setting for each group.

-a Sets a budget setting for each account code.

name Specifies target of a user (-u), group (-g), or account code (-a) to which a budget is set. Specifies name of account code written in the account code file (/etc/opt/nec/nqsv/acctcode) as account code. The maximum length of name of account code is 15 bytes.

initial Sets the initial accounting budget.

remain

Sets the remaining budget. Although the budget result is usually updated by the Accounting server, this command can perform operations such as clearing the budget result.

estimate

Sets the estimate of billing. Although the estimate is usually updated by the Accounting server, this command can perform operations such as clearing the budget result.

SEE ALSO

sbuedit(1M-N), budgetcheck(1-N).

2.2. nmapmgr(1M)

NAME

nmapmgr -- Management of NQSV Machine ID Database

SYNOPSIS

nmapmgr

DESCRIPTION

Command nmapmgr manages the machine ID database used by the NQSV. This database contains the following four basic elements:

alias

An alias of the principal-name of a machine in the network. alias are understood only by local NQS hosts. An alias is a character string less than 255 bytes in length. Machines with several IP addresses sometimes have different names for individual addresses.
Aliases are used in these cases.

mid A machine ID number allocated to be unique among machines in the architecture of the NQSV network. The NQSV uses this mid to identify a specified machine. The minimum value of this value is 0. The maximum value of this value is (2^31)-1.

principal-name

A unique host name in the machine ID database managed by the NQSV. The network is accessed using this name.

type The type of the NQS that is supported. The following types are currently available.

nqsv	NQSV (Default)

Executing Command nmapmgr will display Prompt NMAPMGR>:. The following subcommand can be input in this condition. Use Subcommand exit to return to the shell.
Only the super users can execute operations that alter the NQSV network configuration.

COMMANDS

All commands must be input in one line.

Add Mid *mid principal-name*

Adds a new machine to the NQSV network configuration by specified *mid* and *principal-name*.

Add Name *alias to-mid*

Adds an *alias* of the Machine of specified *to-mid*.

Change Name *mid principal-name*

Changes the *principal-name* of the machine of specified *mid*.

Create

Clears the machine ID database.

Delete Mid *mid*

Deletes the specified *mid* from the machine ID database.

Delete Name *alias*

Deletes the specified *alias* from the machine ID database.

Exit

Exits Command nmapmgr.

Get Mid *name*

Displays mid of the machine of a specified principal-name or alias.

Get Name *mid*

Displays the principal-name of the machine of specified *mid*.

Help

Displays a list of subcommands.

Quit

Exits Command nmapmgr.

Set Type *mid type*

Sets the type of NQS specified by *type* in the Machine of specified *mid*.

Show State [*name*]

Displays mapping states of a principal-name or an alias specified by *name*. Omitting *name* will display mapping states of all principal-names and aliases.

FILES

/etc/opt/nec/nqsv/nmap

/etc/opt/nec/nqsv/nmap/machines

NOTES

This command is based on the command developed by Sterling Software Inc. and incorporates extended functions.

2.3. nqs_ntfr(1M)

NAME

nqs_ntfr -- Notify BSV of failures of execution host

SYNOPSIS

nqs_ntfr [-V] [-o *operation*] [-m *message*] *hostname* | *IP_Address*

nqs_ntfr -r *hostname* | *IP_Address*

nqs_ntfr -R *hostname* | *IP_Address*

nqs_ntfr -s *hostname* | *IP_Address*

DESCRIPTION

nqs_ntfr command notifies BSV of failures that occurs on execution host specified with *hostname* or *IP_address.*

Operation taken by BSV against JSV on receiving failure notification is specified with *operation* option.

The following options can be specified. -r, -R, and -s are mutually exclusive.:

-m *message*

Specify failure description as *message*

The maximum length of *message* is 255 bytes.

-o *operation*

Specify operation against JSV on failure notification

As operation, either of the followings can be specified.

Note that JSV can be operated only when JSV is in LINKUP state.

nothing	No operation against JSV
down	LINKDOWN JSV
unbind	UNBIND JSV from queue

If -o option is not specified, the default is down.

-r

Redundancy function option to direct the start of recovery of managed host on which failed BSV and/or JM are working.

-R

Redundancy function option to notify recovery completion of managed host on which failed BSV and/or JM are working. This option can be also used to notify recovery completion of substitute host which failed to start or stop.

-s

Redundancy function option to notify failure occurrence of managed host on which BSV and/or JM are working.

-V Display version of command and version of NQSV API in use and then exit.

SEE ALSO

qrls(1),qhold(1),qstat(1)

2.4. qmgr(1M)

NAME

qmgr -- Configuration and Operation Management of NQSV

SYNOPSIS

qmgr [-P *privilege*] [-V] [-h *batch_server_host*]

DESCRIPTION

qmgr is the program used by the system manager and system operators to control the NQSV system in *batch_server_host*.

The following options can be specified.

-h *batch_server_host*

Specify a batch server for control of the NQSV system.
In case this option is not specified, the default batch server (batch-server host name specified in /etc/opt/nec/nqsv/api_client.conf) will be used. If a default batch server is not located, the batch server host is judged as a local host.

-P *privilege*

Specify access privilege for command execution.
The following can be specified in *privilege*.

m	Manager privilege.
o	Operator privilege.
g	Group manager privilege.
s	Special user privilege.
u	User privilege.

Among access privileges, Manager privilege is highest and User privilege is lowest.
When -P option is not specified, access privilege for command execution is set by Environment Variable NQSV_PRIV or NQSII_PRIV.
If NQSV_PRIV or NQSII_PRIV is not set, User privilege is set up.

-V

Display only the version of the command and NQSV API.

Activating Command qmgr, qmgr returns Prompt Mgr: and inputs the sub-commands that are describes below. Execute Command exit to terminate Command qmgr.
The following edit commands can be used in Mgr: prompt lines.

Ctrl-f	Moves one character in forward direction.
Ctrl-b	Moves back one character in reverse direction.
Ctrl-h	Deletes characters in front.
Ctrl-p	Displays an input command one before.
Ctrl-n	Displays an input command one after.
<TAB>	Displays or complements a sub-command that matches a character string that is input.

For help sub-command and show sub-command, The output can be displayed by pager program. The pager program can be specified with PAGER Environment Variables. The default pager program is /bin/more.

SUB-COMMANDS

The following sub-commands are supported. qmgr sub-commands are not case sensitive. The key words may be which of the uppercase, the lowercase.
Three types of access privilege are available for the qmgr sub-commands.
The Manager privilege is allowed to use all commands, while the Operator privilege is allowed to use only limited commands. User privilege and Special user privilege are allowed to use only sub-commands exit, help, list and show.
For details of the subcommand of the Budget Control Function, please refer to [Accounting & Budget Control].

abort execution_queue = *queue* [*seconds*]

All requests currently residing in an execution *queue* specified in *queue* are aborted as follows: (All RUNNING, SUSPENDING,SUSPENDED,RESUMING and CHKPNTING requests)

First, Signal SIGTERM is sent to each process of requests currently executed in the execution queue.
Next, after seconds specified in *seconds* are over, Signal SIGKILL is sent to remaining processes of the requests that are executed in a specified execution queue. If *seconds* is not specified, the delay will be 60 seconds. This command

deletes all aborted requests from execution queues and result files of the requests are returned.

The owner of the request or manager privilege is needed.

abort interactive_queue = *queue* [*seconds*]

All requests currently residing in an interactive queue specified in *queue* are aborted as follows:(All RUNNING, SUSPENDING, SUSPENDED and RESUMING requests)

First, Signal SIGTERM is sent to each process of requests currently executed in the execution queue.
Next, after seconds specified in *seconds* are over, Signal SIGKILL is sent to remaining processes of the requests that are executed in a specified execution queue. If *seconds* is not specified, the delay will be 60 seconds.
This command deletes all aborted requests from execution queues.

The owner of the request or manager privilege is needed.

add execution_queue groups = *group queue*
add execution_queue groups =(*group*[,*group*...]) *queue*

Adds groups specified in *group* to the group list of execution queue specified in *queue*. The mode of the group list needs to be set by sub-command **set execution_queue access | noaccess**.

The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and *group-id* is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

add execution_queue users = *user queue*
add execution_queue users = (*user* [,*user*...]) *queue*

Adds users specified in *user* to the user list of execution queue specified in *queue*.
The mode of the userlist needs to be set by sub-command **set execution_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

add interactive_queue groups = *group queue*
add interactive_queue groups = (*group*[,*group*...]) *queue*

Adds groups specified in *group* to the group list of interactive queue specified in *queue*.
The mode of the group list needs to be set by sub-command **set interactive_queue access | noaccess**.

The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and *group-id* is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

add interactive_queue users = *user queue*
add interactive_queue users = (*user* [,*user*...]) *queue*

Adds users specified in *user* to the user list of interactive queue specified in *queue*.
The mode of the user list needs to be set by sub-command **set interactive_queue access**

| noaccess.

The following two methods are available to specify *user*.

user-name

[*user-id*]

For example, if the *user-name* is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

add routing_queue destination = *destination queue* [*position*]
add routing_queue destination = (*destination*[,*destination*...]) *queue* [*position*]

Destinations specified in *destination* will be added as valid destinations of routing queue specified in *queue*. If there are more than one destination, routing to *destination* is attempted in the order of settings.

Specify in *position* whether to put destinations to be specified from now on before or after destinations that have already been specified in the *queue*.
If not specified, *destinations* are added to the end of destinations already specified in routing queues.

The following items can be specified in *position*.

after *queue_name*

Put a new destination after a destination specified by *queue_name*.

before *queue_name*

Put a new destination before a destination specified by *queue_name*.

first

Put a new destination before all destinations.

last

Put a new destination after all destinations.

Operator privilege is needed.

add routing_queue groups = *group queue*
add routing_queue groups = (*group*[,*group*...]) *queue*

Adds groups specified in *group* to the group list of routing queue specified in *queue*.
The mode of the group list needs to be set by sub-command **set routing_queue access | noaccess**.

The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and *group-id* is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

add routing_queue users = *user queue*
add routing_queue users = (*user* [,*user*...]) *queue*

Adds users specified in *user* to the user list of routing queue specified in *queue*.
The mode of the user list needs to be set by sub-command **set routing_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

attach baremetal_host host = *host_name* **job_server_id** = *jsv_id* **cpu** = *cpunum*
memory = *memsz* **gpu** = *gpunum*
attach baremetal_host file = *file_name*

Registers the baremetal server specified by host_name to Batch server.

jsv_id is the job server number of the job server to execute on the baremetal server. cpunum is CPU number, memsz is memory size, and gpunum is GPU number of the baremetal host.

To register two or more baremetal servers, specify a file that host names, their job server numbers and their resources (CPU number, memory size and GPU number) are written to *file_name*.

For example:

```
#HOSTNAME JSVNO CPU  MEM   GPU
Bhost1      101  4   10GB   0
Bhost2      102  4   20GB   1
```

In the file, lines begin with '#' are treated as comment lines.

Manager privilege is needed.

attach execution_host host = *host_name* **job_server_id** = *jsv_id*
attach execution_host file = *file_name*

Registers the execution host specified by host_name to Batch server.

jsv_id is the job server number of the job server to execute on the execution host.

To register two or more execution hosts, specify a file that host names and their job server numbers are written to *file_name*.

For example:

```
#HOSTNAME                  JSVNO
host_001         1
host_002         2
host_003         3
```

In the file, lines begin with '#' are treated as comment lines.

Manager privilege is needed.

bind execution_queue job_server *queue* **job_server_id** = *n*

Binds the execution queue specified by *queue* to a job server that matches the job server ID specified by *n.*

Manager privilege is needed.

bind execution_queue node_group *queue* **node_group** = *ngrp_name*

Binds the node group specified by *ngrp_name* to the execution queue specified by *queue*.

Manager privilege is needed.

bind execution_queue scheduler *queue* **scheduler_id** = *n*

Binds the execution queue specified by *queue* to a scheduler that matches the scheduler ID specified by *n*.

Manager privilege is needed.

bind interactive_queue job_server *queue* **job_server_id** = *n*

Binds the interactive queue specified by *queue* to a job server that matches the job server ID specified by *n*.

Manager privilege is needed.

bind interactive_queue node_group *queue* **node_group** = *ngrp_name*

Binds the node groupspecified by *ngrp_name* to the interactive queue specified by *queue*.

Manager privilege is needed.

bind interactive_queue scheduler *queue* **scheduler_id** = *n*

Binds the interactive queue specified by *queue* to a scheduler that matches the scheduler ID specified by *n*.

Manager privilege is needed.

create cloud_template = *template_name* **image** = *image*

cpu = *cpunum* **memsz** = *memory_size*

[**gpu =** *gpunum*] [**ve** = *venum*] [**boot_timeout** = *timeout*]

[**stop_timeout** = *timeout*] [**custom** = "*custom_define*"] [**comment** = "*comment*"]

Creates a cloud bursting template in the name specified by *template_name*.

Specify following parameters for the template.

image Image name of the cloud instance to start.

cpu Number of CPUs. Specify the integer of one or more.

memsz Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).

gpu Number of GPUs. Specify the integer of zero or more.
The default value is 0.

ve Number of VEs. Specify the integer of zero or more.
The default value is 0.

boot_timeout Timeout for booting the cloud instance.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 300 seconds.

stop_timeout Timeout for stopping the cloud instance.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 300 seconds.

custom The custom value which is specified for the booting environment.

comment The comment for the template.

Manager privilege is needed.

create container_template = *template_name* **image** = *image*

cpu = *cpunum* **memsz** = *memory_size*

[**gpu =** *gpunum*] [**ve** = *venum*] [**hca =** (*hcanum*)]

[**boot_timeout** = *timeout*] [**stop_timeout** = *timeout*]

[**custom** = "*custom_define*"] [**comment** = "*comment*"]

Creates a container template in the name specified by *template_name*.

Specify following parameters for the template.

image Image name of the container.

cpu Number of CPUs. Specify the integer of one or more.

memsz Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).

gpu Number of GPUs. Specify the integer of zero or more.

ve Number of VEs. Specify the integer of zero or more.

hca Number of HCA ports. The "hcanum" format is follows.
Format: “io”, “mpi”, “all”
Specify the integer of zero or more to "io", "mpi" or "all".
Example) Specify one HCA port for all.
hca = (0,0,1)

boot_timeout Timeout for booting the container.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 900 seconds.

stop_timeout Timeout for stopping the container.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 900 seconds.

custom The custom value which is specified for the booting environment.

comment The comment for the template.

Manager privilege is needed.

create custom_resource = *cr_name* **consumer** = { **job** | **request** }
[**check_mode** = {**off** | **moment** | **integrate**}] [**terminate_**job = {**off** | **on**}]
[**unit** = unit]
[**type =** { **bsv** | **host** } **available** = *n*]

Creates a custom resource in the name specified by *cr_name*.
Specify either unit "job" or "request" for that consume a custom resource in **consumer**.
Specify the mode of resource monitoring "off" or "moment" or "integrate" for **check_mode**.
Default value is "off"
terminate_job is the behavior of termination when the job exceeded the custom resource.
Default value is "off"
unit is the unit of the custom resource. If it is omitted the resource has no unit.
Specify either object "bsv" or "host", that is used for amount control of the custom resource in **type**.
In **available**, the maximum value of the amount of custom resource that can be used at the same time by the object type specified by **type** by the integer of 0-2147483647.

Manager privilege is needed.

create execution_queue = *queue* **priority** = *n*

Creates an execution queue in the name specified by *queue* in queue priority *n* (0 ... 63). "63" defines the highest priority and "0" the lowest priority.

Manager privilege is needed.

create interactive_queue = *queue* **priority** = *n*

Creates an interactive queue in the name specified by *queue* in queue priority *n* (0 ... 63). "63" defines the highest priority and "0" the lowest priority.

Manager privilege is needed.

create network_queue = *queue* **staging_machine** = *staging_machine* **priority** = *queue-priority* [**run_limit** = *n*]

Creates a network queue *queue* whose destination is a host specified by *staging_machine*.

The following two methods can be used to specify *staging_machine*.

machine-name
[*machine-id*]

For example, if the *machine-name* is "machine1" and *machine-id* is "100", specify "machine1" or "[100]".

Network queues must be created for each destination for result file routing. If a network queue matching a destination host cannot be found, a default network queue (DefaultNetQue) is used. More than one network queue can be created with one host. However, only one queue with the highest queue priority is used and other queues will be ignored. One network queue cannot specify more than one host. The destination host name cannot be changed after creating queues.

run_limit specifies the number of requests that can be executed simultaneously by individual network queues.

The default is 1.

Manager privilege is needed.

create node_group = *ngrp_name* [**type** = *ngrp_type*] [**comment** = *comment*]
[**switch_layer** = *layer*] [**template** = *template_name*] [**priority** = *priority*]
[**network** = *network_name*]

Creates a node group in the name specified by *ngrp_name*.
ngrp_type is the type of node group.
The types are as follows:

common
    General type node group (default)
nw_topo
    The node group for showing groups of execution host groups
    under the same switch for scheduling considering network topology.
cloud
    The node group for cloud bursting.

A comment can be set to the node group by specifying *comment*.

Specify the level of the switch layer in *layer*. 1 to 4 can be specified. It can be set only when nw_topo is specified for *ngrp_type*. The default value is 1.

Specify the predefined cloud bursting template in *template_name*. The template shows the specifications of the instance to start. It can be set only when cloud is specified for *ngrp_type*.

Specify the priority of the cloud bursting node group in *priority*. 0 to 63 can be specified. It can be set only when cloud is specified for *ngrp_type*. The default value is 1.

Specify the name of the network in the cloud environment where the cloud instance to be started is located in *network_name*. Specify when you need to distinguish between networks in a cloud environment. You can specify up to 255 characters. It can be set only when cloud is specified for *ngrp_type*.

Manager privilege is needed.

create openstack_template = *template_name*   **image** = *OS_image*

flavor = *flavor_name* **cpu** = *cpunum* **memsz** = *memory_size*
[**gpu =** *gpunum*] [**boot_timeout** = *timeout*] [**stop_timeout** = *timeout*]
[**custom** = "*custom_define*"] [**comment** = "*comment*"]

Creates a OpenStack template in the name specified by *template_name*.

Specify following parameters for the template.

image	OS disk image name which specified in OpenStack.
flavor	Flavor name which specified in OpenStack.
cpu	Number of CPUs. Specify the integer of one or more.
memsz	Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).
gpu	Number of GPUs. Specify the integer of one or more.
boot_timeout	Timeout for booting the VM or Baremetal. The unit is second. Specify integer value of 1 to 2147483647. Default is 900 seconds.
stop_timeout	Timeout for stopping the VM or Baremetal. The unit is second. Specify integer value of 1 to 2147483647. Default is 900 seconds.
custom	The custom value which is specified for the booting environment.
comment	The comment for the template.

Manager privilege is needed.

create routing_queue = *queue* **priority** = *n* [**run_limit** = *n*]
[**destination** = *destination*]
[**destination** = (*destination*[,*destination*...])]

Creates a routing queue in the name specified by *queue* in queue priority *n* (0 ... 63). "63" defines the highest priority and "0" the lowest priority. Specify in destination a list of one or more destinations to be sent by this routing queue.

run_limit specifies the number of requests that can be executed simultaneously by individual routing queues.
The default is 1.

Manager privilege is needed.

delete batch_server group_submit_limit groups = *group*
delete batch_server group_submit_limit groups = (*group*[, *group* ...])

Delete the setting of the number of submitted requests of the group which is specified by group *group*.

Operator privilege is needed.

delete batch_server user_submit_limit users = *user*
delete batch_server user_submit_limit users = (*user*[, *user* ...])

Delete the setting of the number of submitted requests of the user which is specified by user *user*.

Operator privilege is needed.

delete cloud_template = *template_name*

Delete the cloud bursting template which is specified by *template_name*.
The template which is used by any request cannot be deleted.

Manager privilege is needed.

delete container_template = *template_name*

Delete the container template which is specified by *template_name*.
The template which is used by any request cannot be deleted.

Manager privilege is needed.

delete custom_resource = *cr_name*

Delete the custom resource which is specified by *cr_name*.

Manager privilege is needed.

delete openstack_template = *template_name*

Delete the OpenStack template which is specified by *template_name*.
The template which is used by any request cannot be deleted.

Manager privilege is needed.

delete execution_queue groups = *group queue*
delete execution_queue groups = (*group*[,*group*...]) *queue*

Deletes groups specified in *group* to the group list of execution queue specified in *queue*. The mode of the group list needs to be set by sub-command **set execution_queue access | noaccess**.

The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and *group-id* is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

delete execution_queue group_submit_limit groups = *group* *queue*
delete execution_queue group_submit_limit groups = (*group*[,*group* ...]) *queue*

Delete the setting of the number of submitted requests of the group *group*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete execution_queue hold_privilege *queue*

Reset the privilege that can hold request of execution queue specified by *queue*.

Operator privilege is needed.

delete execution_queue jobs_range groups = *group* *queue*
delete execution_queue jobs_range groups = (*group*[,*group* ...]) *queue*
delete execution_queue jobs_range users = *user* *queue*

delete execution_queue jobs_range users = (*user*[,*user*...]) *queue*

Delete the upper limit and lower limit of the number of request's jobs which is specified to the group *group* or user *user*. These limits are specified to the queue *queue*.

Operator privilege is needed.

delete execution_queue per_req elapse_time_limit groups = *group* *queue*
delete execution_queue per_req elapse_time_limit groups = (*group*[,*group* ...]) *queue*
delete execution_queue per_req elapse_time_limit users = *user* *queue*
delete execution_queue per_req elapse_time_limit users = (*user*[,*user*...]) *queue*

Delete the setting of the limit of the elapse time of request which is specified to the group *group* or user *user*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete execution_queue queue = *queue*

Deletes execution queue specified by *queue*. Execution queues can be deleted if there are no requests in them and the execution queues are in a disabled state.
(See **disable execution_queue**.)

Manager privilege is needed.

delete execution_queue refuse_submission = (*from*[,*from*...]) *queue*

Deletes the permitted route *from* of submitting request in the execution queue *queue*.

The following can be specified in *from*

qsub | submit	Accept to submit via qsub
qmove	Accept to submit via qmove
local_routing	Accept to routing via local routing queue
remote_routing	Accept to routing via remote routing queue

Operator privilege is needed.

delete execution_queue suspend_privilege *queue*

Reset the privilege for suspending requests of execution queue specified by *queue*.

Operator privilege is needed.

delete execution_queue userexit location = *location* **queue** = *queue*

Delete the script executed in the state specified by *location* in execution queue specified by *queue*.

The following can be specified in *location*.

pre-running	At PRE-RUNNING status
post-running	At POST-RUNNING status

Operator privilege is needed.

delete execution_queue users = *user queue*
delete execution_queue users = (*user*[,*user* ...]) *queue*

Deletes users specified in *user* to the user list of execution queue specified in *queue*.
The mode of the user list needs to be set by sub-command **set execution_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

delete execution_queue user_submit_limit users = *user queue*
delete execution_queue user_submit_limit users = (*user*[,*user* ...]) *queue*

Delete the upper limit of the number of submitted request which is specified to the user *user*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete interactive_queue groups = *group queue*
delete interactive_queue groups = (*group*[,*group*...]) *queue*

Deletes groups specified in *group* to the group list of interactive queue specified in *queue*.
The mode of the group list needs to be set by sub-command **set interactive_queue access | noaccess**.

The following two methods are available to specify *group*.
group-name
[*group-id*]

For example, if the *group-name* is "group1" and group-id is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

delete interactive_queue group_submit_limit groups = *group queue*
delete interactive_queue group_submit_limit groups = (*group*[,*group* ...]) *queue*

Delete the upper limit of the number of submitted request which is specified to the group *group*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete interactive_queue jobs_range groups = *group queue*
delete interactive_queue jobs_range groups = (*group*[,*group* ...]) *queue*
delete interactive_queue jobs_range users = *user queue*
delete interactive_queue jobs_range users = (*user*[, *user*...]) *queue*

Delete the upper limit and lower limit of the number of request's jobs which is specified to the group *group* or user *user*. These limits are specified to the queue *queue*.

Operator privilege is needed.

delete interactive_queue per_req elapse_time_limit groups = *group* *queue*
delete interactive_queue per_req elapse_time_limit groups = (*group*[,*group* ...]) *queue*
delete interactive_queue per_req elapse_time_limit users = *user queue*
delete interactive_queue per_req elapse_time_limit users = (*user*[,*user*...]) *queue*

Delete the setting of the limit of the elapse time of request which is specified to the group *group* or user *user*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete interactive_queue queue = *queue*

Deletes interactive queue specified by *queue*. Interactive queues can be deleted if there are no requests in them and the interactive queues are in a disabled state. (See **disable interactive_queue**.)

Manager privilege is needed.

delete interactive_queue refuse_submission = (*from*[,*from*...]) *queue*

Deletes the permitted route *from* of submitting request in the interactive queue *queue*.

The following can be specified in *from*

qsub ｜ submit	Accept to submit via qlogin or qrsh

Operator privilege is needed.

delete interactive_queue restrict_shell *queue*

Deletes the setting of restrict shell of the interactive queue specified by *queue*.

Operator privilege is needed.

delete interactive_queue suspend_privilege *queue*

Reset the privilege for suspending requests of interactive queue specified by *queue*.

Operator privilege is needed.

delete interactive_queue userexit location = *location* **queue** = *queue*

Delete the script executed in the state specified by *location* in interactive queue specified by *queue*.

The following can be specified in *location*.

pre-running	At PRE-RUNNING status
post-running	At POST-RUNNING status

Operator privilege is needed.

delete interactive_queue users = user *queue*
delete interactive_queue users = (*user* [,*user*...]) *queue*

Deletes users specified in *user* to the user list of interactive queue specified in *queue*.
The mode of the user list needs to be set by sub-command **set interactive_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the user-name is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

delete interactive_queue user_submit_limit users = *user* *queue*
delete interactive_queue user_submit_limit users = (*user*[,*user* ...]) *queue*

Delete the upper limit of the number of submitted request which is specified to the user *user*. This limit is specified to the queue *queue*.

Operator privilege is needed.

delete network_queue queue = *queue*

Deletes network queue specified by *queue*. Network queues can be deleted if there are no requests in them and the network queues are in a disabled state. (See **disable network_queue**.)

The Default Network Queue (DefaultNetQue) cannot be disabled by this sub-command.

Manager privilege is needed.

delete node_group = *ngrp_name*

Deletes a node group whose name is *ngrp_name*. To delete a node group, the node group is required to unbind from all queues.

Manager privilege is needed.

delete routing_queue destination = destination *queue*
delete routing_queue destination = (*destination*[,*destination*...]) *queue*

Deletes destination queues specified by *destination* from the destination queue list of routing queue specified by *queue*. However, requests currently routed to the destination queue to be deleted from the specified queue are guaranteed for proper routing. In case all destination queues of routing queues are deleted by this method, the routing queues will remain in a state similar to a stop state.

Operator privilege is needed.

delete routing_queue groups = group *queue*
delete routing_queue groups = (*group*[,*group*...]) *queue*

Deletes groups specified in *group* to the group list of routing queue specified in *queue*.
The mode of the group list needs to be set by sub-command **set routing_queue access | noaccess**.

The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the group-name is "group1" and *group-id* is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

delete routing_queue group_submit_limit groups = *group* *queue*
delete routing_queue group_submit_limit groups = (*group*[,*group* ...]) *queue*

Delete the setting of the number of submitted requests of the group which is specified by group *group*.

Operator privilege is needed.

delete routing_queue queue = *queue*

Deletes routing queue specified by *queue*. Routing queues can be deleted if there are no requests in them and the routing queues are in a disabled state.
(See **disable routing_queue**.)

Manager privilege is needed.

delete routing_queue refuse_submission = (*from*[,*from*...]) *queue*

Deletes the permitted route *from* of submitting request in the routing queue *queue*.

The following can be specified in *from*

qsub | submit   Accept to submit via qsub

qmove	Accept to submit via qmove
local_routing	Accept to routing via local routing queue
remote_routing	Accept to routing via remote routing queue

Operator privilege is needed.

delete routing_queue users = *user queue*
delete routing_queue users = (*user* [,*user*...]) *queue*

Deletes users specified in *user* from the user list of routing queue specified in *queue*.
The mode of the user list needs to be set by sub-command **set routing_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and *user-id* is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

delete routing_queue user_submit_limit users = *user queue*
delete routing_queue user_submit_limit users = (*user*[,*user* ...]) *queue*

Delete the setting of the number of submitted requests of the user which is specified by user *user*.

Operator privilege is needed.

detach baremetal_host host = *host_name*
detach baremetal_host host = (*host_name1* , *host_name2* , ...)
detach baremetal_host job_server_id = *jsv_id*
detach baremetal_host job_server_id = (*jsv_id1*, *jsv_id2*, ...)
detach baremetal_host job_server_id = *jsv_id1-jsv_id2*
detach baremetal_host all

Removes the registration of the baremetal server from the batch server.

To remove registration, it is required that the job server on the baremetal server has no jobs and the job server is linked down.

There are three ways to specify baremetal server to remove registration as follows:

- Specify host name by *host_name*.
- Specify job server number of the baremetal server by *jsv_id*.
- Specify all to remove registrations of all baremetal servers.

To remove two or more registrations, *host_names* or jsv_ids can be specified in parentheses delimited by comma. And job server numbers can be specified as *jsv_id1-jsv_id2* if the numbers are sequential.

Manager privilege is needed.

detach execution_host host = *host_name*
detach execution_host host = (*host_name1*, *host_name2*,...)
detach execution_host job_server_id = *jsv_id*
detach execution_host job_server_id = (*jsv_id1*,*jsv_id2*,...)
detach execution_host job_server_id = *jsv_id1-jsv_id2*
detach execution_host all

Removes the registration of the execution host from the batch server. To remove registration, it is required that the job server on the execution host has no jobs and the job server is linked down.

There are three ways to specify execution host to remove registration as follows:

- Specify host name by *host_name*.
- Specify job server number of the execution host by *jsv_id*.
- Specify all to remove registrations of all execution hosts.

To remove two or more registrations, *host_names* or jsv_ids can be specified in parentheses delimited by comma. And job server numbers can be specified as *jsv_id1-jsv_id2* if the numbers are sequential.

Manager privilege is needed.

disable execution_queue = *queue*

Disables execution queue specified by *queue*. New requests cannot be input to disabled execution queue.

Operator privilege is needed.

disable interactive_queue = *queue*

Disables interactive queue specified by *queue*. New requests cannot be input to disabled interactive queue.

Operator privilege is needed.

disable network_queue = *queue*

Disables network queue specified by *queue*. New requests cannot be input to disabled network queue.

The Default Network Queue (DefaultNetQue) cannot be deleted by this sub-command.

Operator privilege is needed.

disable routing_queue = *queue*

Disables routing queue specified by *queue*. New requests cannot be input to disabled routing queue.

Operator privilege is needed.

edit custom_resource add type = **bsv** **available** = *n* *cr_name*
edit custom_resource add type = **host** [**target** = *host_name*]
available = *n* *cr_name*

Adds consume amount control information for **type** and **target** to the custom resource specified by *cr_name.*

Following value can be specified for **type**. It is the type of target object of consume amount control.

bsv	Set BSV for the target object of consume amount control. It is scheduled to control the amount of resources in the whole BSV.

host Set execution host for the target object of consume amount control. It is scheduled to control the amount of resources in the whole one execution host. If the consumer is set to "request", host cannot be specified for **type**. The individual host name can be specified in *host_name.*

In **available**, the maximum value of the amount of custom resource that can be used at the same time by the object type specified by **type** and **target** by the integer of 0-2147483647.

It is overwritten in a new value.

Manager privilege is needed.

edit custom_resource delete type = **bsv** *cr_name*

edit custom_resource delete type = **host** [**target** = *host_name*] *cr_name*

Delete consume amount control information for **type** and **target** from the custom resource specified by *cr_name.*

Following value can be specified for **type**. It is the type of target object of consume amount control.

bsv Delete the BSV target object of consume amount control.

host Delete the host target object of consume amount control. If the **target** is not specified, all target object in type host is deleted. If the target is specified, only the host which is specified by *host_name* is deleted.

Manager privilege is needed.

edit node_group add job_server_id = *n ngrp_name*

edit node_group add job_server_id = (*n*, *m*, ...) *ngrp_name*

edit node_group add job_server_id = *n-m ngrp_name*

edit node_group add node_group = *src_ngrp ngrp_name*

edit node_group add node_group = (*src_ngrp1*,*src_ngrp2*,...) *ngrp_name*

Adds jobservers to the node group specified by *ngrp_name*.

The ways to specify the jobservers to be added to the node group is as follows:

- Specify job server number by *n.*
- Specify two or more job server numbers in parentheses delimited by comma.
- Specify a range of job server numbers as *n-m.*

- Specify node group name to add all jobservers in the node group.
- Specify two or more node group names in parentheses delimited by comma.

Job servers that are added to cloud-type node groups cannot belong to other node groups. Operator privilege is needed.

edit node_group add template = *template_name ngrp_name*
edit node_group add template = (*template_name1*,*template_name2*,...) *ngrp_name*

Adds cloud bursting templates to the cloud bursting node group specified by *ngrp_name*. If you want to add multiple templates at once, specify them by separating them with commas, such as *template_name1*, *template_name2*. You can add up to 20 templates to a cloud bursting node group.

Operator privilege is needed.

edit node_group delete job_server_id = *n ngrp_name*
edit node_group delete job_server_id = (*n*, *m*, ...) *ngrp_name*
edit node_group delete job_server_id = *n-m ngrp_name*
edit node_group delete node_group = *src_ngrp ngrp_name*
edit node_group delete node_group = (*src_ngrp1*,*src_ngrp2*,...) *ngrp_name*

Removes jobservers from the node group specified by *ngrp_name*.

The ways to specify the jobservers to be removed from the node group is as follows:

- Specify job server number by *n*.
- Specify two or more job server numbers in parentheses delimited by comma.
- Specify a range of job server numbers as *n-m*.
- Specify node group name to add all jobservers in the node group.
- Specify two or more node group names in parentheses delimited by comma.

Operator privilege is needed.

edit node_group delete template = *template_name ngrp_name*
edit node_group delete template = (*template_name1*,*template_name2*,...) *ngrp_name*

Removes cloud bursting templates from the cloud bursting node group specified by *ngrp_name*. If you want to remove multiple templates at once, specify them by separating them with commas, such as *template_name1*,*template_name2*.

Operator privilege is needed.

enable execution_queue = *queue*

Puts an execution queue specified by *queue* into an enabled state. Enabling an execution queue will allow new requests to be submitted to this execution queue.

Operator privilege is needed.

enable interactive_queue = *queue*

Puts an execution queue specified by *queue* into an enabled state. Enabling an interactive queue will allow new requests to be submitted to this interactive queue.

Operator privilege is needed.

enable network_queue = *queue*

Puts a network queue specified by *queue* into an enabled state. Enabling a network queue will allow new requests to be submitted to this network queue.

Operator privilege is needed.

enable routing_queue = *queue*

Puts a routing queue specified by *queue* into an enabled state. Enabling a routing queue will allow new requests to be submitted to this routing queue.

Operator privilege is needed.

exit

Exits Command *qmgr*.

help [*command*]

Gets help information. Using sub-command help without an argument will display information on commands that can be used. By using sub-command help bundled with an argument, information on specified commands will be displayed.
Commands can be specified partially as long as commands are unique.
Specification of a command argument in more detail will yield more detailed information that will be received.

hold job_server *job_server_id*

Holds all of requests running on the job server specified by *job_server_id*.
(requests with the state of RUNNING) Note that the requests that is set to Disable hold will not be held.

The owner of the request or Operator privilege is needed.

list all [**file**=*file_name*]

Saves information related to the NQSV batch server, execution hosts (job servers), node groups, queues, custom resources and templates currently in operation into a file specified by *file_name* after converting it into the form of sub-command of qmgr. But it doesn't output a sub-command about the bind command.
The NQSV environment can be restored easily by reading the created file in qmgr.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

list batch_server [**file**=*file_name*]

Saves information related to the NQSV batch server currently in operation in a file specified by *file_name* after converting it into the form of sub-command of qmgr. The NQSV environment can be restored easily by reading the created file in qmgr.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

list bind [**file**=*file_name*]

Saves bind information of between NQSV queues and job servers currently in operation in a file specified by *file_name* after converting it into the form of sub-command of qmgr. The NQSV environment can be restored easily by reading the created file in qmgr.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

list custom_resource [**file** = *file_name*]

Saves information related to the custom resources currently in operation in a file specified by *file_name* after converting it into the form of sub-command of qmgr.

list node [**file**=*file_name*]

Saves information of NQSV execution hosts (jobservers) and node groups currently in operation after converting it into the form of sub-command of qmgr. The NQSV environment can be restored easily by reading the created file in qmgr.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

list queue [**file**=*file_name*]

Saves information related to the NQSV queues currently in operation in a file specified by *file_name* after converting it into the form of sub-command of qmgr. But it doesn't output a sub-command about the bind command. The NQSV environment can be restored easily by reading the created file in *qmgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

list template [**file**=*file_name*]

Saves information of NQSV template currently in operation after converting it into the form of sub-command of qmgr.
The NQSV environment can be restored easily by reading the created file in qmgr.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be overwritten forcibly.

load nqsd_conf

Reloads the nqsd.conf file.

Manager privilege is needed.

lock cloud_template = t*emplate_name*

Lock the cloud bursting template specified by *template_name*.

Manager privilege is needed.

lock container_template = t*emplate_name*

Lock the container template specified by *template_name*.

Manager privilege is needed.

lock node_group = *ngrp_name*

Lock the cloud bursting node group specified by *ngrp_name*.

Manager privilege is needed.

lock openstack_template = t*emplate_name*

Lock OpenStack template specified by *template_name*.

Manager privilege is needed.

migrate job = *job_id* **job_server_id**=*job_server_id*

Migrates jobs specified by *job_id* to the job server specified by *job_server_id* by the job migration function. (Only requests in a HELD state can be moved.)

Operator privilege is needed.

purge execution_queue = *queue*

Deletes all queued requests (All QUEUED, WAITING, HELD and STAGING requests) in specified execution queue.

The owner of the request or manager privilege is needed.

purge interactive_queue = *queue*

Deletes all queued requests (All QUEUED,WAITING and STAGING requests) in specified interactive queue.

The owner of the request or manager privilege is needed.

purge routing_queue = *queue*

Deletes all queued requests (All QUEUED,WAITING and HELD requests) in specified routing queue.

The owner of the request or manager privilege is needed.

reset baremetal_host host = *host_name*

Clear the template information of baremetal server specified by *host_name* which is LINKDOWN by hardware trouble etc.
This command can be executed when there is no job on the host.

Manager privilege is needed.

set batch_server acct_server [**hostname** = *host_name*] [**port** = *port_number*]

Set a server which is used at checking of excess the budget limitation.
Set a host name of accounting server of ASV to *host_name*.
Set a TCP port number for server's listen port of a request of excess the budget limitation to *port_number*.

Operator privilege is needed.

set batch_server allow_absolute_exepath { **on** | **off** }

Set whether to automatically delete connected requests that failed to be submitted.

on Delete automatically connected requests to fail to submit. (Default)

off Not delete automatically connected requests to fail to submit.
Those requests remain in HELD state.

Manager privilege is needed.

set batch_server auto_delete_failed_request { **on** | **off** }

Set the batch server to allow or not of specifying absolute path for staging file with external staging.

on Allow to specify the absolute path for staging file.

off Refuse to specify the absolute path for staging file. (Default)

Manager privilege is needed.

set batch_server nqs_budget_chk = **on** [**target** = *target*]
set batch_server nqs_budget_chk = **off**

Set whether excess of the budget limitation is checked or not when using budget control by ASV.

on Excess of the budget limitation is checked.
Optionally, target to be checked for budget overruns can be specified after the "target =" in parentheses.
The *target* can be as follows.

request
Only requests are checked for the budget overruns.

reservation
Only reservations are checked for the budget overruns.

request,reservation
Both requests and reservations are checked for the budget overruns. (default)

off Excess of the budget limitation isn't checked. (default)

Manager privilege is needed.

set batch_server get_resource_interval = *interval*

Set an interval to get resource quantity of jobs.
Intervals can be set in seconds. "0" will stop getting of resources.

Operator privilege is needed.

set batch_server group_submit_limit = *limit*

Set the number of requests which one group can submit to the system.
The number of requests that can be submitted is a total of existing requests in the system at the same time.

If this sub-command is used, one group can submit requests only in a number specified by *limit*. An error message will be output if requests in excess of the number specified by *limit* are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

This limit is applied to requests of the group which an upper limit by individually specifying group name is not set to.

Operator privilege is needed.

set batch_server group_submit_limit = *limit*   **groups** = *group*
set batch_server group_submit_limit = *limit*   **groups** = (*group*[, *group* ...])

Set the number of batch requests which the group designated in *group* can submit at one time. Specify group name to *group*. An error message will be output if requests of group in excess of the number specified by limit are submitted at one time. "0" or "unlimited" indicates unlimited.

Operator privilege is needed.

set batch_server heartbeat_interval = *interval*

Set an interval to send heartbeat between batch servers and job servers.
Intervals can be set in seconds. "0" will stop heartbeat sending.
Operator privilege is needed.

set batch_server jacct_dir = *dir_name*

Specify the directory name on the batch server host which outputs the job account files into *dir_name* when using budget control by ASV. The default is /var/opt/nec/nqsv/acm/jacct.

Manager privilege is needed.

set batch_server load_interval = *interval*

Set an interval to get load information of execution hosts and HW resource loading information. Intervals can be set in seconds. "0" will stop getting load information and HW resource loading information.

Operator privilege is needed.

set batch_server log_file [**file**=*file_name*] [**size**=(*file_size*)] [**level**=*log_level*] [**save**=*save_num*]

Set NQSV log files.
Set a log file name in *file_name*. Set a high-limit size of log files in *file_size*. Values in 1024 -2^31-1 bytes can be specified. A unit can be specified in size specification. If a unit is not specified, the unit will be byte. "0" indicates unlimited. Reaching the high-limit value for size, a file will automatically be reset and the previous log file will be saved in the backup file.
Set the output level of log files in *log_level*.
Set the number of backup files to be saved in *save_num*.

DEBUG messages output at each level are as follows.
The DEBUG message of specified level and below will be output. (In case of Level 2, Level 1 and Level 2 messages will be output.)
In normal operation, Level 1 or Level 2 is recommended.

Level 0	No DEBUG message is output.
Level 1	Processing status of start/stop of batch server Connecting status of batch scheduler Connecting status of job server Processing status of batch request Exit status of master job Processing status of network request
Level 2	Processing status of jobs Connection status of　NQSV/API client Exit status of all jobs Standard/error output of user EXIT script Standard/error output of staging script
Level 3	Processing status of job controlling packet (task packet)
Level 4	Processing summary of communication packet
Level 5	Displaying the content of communication packet

Operator privilege is needed.

set batch_server network_queue retry_interval = *interval*

Set a waiting interval for next routing retry by a network queue in seconds.

Operator privilege is needed.

set batch_server network_queue retry_span = *span*

Set a span for retry by a network queue. Values in 1 to 259200 seconds (3 days) can be specified to *span* with unit of second.
And the default value is 259200 seconds.

Operator privilege is needed.

set batch_server network_queue run_limit = *limit*

Set the value specified in *limit* as the number of requests that can be executed simultaneously on network queues in the entire system. Execution of routing will be stopped if the number for the entire system reaches the value set by this command even though the run limit of individual network queues have a margin. In case a value larger than the total of run limit of each network queues in the system is specified in limit, this setting will become totally meaningless.

Operator privilege is needed.

set batch_server req_account { **on** | **off** }

Start and stop output of the request account information and
the job account information.

on	Start output
off	Stop output

Manager privilege is needed.

set batch_server req_account_file = *file_name*

Specify the complete path on the batch server host which outputs the request account file for *file_name*.

Manager privilege is needed.

set batch_server reservation_acct = { **on** | **off** }

Set charge for Resource Reservation of JobManipulator or charge for request which is submitted in Resource Reservation when using budget control by ASV.

on	charge for Resource Reservation of JobManipulator
off	charge for request which is submitted in Resource Reservation.

Manager privilege is needed.

set batch_server routing_queue retry_interval = *interval*

Set a waiting interval for next routing retry by a routing queue in seconds.

Operator privilege is needed.

set batch_server routing_queue retry_span = *span*

Set a span for retry by a routing queue. Values in 1 to 259200 seconds (3 days) can be specified to *span* with unit of second. And the default value is 259200 seconds.

Operator privilege is needed.

set batch_server routing_queue run_limit = *limit*

Set the value specified in *limit* as the number of requests that can be executed simultaneously on routing queues in the entire system. Execution of routing will be stopped if the number for the entire system reaches the value set by this command even though the run limit of individual routing queues has a margin. In case a value larger than the total of run limits of each routing queues in the system is specified in limit, this setting will become totally meaningless.

Operator privilege is needed.

set batch_server rsvacct_file = *path_name*

Specify the complete path on the batch server host which outputs the reserve account file for *path_name*.

Manager privilege is needed.

set batch_server specify_group_request { **on** | **off** }

Setting of whether Designated Group Execution Function of request is used or not.

on The execution function of the group specifying of the request is used. You can specify secondarily group when you submit a request and jobs are executed under the group.

off The execution function of the group specifying of the request isn't used.

Operator privilege is needed.

set batch_server submit_limit = *limit*

Set the number of requests which can be submitted to the system.
The number of requests that can be submitted is a total of existing requests in the system at the same time.

An error message will be output if requests in excess of the number specified by *limit* are submitted.
A value ranging from 1 to 1000000 can be specified in *limit*. The default value is 1000.

Operator privilege is needed.

set batch_server subrequest_entry_limit = *limit*

Sets the maximum number of sub-requests of each parametric request which are created simultaneously in batch server. The default value is 100.

Operator privilege is needed.

set batch_server user_submit_limit = *limit*

Set the number of requests which one user can submit to the system.
The number of requests that can be submitted is a total of existing requests in the system at the same time.

If this sub-command is used, one user can submit requests only in a number specified by limit. An error message will be output if requests in excess of the number specified by *limit* are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

This setting is applied to a request which isn't set limitation with individual username.

Operator privilege is needed.

set batch_server user_submit_limit = *limit* **users** = *user*
set batch_server user_submit_limit = *limit* **users** = (*user*[, *user* ...])

Set the number of batch requests which the user designated in *user* can submit at one time. Specify user name to *user*. An error message will be output if requests of user *user* in excess of the number specified by limit are submitted at one time.
"0" or "unlimited" indicates unlimited.

Operator privilege is needed.

set cloud_template image = *image* *template_name*
set cloud_template cpu = *cpunum* *template_name*
set cloud_template memsz = *memory_size* *template_name*
set cloud_template gpu = *gpunum* *template_name*
set cloud_template ve = *venum* *template_name*
set cloud_template boot_timeout = *timeout* *template_name*
set cloud_template stop_timeout = *timeout* *template_name*
set cloud_template custom = "*custom_define*" *template_name*
set cloud_template comment = "*comment*" *template_name*

Set the parameters of the cloud bursting template specified by *template_name.*
The template which is used by any request cannot be set.

Specify following parameters for the template.

image	Image name of the cloud instance to start.
cpu	Number of CPUs. Specify the integer of one or more.
memsz	Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).
gpu	Number of GPUs. Specify the integer of one or more.
ve	Number of VEs. Specify the integer of one or more.
boot_timeout	Timeout for booting the cloud instance. The unit is second. Specify integer value of 1 to 2147483647. Default is 300 seconds.

stop_timeout Timeout for stopping the cloud instance.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 300 seconds.

custom The custom value which is specified for the booting environment.

comment The comment for the template.

Manager privilege is needed.

set container_template image = *image* *template_name*
set container_template cpu = *cpunum* *template_name*
set container_template memsz = *memory_size* *template_name*
set container_template gpu = *gpunum* *template_name*
set container_template ve = *venum* *template_name*
set container_template hca = (*hcanum*) *template_name*
set container_template boot_timeout *= timeout* *template_name*
set container_template stop_timeout *= timeout* *template_name*
set container_template custom = "*custom_define*" *template_name*
set container_template comment = "*comment*" *template_name*

Set the parameters of the container template specified by *template_name.*
The template which is used by any request cannot be set.

Specify following parameters for the template.

image Image name of the container.

cpu Number of CPUs. Specify the integer of one or more.

memsz Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).

gpu Number of GPUs. Specify the integer of one or more.

ve Number of VEs. Specify the integer of one or more.

hca Number of HCA ports. Number of HCA ports. The "hcanum" format is follows.
Format: “io”, “mpi”, “all”
Specify the integer of zero or more to "io", "mpi" or "all".
Example) Specify one HCA port for all.

hca = (0,0,1)**boot_timeout** Timeout for booting the container.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 900 seconds.

stop_timeout Timeout for stopping the container.

The unit is second. Specify integer value of 1 to 2147483647.

Default is 900 seconds.

custom The custom value which is specified for the booting environment.

comment The comment for the template.

Manager privilege is needed.

set custom_resource consumer = { **job** | **request** } *cr_name*

Change consume unit of custom resource specified by *cr_name.*

It cannot be changed from "job" to "request", if host type consume amount control information.

Manager privilege is needed.

set custom_resource check_mode = { **off** | **moment** | **integrate** } *cr_name*

off Resource check and report is turned off.

moment Resource check and report is turned on. The Resource usage is treated as momentary value.

integrate Resource check and report is turned on. The Resource usage is treated as accumulated value.

Please refer to the "Accounting" for details of "momentary value" and "accumulated value" If terminate_job mode is already set to on, this mode cannot be turned off. Please turn the terminate_job mode off in advance in this case.

Manager privilege is required.

set custom_resource terminate_job = { **off** | **on** } *cr_name*

Changes the job termination mode in the case of excessive usage of custom resource specified by cr_name to off or on.

Job is not terminated even after exceed the custom resource usage.

Job is terminated, when exceed the custom resource usage.

If the check mode is already set to off, this mode cannot be turned on. Please turn the check mode to off in advance in this case.

Manager privilege is required.

set custom_resource unit = *unit* *cr_name*

Sets the unit of custom resource specified by *cr_name.* The unit should be a string with no more than 5 characters. If the string has more than 5 characters, it is trimmed off to 5 characters. If *unspecified* is given as an unit, the existing unit is canceled and treated as unit less number.

Manager privilege is required.

set execution_queue access *queue*

Enables users in the user list and groups in the group list to access execution queue specified in *queue*. If the mode is altered, the content of the list except "root" till that time will be deleted entirely.

Manager privilege is needed.

set execution_queue auto_bind_jobserver { **on** | **off** } *queue*

Set bind automatically or not the execution queue specified by *queue* to a job server when a job server starts. In case of **on**, a job server is bound automatically. In case of **off**, a job server is not bound automatically.

Operator privilege is needed.

set execution_queue cpunum_per_gpunum = *cpunum queue*

Set the default number of CPU cores per GPU for requests submitted to the GPU-CPU Affinity feature-enabled execution-queue specified in *queue*. An integer from 1 to 2147483647 can be specified for *cpunum*. The default value is 1. The value specified for *cpunum* is the value calculated by the following formula.
Number of CPU cores in 1 socket / Number of GPUs physical close to the socket

Operator privilege is needed.

set execution_queue custom_resource = *cr_name* **standard** = *std* *queue*
set execution_queue custom_resource = *cr_name* **range** = (*min,max*) *queue*
set execution_queue custom_resource = *cr_name* **permit_unused** = { **yes** | **no** } qu*eue*

Set the limit of custom resource specified by *cr_name* in the queue which is specified by *queue.*

standard is the standard value which is applied when the request is submitted. The integer value of 1 to 2147483647 or unused (0) can be specified for *std*.

range is the value range (lower limit and upper limit) of custom resource usage which can be specified when the request submitted. The integer value of 1 to 2147483647 can be specified for *min* and *max* .

permit_unused is the flag to allow or refuse to specify unused (0) value when submitting request.

yes	Allow to specify unused (0) when submitting request (Default).
no	Refuse to specify unused (0) when submitting request.

Operator privilege is needed.

set execution_queue defined_ve_number = *venum queue*

Specify default number of VE for automatic calculation of job number to the *queue* at submitting of a request which is specified total number of VE nodes.A value ranging from 1 to 256 can be specified in *venum*. The default is 1.

Operator privilege is required.

set execution_queue delete_failed_urgent_request = { **on** | **off** } *queue*

Specify the function to delete the urgent request that is submitted into the queue if it cannot execute. If on is designated, the urgent request that cannot execute is deleted. If off is designated, the urgent request that cannot execute isn't deleted. Then the interrupted normal request is rerun. The default is off.

Operator privilege is required.

set execution_queue exclusive_submit { ***on*** | ***off*** } *queue*

Specify the allowance for submitting exclusive execution request
on allows the submission of the exclusive execution request. **off** rejects the submission of the exclusive execution request.

Manager privilege is needed.

set execution_queue file_stageout = { **on** | **off** } *<queue>*

Specify whether or not to perform stage-out for the batch queue.
on allows the stage-out. **off** rejects the file staging out.

Operator privilege is needed.

set execution_queue gpu_affinity = { **on** | **off** } *<queue>*

Specify whether to use the GPU-CPU Affinity feature in the request submitted to the execution queue specified in *queue*. If **on** is specified, use the GPU-CPU Affinity feature. If **off** is specified, the GPU-CPU Affinity feature is not used. If you want to use the GPU-CPU Affinity feature, you need to enable Socket Scheduling feature. The default value is **off**.

Operator privilege is needed.

set execution_queue group_submit_limit = *limit queue*

Set the number of batch requests which one group can submit to an execution queue.
The number of batch requests that can be submitted means the total number of batch requests existent in an execution queue at the same time.

If this sub-command is used, one group can submit requests to each execution queue only in a number specified by *limit*. An error message will be output if requests in excess of the number specified by limit are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of execution queue to be set this limit by *queue.*

This setting is applied to a request which isn't set limitation with individual group name.

Operator privilege is needed.

set execution_queue group_submit_limit = *limit* **groups** = *group* *queue*
set execution_queue group_submit_limit = *limit* **groups** = (*group*[, *group* ...]) *queue*

Set the number of batch requests which the group designated in *group* can submit at one time. Specify group name to *group*. An error message will be output if requests of group *group* in excess of the number specified by limit are submitted at one time. "0" or "unlimited" indicates unlimited.

Operator privilege is needed.

set execution_queue groups = *group queue*

set execution_queue groups =(*group*[,*group*...]) *queue*

Set groups specified by group in the group list of execution queue specified by *queue*. The previous group list information will be deleted. The mode of the group list needs to be set by sub-command **set execution_queue access |noaccess**.

The following two methods are available to specify *group*.

group-name

[*group-id*]

For example, if the *group-name* is "group1" and group-id is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

set execution_queue hca_number_range = (*min*, *max*) mode=*hca-mode queue*

Specify the range of the limit (lower limit, upper limit) on the number of HCA port to the queue which is specified by *queue*. Integer of more than or equal to 0 and less than or equal to 32 or "unlimited" can be specified to min and max. Min have to be less than or equal to max. If min and max are equal, only such value can be specified at submitting of a request.

Following value can be specified for *hca-mode*.

io	HCA for ScaTeFS direct IO
mpi	HCA for MPI
all	Both for ScaTeFS direct IO and MPI

This value limits the HCA port number which specified for the request which submitted to the *queue*.

The default value is "(0, unlimited)". The limit for each hca-mode is independent and each value is not affects to the others.

Operator privilege is needed.

set execution_queue hold_privilege = *privilege queue*

Set the privilege that can hold request of execution queue specified by *queue*.

The following can be specified in *privilege*.

user 　The privilege to hold is more than User privilege.
specialuser 　The privilege to hold is more than Special user privilege.
groupmanager 　The privilege to hold is more than Group Manager privilege.
operatorThe privilege to hold is more than Operator privilege.
manager 　The privilege to hold is more than Manager privilege.
scheduler 　The privilege to hold is more than Scheduler privilege.

Operator privilege is needed.

set execution_queue hook_function { **on** | **off** } *queue*

Set using or not hook script function to the execution queue specified by *queue*.
The default is **off.**

on 　Use hook script function.
off 　Do not use hook script function (Default).

Operator privilege is needed.

set execution_queue intmpi_process_manager = hydra *queue*

Set **hydra** to the execution queue specified by *queue* as a process manager used by IntelMPI request.
The default is **hydra**.

Operator privilege is needed.

set execution_queue jobs_range = (*jobs_min*, *jobs_max*) *queue*

Set the range of number of jobs per a batch request to the execution queue specified by *queue*. A value ranging from 1 to 10240 can be specified in *jobs_min*, *jobs_max*. An error message will be output if a value is not in this range.

The following can be specified for each element of job_range.

jobs_min	The low limit number of jobs per a batch request
jobs_max	The high limit number of jobs per a batch request

Operator privilege is needed.

set execution_queue jobs_range = (*jobs_min* , *jobs_max*) **groups** = *group* *queue*
set execution_queue jobs_range = (*jobs_min* , *jobs_max*) **groups** = (*group*[,*group* ...]) *queue*
set execution_queue jobs_range = (*jobs_min* , *jobs_max*) **users** = *user* *queue*
set execution_queue jobs_range = (*jobs_min* , *jobs_max*) **users** = (*user*[, *user*...]) *queue*

Set the range every user *user* or group *group* of jobs per batch request to the execution queue specified by *queue*. Specify group name to *group* and/or user name to *user*. A value ranging from 1 to 10240 can be specified in jobs_min,
jobs_max. An error message will be output if a value is not in this range.

The following can be specified for each element of job_range.

jobs_min	The low limit number of jobs per a batch request
jobs_max	The high limit number of jobs per a batch request

Operator privilege is needed.

set execution_queue kernel_param nice = *value queue*

A nice value limit is set in execution queue specified by *queue*. Newly set values do not affect requests that are already queued in queues. This nice value will be succeeded to requests and will become a nice value when requests are executed. Specify a positive or negative integer in value.

Operator privilege is needed.

set execution_queue kernel_param rsg_number = *value queue*

Set a resource sharing group number in execution queue specified by *queue*.

Operator privilege is needed.

set execution_queue lhost cpu_number_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the limit on the number of CPUs per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to *min* and *max*. *Min* have to be less than or equal to *max*.
In this setting, when a request is submitted to the *queue*, the limit on the number of CPUs per logical host(per job) of it is limited in the specified range. If *min* and *max* are equal, only such value can be specified at submitting of a request.
The default value is "(1, unlimited)".

Operator privilege is needed.

set execution_queue lhost cpu_time_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit of CPU time per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of CPU time per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost gpu_number_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the limit on the number of GPUs that can be used per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to min and max. Min have to be less than or equal to max.
In this setting, when a request is submitted to the queue, the limit on the number of GPUs that can be used per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request.

The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost memory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit on maximum memory size that can be used per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit on maximum memory size that can be used per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost stderr_size_range = (*min*, *max*[, *warn*]) *queue*

Specify the (lower limit, upper limit) of the limit on maximum file size for standard error that can be used per logical host (per job) to the queue which is specified by queue. Min have to be less than or equal to max. In this setting, when a request is submitted to a queue, the limit on maximum file size for standard error that can be used per logical host (per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost stdout_size_range = (*min*, *max*[, *warn*]) *queue*

Specify the (lower limit, upper limit) of the limit on maximum file size for standard output that can be used per logical host (per job) to the queue which is specified by queue. Min have to be less than or equal to max.

In this setting, when a request is submitted to a queue, the limit on maximum file size for standard output that can be used per logical host (per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost ve_number_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the limit on number of VE node per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to min and max. Min have to be less than or equal to max.
In this setting, when a request is submitted to the queue, the limit on number of VE node per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost vecpu_time_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit of VE CPU time per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of VE CPU time per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost vememory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum VE memory size per logical

host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum VE memory size per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue lhost virtual_memory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum virtual memory size per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum virtual memory size per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue necmpi_process_manager = { **hydra** | **mpd** } *queue*

Set either **hydra** or **mpd** to the execution queue specified by queue as a process manager used by NEC MPI request. The default is mpd.

Operator privilege is needed.

set execution_queue noaccess *queue*

Disables users in the user list and groups in the group list to access execution queue specified in *queue*. If the mode is altered, the content of the list till that time will be deleted entirely.

Manager privilege is needed.

set execution_queue numa_control =｛on ｜ off｝ *queue*

Set enable or not socket scheduling feature to the execution queue specified by *queue*. If this feature is enabled, this queue must be bind with scalar machine (Linux) which support NUMA architecture. All bound execution host must be same socket configuration.
The default is **off.**

on	Enables socket scheduling feature.
off	Disables socket scheduling feature. (Default).

Operator privilege is needed.

set execution_queue numa_option core_bind_policy = *policy* *queue*

Set the core bind policy to the execution queue specified by *queue* .
Following policy can be specified for *policy*.

concentration	Socket concentration policy (Default). Lesser free core socket is preferentially selected.
balance	Socket decentralization policy. More free core socket is preferentially selected.

Operator privilege is needed.

set execution_queue numa_option memory_allocation_policy = *policy* *queue*

Set the memory allocation policy to the execution queue specified by *queue* .
Following policy can be specified for *policy*.

membind	membind policy. Use the memory which belongs the socket that the job executed only. If the memory is insufficient, it uses swap.
localalloc	localalloc policy (Default). Use the memory which belongs the socket that the job executed

preferentially. If the memory is insufficient, it uses the memory which belongs the socket that is used by the other job.

interleave interleave policy.
Use the memory alternatively the socket that the job executed. If the memory is insufficient, it uses the memory which belongs the socket that is used by the other job.

Operator privilege is needed.

set execution_queue numa_unit_check =｛on ｜off｝ *queue*

Set use or not for numa unit check feature of the queue specified by *queue.*
This feature checks the ratio of per job CPU number limit and memory size limit that specified to the request are match with CPU (core) number and memory size on the socket of execution host.
If this feature is enabled, submitting the request that CPU and memory ratio is not match is refused.

Operator privilege is needed.

set execution_queue partial_process_swapping =｛on ｜off｝ *queue*

Specify if partial process swapping of VEOS is used when the request that is submitted into the queue is suspended. If on is designated, the request is suspended with partial process swapping. If off is designated, the request is suspended without partial process swapping. The default is on.

Operator privilege is needed.

set execution_queue per_job cpu_number_limit = *limit queue*

Set a maximum number of CPUs limit for each job in execution queue specified by *queue*.
This value will be compared with the CPU run limit set in requests when queuing in requests.
Cataloging of requests will be rejected if the value is larger than the value set in queue.
This sub-command is valid only if it is supported by the system.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_job cpu_time_limit = (*limit*) *queue*

Set a CPU time limit value for each job in execution queue specified by *queue*.
This value will be compared with the CPU time limit value set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_job gpu_number_limit = *limit* *queue*

Set a maximum number of GPUs limit for each job in execution queue specified by *queue* that can be used. This value will be compared with the GPU run limit set in requests when queuing in requests. Cataloging of requests will be rejected if the value is larger than the value set in queue.

This sub-command is valid only if GPU is supported by the execution host.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_job memory_size_limit = (*limit*) *queue*

Set a memory-size limit value for each job in execution queue specified by *queue*.
This value will be compared with the memory-size limit value for each job set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.

The default is unlimited.

Operator privilege is needed.

set execution_queue per_job virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory-size limit value for each job in execution queue specified by *queue*. This value will be compared with the virtual memory-size limit value for each job set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc core_size_limit = (*limit*) *queue*

Set a core-file size limit value for each process in execution queue specified by *queue*. This value will be compared with the core-file size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc cpu_time_limit = (*limit*) *queue*

Set a CPU time limit value for each process in execution queue specified by *queue*. This value will be compared with the CPU time limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.

Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc data_size_limit = (*limit*) *queue*

Set a limit value for data segment size for each process in execution queue specified by *queue*. This value will be compared with the limit value for data segment size for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc file_size_limit = (*limit*) *queue*

Set a permanent-file size limit value for each process in execution queue specified by *queue*. This value will be compared with the permanent-file size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc open_file_number_limit = *limit queue*

Set a limit value for the number of files simultaneously opened for each process in execution queue specified by *queue*. This value will be compared with the limit value for the number of files simultaneously opened for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.

A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc stack_size_limit = (*limit*) *queue*

Set a stack-segment size limit value in execution queue specified by *queue*.
This value will be compared with the stack-segment size limit value set in Requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc vecpu_time_limit = (*limit*) *queue*

Set a VE CPU time limit value for each process in execution queue specified by *queue*. This value will be compared with the VE CPU time limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc vememory_size_limit = (*limit*) *queue*

Set a VE memory size limit value for each process in execution queue specified by *queue*. This value will be compared with the VE memory size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.

A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_prc virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory size limit value for each process in execution queue specified by *queue*. This value will be compared with the virtual memory size limit value for each process set in requests to be input.
Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_req elapse_time_limit = (*limit*) *queue*

Set an elapsed-time limit value for each request in execution queue specified by *queue*. This value will be compared with the elapsed-time limit value for each request set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set execution_queue per_req elapse_time_limit = (*limit*) **groups** = *group* *queue*
set execution_queue per_req elapse_time_limit = (*limit*) **groups** = (*group*[,*group* ...]) *queue*
set execution_queue per_req elapse_time_limit = (*limit*) **users** = *user* *queue*
set execution_queue per_req elapse_time_limit = (*limit*) **users** = (*user*[,*user*...]) *queue*

Set an elapsed-time limit value every user *user* or group *group* for each request in execution queue specified by *queue*. Specify group name to *group* and/or user name to *user*. An error message will be output if requests in excess of the number specified by this limit are submitted when the request is submitted with user *user* and/or group *group*.

Operator privilege is needed.

set execution_queue priority = *priority queue*

Set execution queue priority *priority* in *queue*.

Operator privilege is needed.

set execution_queue qattach = *mode queue*

Sets the request submitted to the execution queue specified by *queue* is allowed
to be attached by qattach(1) or not to *mode*.
To allow attaching, specify "enable" to *mode*, and to disallow, specify "disable".
The default is "enable".

Operator privilege is needed.

set execution_queue refuse_submission = (*from*[,*from*...]) *queue*

Set the rejected route from of submitting request in the execution queue *queue*.

The following can be specified in *from*.

qsub ｜ submit	Refuse to submit via qsub
qmove	Refuse to submit via qmove
local_routing	Refuse to routing via local routing queue
remote_routing	Refuse to routing via remote routing queue

Operator privilege is needed.

set execution_queue request_priority_range *privilege* = (*min*,*max*) *queue*

Set the range of request priority per each *privilege* which can be set with a batch request submitting to the execution queue specified by *queue*.
A value from -1024 to 1023 can be specified in *min*, *max*.
An error message will be output if a value is out of this range.

Specify the following values for *min*, *max* of each *privilege*.

manager

The high/low limit of request priority with submitting requests by manager privilege

operator

The high/low limit of request priority with submitting requests by operator privilege

specialuser

The high/low limit of request priority with submitting requests by special user privilege

user

The high/low limit of request priority with submitting requests by user privilege

Manager privilege is needed.

set execution_queue rerun_default = *mode* [**limit** = *n*] *queue*

Set the default value to *mode* for specify the enabling rerun of the request that is submitted to the queue *queue*. To set the default value to enable, set "yes" and to disable, set "no" to *mode*. The value which is specified to rerun_default affects only for the request that not specified -r option of qsub command.
If *mode* is "yes", the upper limit of the rerun count can be set. *n* can be set an integer 1 between 2147483647. If **limit** is omitted, the rerun count is unlimited. If *mode* is "no", **limit** is ignored.
The default is "yes" and the rerun count is unlimited.

Operator privilege is needed.

set execution_queue standard hca_number = *limit* mode=*hca-mode queue*

Specify a default for the limit on the number of HCA port to the queue which is specified by queue. Integer of more than or equal to 0 and less than or equal to 32 or "unlimited" can be specified to limit. If HCA port number limit is not set to requests, this value will become HCA port number limit of requests.

Following value can be specified for hca-mode.

io	HCA for ScaTeFS direct IO
mpi	HCA for MPI
all	Both for ScaTeFS direct IO and MPI

The default value is 0. The limit for each hca-mode is independent and each value is not affects to the others.

Operator privilege is needed.

set execution_queue standard lhost cpu_number_limit = *limit* *queue*

Specify a default for the limit on the number of CPUs per logical host(per job) in execution queue specified by queue. If the limit on the number of CPUs per logical host is not set in requests, this value will become the limit on the number of CPUs per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost cpu_time_limit = (*limit*) *queue*

Specify a default for the limit of CPU time per logical host(per job) in execution queue specified by queue. If the limit of CPU time per logical host is not set in requests, this value will become the limit of CPU time per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost gpu_number_limit = *limit* *queue*

Specify a default for the limit on the number of GPUs that can be used per logical host(per job) in execution queue specified by queue. If the limit on the number of GPUs that can be used per logical host is not set in requests, this value will become the limit

on the number of GPUs that can be used per logical host(per job)for requests.
The default value is 0.

Operator privilege is needed.

set execution_queue standard lhost memory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum memory size that can be used per logical host(per job) in execution queue specified by queue. If the the limit on maximum memory size that can be used per logical host is not set in requests, this value will become the limit on maximum memory size that can be used per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost stderr_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum file size for standard error that can be used per logical host (per job) in execution queue specified by queue. If the limit on maximum file size for standard error that can be used per logical host is not set in requests, this value will become the limit on maximum file size for standard error that can be used per logical host (per job) for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost stdout_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum file size for standard output that can be used per logical host (per job) in execution queue specified by queue. If the limit on maximum file size for standard output that can be used per logical
host is not set in requests, this value will become the limit on maximum file size for standard output that can be used per logical host (per job) for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost ve_number_limit = *limit* *queue*

Specify a default for the limit on number of VE node per logical host(per job) in execution queue specified by queue. If the limit on number of VE node per logical host is not set in requests, this value will become the limit on number of VE node per logical host(per job)for requests.
The default value is 0.

Operator privilege is needed.

set execution_queue standard lhost vecpu_time_limit = (*limit*) *queue*

Specify a default for the limit of VE CPU time per logical host(per job) in execution queue specified by queue. If the limit of VE CPU time per logical host is not set in requests, this value will become the limit of VE CPU time per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost vememory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum VE memory size per logical host(per job) in execution queue specified by queue. If the the limit on maximum VE memory size per logical host is not set in requests, this value will become the limit on maximum VE memory size per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard lhost virtual_memory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum virtual memory size per logical host(per job) in execution queue specified by queue. If the the limit on maximum virtual memory size per logical host is not set in requests, this value will become the limit on maximum virtual memory size per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_job cpu_number_limit = *limit queue*

Set a default for the CPU run limit for each job in execution queue specified by *queue*. If the CPU run limit for each job is not set in requests, this value will become the CPU run limit for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_job cpu_time_limit = (*limit*) *queue*

Set a CPU time default for each job in execution queue specified by *queue*.
If a CPU time limit value for each job is not set with requests, this value will become a CPU time limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_job gpu_number_limit = *limit* *queue*

Set a default for the GPU run limit for each job in execution queue specified by *queue*.
If the GPU run limit for each job is not set in requests, this value will become the GPU run limit for requests. This sub-command is valid only if it is supported by the system.
The default value is 0.

Operator privilege is needed.

set execution_queue standard per_job memory_size_limit = (*limit*) *queue*

Set a memory size default for each job in execution queue specified by *queue*.
If a memory size limit value for each job is not set in requests, this value will become a memory size limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_job virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory size default for each job in execution queue specified by *queue*. If a virtual memory size limit value for each job is not set in requests, this value will become a virtual memory size limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc core_size_limit = (*limit*) *queue*

Set a default for core file size for each process in execution queue specified by *queue*. If a limit value for the core file size for each process is not set in requests, this value will become a limit value for the core file size for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc cpu_time_limit = (*limit*) *queue*

Set a default for the CPU time for each process in execution queue specified by *queue*. If a limit value for CPU time limit for each process is not set in requests, this value will become a limit value for CPU time of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc data_size_limit = (*limit*) *queue*

Set a default for the data segment size for each process in execution queue specified by *queue*. If a limit value for data segments for each process is not set in requests, this value will become a limit value for data segments of requests.

This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc file_size_limit = (*limit*) *queue*

Set a default for the permanent file size for each process in execution queue specified by *queue*. If a limit value for the permanent file size for each process is not set with requests, this value will become a limit value for the permanent file size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc open_file_number_limit = *limit queue*

Set a default for the number of files simultaneously opened for each process in execution queue specified by *queue*.
If a limit value for the number of files simultaneously opened for each process is not set with requests, this value will become a limit value for the number of files simultaneously opened for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc stack_size_limit = (*limit*) *queue*

Set a stack-segment default for each process in execution queue specified by *queue*. If a stack-segment size limit value for each process is not set in requests, this value will be taken as the stack-segment limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc vecpu_time_limit = (*limit*) *queue*

Set a default for the VE CPU time for each process in execution queue specified by *queue*. If a limit value for VE CPU time limit for each process is not set in requests, this value will become a limit value for VE CPU time of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc vememory_size_limit = (*limit*) *queue*

Set a default for the VE memory size for each process in execution queue specified by *queue*. If a limit value for the VE memory size for each process is not set with requests, this value will become a limit value for the VE memory size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_prc virtual_memory_size_limit = (*limit*) *queue*

Set a default for the virtual memory size for each process in execution queue specified by *queue*. If a limit value for the virtual memory size for each process is not set with requests, this value will become a limit value for the virtual memory size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard per_req elapse_time_limit = (*limit*) *queue*

Set a default for an elapsed time for each request in the execution queue specified by *queue*. If a limit value for elapsed time for each request is not set in

requests, this value will become a limit value for elapsed time of requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard venode vecpu_time_limit = (*limit*) *queue*

Specify a default for the limit of VE CPU time per VE node in execution queue specified by queue. If the limit of VE CPU time per VE node is not set in requests, this value will become the limit of VE CPU time per VE node for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue standard venode vememory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum VE memory size per VE node in execution queue specified by queue. If the the limit on maximum VE memory size per VE node is not set in requests, this value will become the limit on maximum VE memory size per VE node for requests.
The default value is unlimited.

Operator privilege is needed.

set execution_queue submit_cpu_unit = { **cpu** | **socket** | **any** } *queue*

Set the way to specify the per job CPU number limit when request submitting.

cpu	Only cpunum_job can be specified.
socket	Only socknum_job can be specified.
any	Either cpunum_job or socknum_job can be specified. (Default)

Operator privilege is needed.

set execution_queue submit_limit = *limit queue*

Set the number of batch requests which can be submitted to an execution queue.
The number of batch requests that can be submitted means the total number of

batch requests existent in an execution queue at the same time.
An error message will be output if requests in excess of the number specified by *limit* are submitted.

A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of execution queue to be set this limit by *queue*.

Operator privilege is needed.

set execution_queue submit_venode_range = (*min, max*)   *queue*

Specify the (lower limit, upper limit) of the total number of VE nodes which is specified by "qsub --venode" to the queue which is specified by queue. Integer of more than or equal to 0 or "unlimited" can be specified to min and max. Min have to be less than or equal to max. When the value outside the range is specified, submission is error. If min and max are equal, only such value can be specified at submitting of a request. If min and max are both 0, only the request without VE node can be submitted.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue subrequest_limit = *limit queue*

When a parametric request is submitted to the execution queue specified by *queue*, the number of sub-requests to specified by qsub -t is limited to limit.

Operator privilege is needed.

set execution_queue supplementary_groups_check { **on** | **off** } *queue*

Set using or not the supplementary group id for the access privilege check of the execution queue specified by *queue*. In case of **on**, the supplementary group id is used for the access privilege check of the execution queue.
In case of **off**, the supplementary group id is not used.

Operator privilege is needed.

set execution_queue suspend_privilege = *privilege queue*

Set the privilege that can suspend request of execution queue specified by *queue*.

The following can be specified in *privilege*.

user

The privilege to suspend is more than User privilege.

specialuser

The privilege to suspend is more than Special user privilege.

groupmanager

The privilege to suspend is more than Group manager privilege.

operator

The privilege to suspend is more than Operator privilege.

manager

The privilege to suspend is more than Manager privilege.

scheduler

The privilege to suspend is more than Scheduler privilege.

Operator privilege is needed.

set execution_queue userexit location = *location* **script** = (*script*) **queue** = *queue*

Set script which is executed in the timing which was specified by *location* in execution queue specified by *queue*.

The following can be specified in *location*.

pre-running	At PRE-RUNNING status
post-running	At POST-RUNNING status

The form of *script* is as follows.

script [*sep script* ...]

script

It specifies the file name of the script. The specified file is thought of as existing at /opt/nec/nqsv/sbin/uex_prog follower on the batch server host. The path element('/') cannot be included in the file name.

The file can be specified in four.

sep

When specifying ',', it executes back and forth a script in series.

When specifying ':', it executes back and forth a script in parallel.

The blank space is automatically deleted that exists before and after of "(" and ")" that used to specify the file name and the delimiter "," and ":". The file name which includes blank space cannot be specified.

Operator privilege is needed.

set execution_queue userexit_timeout = *time queue*

Set timeout to *time* for User Exit script of the queue specified by *queue*. The unit is second. The timeout is off when 0 is specified. Default is 0.

Operator privilege is needed.

set execution_queue userpp_timeout = *time queue*

Set timeout to *time* for User PP script of the queue specified by *queue*. The unit is second. The timeout is off when 0 is specified. Default is 300 seconds.

Operator privilege is needed.

set execution_queue user_submit_limit = *limit queue*

Set the number of batch requests which one user can submit to an execution queue.
The number of batch requests that can be submitted means the total number of batch requests existent in an execution queue at the same time.
If this sub-command is used, one user can submit requests to each execution queue only in a number specified by *limit*. An error message will be output if requests in excess of the number specified by *limit* are submitted.

A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of execution queue to be set this limit by *queue*.

This limit applied set to requests of the user which an upper limit by individually specifying user name is not set to.

Operator privilege is needed.

set execution_queue user_submit_limit = *limit* **users** = *user* *queue*
set execution_queue user_submit_limit = *limit* **users** = (*user* [,*user*...]) *queue*

Set the number of batch requests which the user designated in *user* can submit to an execution queue *queue*. Specify user name to *user*. An error message will be output if requests of user *user* in excess of the number specified by limit are submitted at one time. You can specify integer value which is 0 or more to limit.
"0" indicates unlimited.

Operator privilege is needed.

set execution_queue users = *user queue*
set execution_queue users = (*user* [,*user* ...]) *queue*

Set a user specified by user in the user list of execution queue specified by *queue*. The previous user list information will be deleted.
The mode of the user list needs to be set by sub-command **set execution_queue access | noaccess**.

The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and user-id is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

set execution_queue venode vecpu_time_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit of VE CPU time per VE node to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of VE CPU time perVE node of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set execution_queue venode vememory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum VE memory size per VE node to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum VE memory size per VE node of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue access *queue*

Enables users in the user list and groups in the group list to access interactive queue specified in *queue*. If the mode is altered, the content of the list except "root" till that time will be deleted entirely.

Manager privilege is needed.

set interactive_queue auto_bind_jobserver { **on** | **off** } *queue*

Set bind automatically or not the interactive queue specified by *queue* to a job server when a job server starts. In case of **on**, a job server is bound automatically. In case of **off**, a job server is not bound automatically.

Operator privilege is needed.

set interactive_queue cpunum_per_gpunum = *cpunum queue*

Set the default number of CPU cores per GPU for requests submitted to the GPU-CPU Affinity feature-enabled interactive queue specified in *queue*. An integer from 1 to 2147483647 can be specified for *cpunum*. The default value is 1. The value specified for *cpunum* is the value calculated by the following formula.
Number of CPU cores in 1 socket / Number of GPUs physical close to the socket

Operator privilege is needed.

set interactive _queue custom_resource = *cr_name* **standard** = *std queue*
set interactive _queue custom_resource = *cr_name* **range** = (*min,max*) *queue*
set interactive _queue custom_resource = *cr_name* **permit_unused** = { **yes** | **no** } qu*eue*

Set the limit of custom resource specified by *cr_name* in the queue which is specified by *queue.*

standard is the standard value which is applied when the request is submitted. The integer value of 1 to 2147483647 or unused (0) can be specified for *std*.

range is the value range (lower limit and upper limit) of custom resource usage which can be specified when the request submitted. The integer value of 1 to 2147483647 can be specified for *min* and *max* .

permit_unused is the flag to allow or refuse to specify unused (0) value when submitting request.

yes	Allow to specify unused (0) when submitting request (Default).
no	Refuse to specify unused (0) when submitting request.

Operator privilege is needed.

set interactive_queue defined_ve_number = *venum queue*

Specify default number of VE for automatic calculation of job number to the *queue* at submitting of a request which is specified total number of VE nodes.A value ranging from 1 to 256 can be specified in *venum*. The default is 1.

Operator privilege is required.

set interactive_queue exclusive_submit { ***on*** | ***off*** } *queue*

Specify the allowance for submitting exclusive execution request
on allows the submission of the exclusive execution request. **off** rejects the submission of the exclusive execution request.

Manager privilege is needed.

set interactive_queue gpu_affinity = { **on** | **off** } *<queue>*

Specify whether to use the GPU-CPU Affinity feature in the request submitted to the interactive queue specified in *queue*. If **on** is specified, use the GPU-CPU Affinity feature. If **off** is specified, the GPU-CPU Affinity feature is not used. If you want to use the GPU-CPU Affinity feature, you need to enable Socket Scheduling feature. The default value is **off**.

Operator privilege privilege is needed.

set interactive_queue group_submit_limit = *limit queue*

Set the number of requests which one group can submit to an interactive queue. The number of requests that can be submitted means the total number of requests existent in an interactive queue at the same time.

If this sub-command is used, one group can submit requests to each interactive queue only in a number specified by limit. An error message will be output if requests in excess of the number specified by *limit* are submitted.

A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of interactive queue to be set this limit by *queue*.

This limit is applied to requests of the group which an upper limit by individually specifying group name is not set to.

Operator privilege is needed.

set interactive_queue group_submit_limit = *limit* **groups** = *group* *queue*
set interactive_queue group_submit_limit = *limit* **groups** = (*group*[, *group* ...]) *queue*

Set the number of batch requests which the group designated in *group* can submit to an interactive queue *queue*. Specify group name to *group*. An error message will be output if requests of group *group* in excess of the number specified by limit are submitted at one time. You can specify integer value which is 0 or more to limit.
"0" indicates unlimited.

Operator privilege is needed.

set interactive_queue groups = *group queue*
set interactive_queue groups =(*group*[,*group* ...]) *queue*

Set groups specified by *group* in the group list of interactive queue specified by *queue*. The previous group list information will be deleted. The mode of the group list needs to be set by sub-command **set interactive_queue access | noaccess**.
The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and group-id is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

set interactive_queue hca_number_range = (*min*, *max*) mode=*hca-mode queue*

Specify the range of the limit (lower limit, upper limit) on the number of HCA port to the queue which is specified by *queue*. Integer of more than or equal to 0 and less than or equal to 32 or "unlimited" can be specified to min and max. Min have to be less than or equal to max. If min and max are equal, only such value can be specified at submitting of a request.

Following value can be specified for *hca-mode*.

io	HCA for ScaTeFS direct IO
mpi	HCA for MPI
all	Both for ScaTeFS direct IO and MPI

This value limits the HCA port number which specified for the request which submitted to the *queue*.
The default value is "(0, unlimited)". The limit for each hca-mode is independent and each value is not affects to the others.

Operator privilege is needed.

set interactive_queue hook_function { **on** | **off** } *queue*

Set using or not hook script function to the interactive queue specified by *queue*.
The default is **off.**

on	Use hook script function.
off	Do not use hook script function (Default).

Operator privilege is needed.

set interactive_queue idle_timer = *timer queue*

Sets an idle timer for requests submitted to the interactive queue specified by *queue* as timer minutes. When 0 is specified to timer, idle timer is reset.
Specifying idle timer by qlogin -l is prior to this setting.

Operator privilege is needed.

set interactive_queue intmpi_process_manager = hydra *queue*

Set **hydra** to the interactive queue specified by *queue* as a process manager used by IntelMPI request.
The default is **hydra**.

Operator privilege is needed.

set interactive_queue jobs_range = (*jobs_min*, *jobs_max*) *queue*

Set the range of number of jobs per request to the interactive queue specified by *queue*. A value ranging from 1 to 10240 can be specified in *jobs_min*, *jobs_max*.

An error message will be output if a value is not in this range.

The following can be specified for each element of job_range.

jobs_min	The low limit number of jobs per request
jobs_max	The high limit number of jobs per request

Operator privilege is needed.

set interactive_queue jobs_range = (*jobs_min* , *jobs_max*) **groups** = *group* *queue*
set interactive_queue jobs_range = (*jobs_min* , *jobs_max*) **groups** = (*group*[,*group* ...]) *queue*
set interactive_queue jobs_range = (*jobs_min* , *jobs_max*) **users** = *user queue*
set interactive_queue jobs_range = (*jobs_min* , *jobs_max*) **users** = (*user*[,*user*...]) *queue*

Set the range every user *user* or group *group* of jobs per batch request to the interactive queue specified by *queue*. Specify group name to *group* and/or user name to *user*. A value ranging from 1 to 10240 can be specified in jobs_min, jobs_max. An error message will be output if a value is not in this range.

The following can be specified for each element of job_range.

jobs_min	The low limit number of jobs per a batch request
jobs_max	The high limit number of jobs per a batch request

Operator privilege is needed.

set interactive_queue kernel_param nice = *value queue*

A nice value limit is set in interactive queue specified by *queue*.
Newly set values do not affect requests that are already queued in queues.
This nice value will be succeeded to requests and will become a nice value when requests are executed. Specify a positive or negative integer in *value*.

Operator privilege is needed.

set interactive_queue kernel_param rsg_number = *value queue*

Set a resource sharing group number in interactive queue specified by *queue*.

Operator privilege is needed.

set interactive_queue lhost cpu_number_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the limit on the number of CPUs per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to *min* and *max*. *Min* have to be less than or equal to *max*.
In this setting, when a request is submitted to the *queue*, the limit on the number of CPUs per logical host(per job) of it is limited in the specified range. If *min* and *max* are equal, only such value can be specified at submitting of a request.
The default value is "(1, unlimited)".

Operator privilege is needed.

set interactive_queue lhost cpu_time_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit of CPU time per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of CPU time per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost gpu_number_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the limit on the number of GPUs that can be used per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to min and max. Min have to be less than or equal to max.
In this setting, when a request is submitted to the queue, the limit on the number of GPUs that can be used per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost memory_size_range = (*min, max* [,*warn*])   *queue*

Specify the (lower limit, upper limit) of the limit on maximum memory size that can be used per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit on maximum memory size that can be used per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost ve_number_range = (*min, max*)   *queue*

Specify the (lower limit, upper limit) of the limit on number of VE node per logical host(per job) to the queue which is specified by queue. Integer of more than or equal to 1 or "unlimited" can be specified to min and max. Min have to be less than or equal to max.
In this setting, when a request is submitted to the queue, the limit on number of VE node per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost vecpu_time_range = (*min, max* [,*warn*])   *queue*

Specify the (lower limit, upper limit) of the limit of VE CPU time per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of VE CPU time per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.

The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost vememory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum VE memory size per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum VE memory size per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue lhost virtual_memory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum virtual memory size per logical host(per job) to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum virtual memory size per logical host(per job) of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue noaccess *queue*

Disables users in the user list and groups in the group list to access interactive queue specified in queue. If the mode is altered, the content of the list till that time will be deleted entirely.

Manager privilege is needed.

set interactive_queue numa_control = { **on** | **off** } *queue*

Set enable or not socket scheduling feature to the interactive queue specified by *queue*. If this feature is enabled, this queue must be bind with scalar machine (Linux) which support NUMA architecture. All bound execution host must be same socket configuration.
The default is **off.**

on	Enables socket scheduling feature.
off	Disables socket scheduling feature. (Default).

Operator privilege is needed.

set interactive_queue numa_option core_bind_policy = *policy* *queue*

Set the core bind policy to the interactive queue specified by *queue* .
Following policy can be specified for *policy*.

concentration	Socket concentration policy (Default). Lesser free core socket is preferentially selected.
balance	Socket decentralization policy. More free core socket is preferentially selected.

Operator privilege is needed.

set interactive_queue numa_option memory_allocation_policy = *policy* *queue*

Set the memory allocation policy to the interactive queue specified by *queue* .
Following policy can be specified for *policy*.

membind	membind policy. Use the memory which belongs the socket that the job executed only. If the memory is insufficient, it uses swap.
localalloc	localalloc policy (Default). Use the memory which belongs the socket that the job executed

preferentially. If the memory is insufficient, it uses the memory which belongs the socket that is used by the other job.

interleave — interleave policy.
Use the memory alternatively the socket that the job executed. If the memory is insufficient, it uses the memory which belongs the socket that is used by the other job.

Operator privilege is needed.

set interactive_queue numa_unit_check = { **on** | **off** } *queue*

Set use or not for numa unit check feature of the queue specified by *queue.*
This feature checks the ratio of per job CPU number limit and memory size limit that specified to the request are match with CPU (core) number and memory size on the socket of execution host.
If this feature is enabled, submitting the request that CPU and memory ratio is not match is refused.

Operator privilege is needed.

set interactive_queue per_job cpu_number_limit = *limit queue*

Set a maximum number of CPUs limit for each job in interactive queue specified by *queue* that can be used. This value will be compared with the CPU run limit set in requests when queuing in requests. Cataloging of requests will be rejected if the value is larger than the value set in queues.
This sub-command is valid only if it is supported by the system.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_job cpu_time_limit = (*limit*) *queue*

Set a CPU time limit value for each job in interactive queue specified by *queue*.
This value will be compared with the CPU time limit value set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.

This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_job gpu_number_limit = *limit* *queue*

Set a maximum number of GPUs limit for each job in interactive queue specified by *queue* that can be used. This value will be compared with
the GPU run limit set in requests when queuing in requests. Cataloging of requests will be rejected if the value is larger than the value set in queue. This sub-command is valid only if GPU is supported by the execution host.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_job memory_size_limit = (*limit*) *queue*

Set a memory-size limit value for each job in interactive queue specified by *queue*. This value will be compared with the memory-size limit value for each job set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_job virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory-size limit value for each job in interactive queue specified by *queue*. This value will be compared with the virtual memory-size limit value for each job set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.

Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc core_size_limit = (*limit*) *queue*

Set a core-file size limit value for each process in interactive queue specified by *queue*. This value will be compared with the core-file size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc cpu_time_limit = (*limit*) *queue*

Set a CPU time limit value for each process in interactive queue specified by *queue*. This value will be compared with the CPU time limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc data_size_limit = (*limit*) *queue*

Set a limit value for data segment size for each process in interactive queue specified by *queue*. This value will be compared with the limit value for data segment size for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.

This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc file_size_limit = (*limit*) *queue*

Set a permanent-file size limit value for each process in interactive queue specified by *queue*. This value will be compared with the permanent-file size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc open_file_number_limit = *limit queue*

Set a limit value for the number of files simultaneously opened for each process in interactive queue specified by *queue*. This value will be compared with the limit value for the number of files simultaneously opened for each process set in requests to be input.
Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc stack_size_limit = (*limit*) *queue*

Set a stack-segment size limit value in interactive queue specified by *queue*. This value will be compared with the stack-segment size limit value set in requests to be input. Cataloging of requests will be rejected if the value is larger

than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc vecpu_time_limit = (*limit*) *queue*

Set a VE CPU time limit value for each process in interactive queue specified by *queue*. This value will be compared with the VE CPU time limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc vememory_size_limit = (*limit*) *queue*

Set a VE memory size limit value for each process in interactive queue specified by *queue*. This value will be compared with the VE memory size limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_prc virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory size limit value for each process in interactive queue specified by *queue*. This value will be compared with the virtual memory size

limit value for each process set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
This sub-command is valid only if it is supported by the system.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_req elapse_time_limit = (*limit*) *queue*

Set an elapsed-time limit value for each request in interactive queue specified by *queue*. This value will be compared with the elapsed-time limit value for each request set in requests to be input. Cataloging of requests will be rejected if the value is larger than the value set in queues.
A newly set value will not affect requests that are already queued in queues.
Refer to Resource Limit for the syntax of *limit*.
The default is unlimited.

Operator privilege is needed.

set interactive_queue per_req elapse_time_limit = (*limit*) **groups** = *group* *queue*
set interactive_queue per_req elapse_time_limit = (*limit*) **groups** = (*group*[,*group* ...]) *queue*
set interactive_queue per_req elapse_time_limit = (*limit*) **users** = *user queue*
set interactive_queue per_req elapse_time_limit = (*limit*) **users** = (*user*[,*user*...]) *queue*

Set an elapsed-time limit value every user *user* or group *group* for each request in interactive queue specified by *queue*. Specify group name to *group* and/or user name to *user*. An error message will be output if requests in excess of the number specified by this limit are submitted when the request is submitted with user *user* and/or group *group*.

Operator privilege is needed.

set interactive_queue priority = *priority queue*

Set interactive queue priority *priority* in *queue*.

Operator privilege is needed.

set interactive_queue qattach = *mode queue*

Sets the request submitted to the interactive queue specified by *queue* is allowed to be attached by **qattach(1)** or not to *mode*.
To allow attaching, specify "enable" to *mode*, and to disallow, specify "disable".
The default is "enable".

Manager privilege is needed.

set interactive_queue real_time_scheduling = *mode queue*

Sets the mode of the request behavior when a request submitted to the interactive queue specified by *queue* cannot be scheduled to start immediately.

The following modes can be specified to *mode*.

submit_cancel
    Cancels the request submission.
wait
    Waits the request is scheduled to start.
manual
    The request acts as specified at submission by qlogin -W command.

The default behavior is submit_cancel.

Operator privilege is needed.

set interactive_queue refuse_submission = (*from*[,*from*...]) *queue*

Set the rejected route from of submitting request in the interactive queue *queue*.

The following can be specified in *from*.

qsub | submit	Refuse to submit via qlogin or qrsh

Operator privilege is needed.

set interactive_queue restrict_shell = *path_name queue*

The interactive request submitted to the queue specified by *queue* is forced to execute *path_name* as the staring shell program even if another shell program is specified by qlogin -S option.

Operator privilege is needed.

set interactive_queue standard hca_number = *limit* mode=*hca-mode queue*

Specify a default for the limit on the number of HCA port to the queue which is specified by queue. Integer of more than or equal to 0 and less than or equal to 32 or "unlimited" can be specified to limit. If HCA port number limit is not set to requests, this value will become HCA port number limit of requests.

Following value can be specified for hca-mode.

io	HCA for ScaTeFS direct IO
mpi	HCA for MPI
all	Both for ScaTeFS direct IO and MPI

The default value is 0. The limit for each hca-mode is independent and each value is not affects to the others.

Operator privilege is needed.

set interactive_queue standard lhost cpu_number_limit = *limit* *queue*

Specify a default for the limit on the number of CPUs per logical host(per job) in execution queue specified by queue. If the limit on the number of CPUs per logical host is not set in requests, this value will become the limit on the number of CPUs per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard lhost cpu_time_limit = (*limit*) *queue*

Specify a default for the limit of CPU time per logical host(per job) in execution queue

specified by queue. If the limit of CPU time per logical host is not set in requests, this value will become the limit of CPU time per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard lhost gpu_number_limit = *limit* *queue*

Specify a default for the limit on the number of GPUs that can be used per logical host(per job) in execution queue specified by queue. If the limit on the number of GPUs that can be used per logical host is not set in requests, this value will become the limit on the number of GPUs that can be used per logical host(per job)for requests.
The default value is 0.

Operator privilege is needed.

set interactive_queue standard lhost memory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum memory size that can be used per logical host(per job) in execution queue specified by queue. If the the limit on maximum memory size that can be used per logical host is not set in requests, this value will become the limit on maximum memory size that can be used per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard lhost ve_number_limit = *limit* *queue*

Specify a default for the limit on number of VE node per logical host(per job) in execution queue specified by queue. If the limit on number of VE node per logical host is not set in requests, this value will become the limit on number of VE node per logical host(per job)for requests.
The default value is 0.

Operator privilege is needed.

set interactive_queue standard lhost vecpu_time_limit = (*limit*) *queue*

Specify a default for the limit of VE CPU time per logical host(per job) in execution queue specified by queue. If the limit of VE CPU time per logical host is not set in requests, this value will become the limit of VE CPU time per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard lhost vememory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum VE memory size per logical host(per job) in execution queue specified by queue. If the the limit on maximum VE memory size per logical host is not set in requests, this value will become the limit on maximum VE memory size per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard lhost virtual_memory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum virtual memory size per logical host(per job) in execution queue specified by queue. If the the limit on maximum virtual memory size per logical host is not set in requests, this value will become the limit on maximum virtual memory size per logical host(per job)for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_job cpu_number_limit = *limit queue*

Set a default for the CPU run limit for each job in interactive queue specified by *queue*. If the CPU run limit for each job is not set in requests, this value will become the CPU run limit for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_job cpu_time_limit = (*limit*) *queue*

Set a CPU time default for each job in interactive queue specified by *queue*.
If a CPU time limit value for each job is not set with requests, this value will become a CPU time limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_job gpu_number_limit = *limit*　*queue*

Set a default for the GPU run limit for each job in interactive queue specified by *queue*.
If the GPU run limit for each job is not set in requests, this value will become the GPU run limit for requests. This sub-command is valid only if it is supported by the system.
The default value is 0.

Operator privilege is needed.

set interactive_queue standard per_job memory_size_limit = (*limit*) *queue*

Set a memory size default for each job in interactive queue specified by *queue*.
If a memory size limit value for each job is not set in requests, this value will become a memory size limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_job virtual_memory_size_limit = (*limit*) *queue*

Set a virtual memory size default for each job in interactive queue specified by *queue*. If a virtual memory size limit value for each job is not set in requests, this value will become a virtual memory size limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc core_size_limit = (*limit*) *queue*

Set a default for core file size for each process in interactive queue specified by *queue*. If a limit value for the core file size for each process is not set in requests, this value will become a limit value for the core file size for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc cpu_time_limit = (*limit*) *queue*

Set a default for the CPU time for each process in interactive queue specified by *queue*. If a limit value for CPU time limit for each process is not set in requests, this value will become a limit value for CPU time of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc data_size_limit = (*limit*) *queue*

Set a default for the data segment size for each process in interactive queue specified by *queue*. If a limit value for data segments for each process is not set in requests, this value will become a limit value for data segments of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc file_size_limit = (*limit*) *queue*

Set a default for the permanent file size for each process in interactive queue specified by *queue*. If a limit value for the permanent file size for each process is not set with requests, this value will become a limit value for the permanent file size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc open_file_number_limit = *limit queue*

Set a default for the number of files simultaneously opened for each process in interactive queue specified by *queue*. If a limit value for the number of files simultaneously opened for each process is not set with requests, this value will become a limit value for the number of files simultaneously opened for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc stack_size_limit = (*limit*) *queue*

Set a stack-segment default for each process in interactive queue specified by *queue*. If a stack-segment size limit value for each process is not set in requests, this value will be taken as the stack-segment limit value for requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc vecpu_time_limit = (*limit*) *queue*

Set a default for the VE CPU time for each process in interactive queue specified by *queue*. If a limit value for VE CPU time limit for each process is not set in requests, this value will become a limit value for CPU time of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc vememory_size_limit = (*limit*) *queue*

Set a default for the VE memory size for each process in interactive queue specified by *queue*. If a limit value for the VE memory size for each process is

not set with requests, this value will become a limit value for the VE memory size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_prc virtual_memory_size_limit = (*limit*) *queue*

Set a default for the virtual memory size for each process in interactive queue specified by *queue*. If a limit value for the virtual memory size for each process is not set with requests, this value will become a limit value for the virtual memory size of requests.
This sub-command is valid only if it is supported by the system.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard per_req elapse_time_limit = (*limit*) *queue*

Set a default for an elapsed time for each request in the interactive queue specified by *queue*. If a limit value for elapsed time for each request is not set in requests, this value will become a limit value for elapsed time of requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard venode vecpu_time_limit = (*limit*) *queue*

Specify a default for the limit of VE CPU time per VE node in execution queue specified by queue. If the limit of VE CPU time per VE node is not set in requests, this value will become the limit of VE CPU time per VE node for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue standard venode vememory_size_limit = (*limit*) *queue*

Specify a default for the limit on maximum VE memory size per VE node in execution queue specified by queue. If the the limit on maximum VE memory size per VE node not set in requests, this value will become the limit on maximum VE memory size per VE node for requests.
The default value is unlimited.

Operator privilege is needed.

set interactive_queue submit_cpu_unit = { **cpu** | **socket** | **any** } *queue*

Set the way to specify the per job CPU number limit when request submitting.

cpu	Only cpunum_job can be specified.
socket	Only socknum_job can be specified.
any	Either cpunum_job or socknum_job can be specified. (Default)

Operator privilege is needed.

set interactive_queue submit_limit = *limit queue*

Set the number of requests which can be submitted to an interactive queue.
The number of requests that can be submitted means the total number of requests existent in an interactive queue at the same time.
An error message will be output if requests in excess of the number specified by *limit* are submitted.

A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of interactive queue to be set this limit by *queue*.

Operator privilege is needed.

set interactive_queue submit_venode_range = (*min, max*) *queue*

Specify the (lower limit, upper limit) of the total number of VE nodes which is specified by "qsub --venode" to the queue which is specified by queue. Integer of more than or equal to 0 or "unlimited" can be specified to min and max. Min have to be less than or equal to

max. When the value outside the range is specified, submission is error. If min and max are equal, only such value can be specified at submitting of a request. If min and max are both 0, only the request without VE node can be submitted.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue supplementary_groups_check { **on** | **off** } *queue*

Set using or not the supplementary group id for the access privilege check of the interactive queue specified by *queue*. In case of on, the supplementary group id is used for the access privilege check of the interactive queue.
In case of off, the supplementary group id is not used.

Operator privilege is needed.

set interactive_queue suspend_privilege = *privilege queue*

Set the privilege that can suspend request of interactive queue specified by *queue*.

The following can be specified in *privilege*.

user
The privilege to suspend is more than User privilege.
specialuser
The privilege to suspend is more than Special user privilege.
groupmanager
The privilege to suspend is more than Group manager privilege.
operator
The privilege to suspend is more than Operator privilege.
manager
The privilege to suspend is more than Manager privilege.
scheduler
The privilege to suspend is more than Scheduler privilege.

Operator privilege is needed.

set interactive_queue userexit location = *location* **script** = (*script*) **queue** = *queue*

Set script which is executed in the timing which was specified by *location* in interactive queue specified by *queue*.

The following can be specified in *location*.

pre-running

At PRE-RUNNING status

post-running

At POST-RUNNING status

The form of *script* is as follows.

script [*sep script* ...]

script

It specifies the file name of the script. The specified file is thought of as existing at /opt/nec/nqsv/sbin /uex_prog follower on the batch server host.

The path element('/') cannot be included in the file name.

The file can be specified in four.

sep

When specifying ',', it executes back and forth a script in series.

When specifying ':', it executes back and forth a script in parallel.

The blank space is automatically deleted that exists before and after of "(" and ")" that used to specify the file name and the delimiter "," and ":". The file name which includes blank space cannot be specified.

Operator privilege is needed.

set interactive_queue userexit_timeout = *time queue*

Set timeout to *time* for User Exit script of the queue specified by*queue*. The unit is second. The timeout is off when 0 is specified. Default is 0.

Operator privilege is needed.

set interactive_queue userpp_timeout = *time queue*

Set timeout to *time* for User PP script of the queue specified by*queue*. The unit is second. The timeout is off when 0 is specified. Default is 300 seconds.

Operator privilege is needed.

set interactive_queue user_submit_limit = *limit queue*

Set the number of requests which one user can submit to an interactive queue.
The number of requests that can be submitted means the total number of requests existent in an interactive queue at the same time.
If this sub-command is used, one user can submit requests to each interactive queue only in a number specified by *limit*.
An error message will be output if requests in excess of the number specified by limit are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of interactive queue to be set this limit by *queue*.

This limit is applied to requests of the user which an upper limit by individually specifying user name is not set to.

Operator privilege is needed.

set interactive_queue user_submit_limit = *limit* **users** = *user queue*
set interactive_queue user_submit_limit = *limit* **users** = (*user* [,*user*...]) *queue*

Set the number of batch requests which the user designated in *user* can submit to an interactive queue *queue*. Specify user name to *user*. An error message will be output if requests of user *user* in excess of the number specified by limit are submitted at one time. You can specify integer value which is 0 or more to limit.
"0" indicates unlimited.

Operator privilege is needed.

set interactive_queue users = *user queue*
set interactive_queue users = (*user* [,*user* ...]) *queue*

Set a user specified by user in the user list of interactive queue specified by *queue*. The previous user list information will be deleted. The mode of the user list needs to be set by sub-command **set interactive_queue access | noaccess**.
The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and user-id is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

set interactive_queue venode vecpu_time_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of the limit of VE CPU time per VE node to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, the limit of VE CPU time per VE node of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set interactive_queue venode vememory_size_range = (*min, max* [,*warn*]) *queue*

Specify the (lower limit, upper limit) of limit on maximum VE memory size per VE node to the queue which is specified by queue. Min have to be less than or equal to max.
In this setting, when a request is submitted to a queue, limit on maximum VE memory size per VE node of it is limited in the specified range. If min and max are equal, only such value can be specified at submitting of a request. Warning value warn can be specified simultaneously. When warn is not specified, max is set as warn automatically.
Refer to "Resource Limit" for the syntax of limitation value.
The default value is "(0, unlimited)".

Operator privilege is needed.

set job_server hca_failure_check = { off | down | unbind } job_server_id = *jsvid*

Configure the action of job server that is specified by *jsvid* when HCA failure detected.
If set **off**, do nothing when HCA failure detected.
If set **unbind**, unbind the JSV from queue when HCA failure detected. If there is running jobs **on** the execution host, which execution continue.
If set **down**, stop JSV when HCA failure detected. If there is running jobs on the execution host, the jobs are rerun.

Operator privilege is needed.

set job_server health_check_action = {nothing | down | unbind} job_server_id = *jsvid*

Configure the action of job server that is specified by *jsvid* when node health check function detects failure.
If set **nothing**, do nothing when failure is detected.
If set **unbind**, unbind the JSV from queue when failure is detected. If there is running jobs **on** the execution host, which execution continue.
If set **down**, stop JSV when failure is detected. If there is running jobs on the execution host, the jobs are rerun.

Operator privilege is needed.

set job_server migration_file_transfer_parameter interface_hostname = *hostname*
socketbuffer_size = (*socket_size*) **iobuffer_size** = (*io_size*) **job_server_id** = *job_server_id*

Set the file routing parameter on job migration to the job server specified by *job_server_id*. Specify the hostname of TCP/IP Network Interface on file routing to hostname. Specify integer above 0 to the size of socket buffer to *socket_size*. The default socket buffer size of OS will be used when 0 is specified. Specify the size of file I/O to *io_size*. A value ranging from 1 to 8 Mega bytes can be specified to it. The default value is 512 kilobytes.

Operator privilege is needed.

set network_queue batch_request_run_limit = *run_limit queue*

Set the network request run limit for every batch request which was specified by

run_limit to the network queue specified by *queue*.
run_limit can specify from 1 to 1000.

Operator privilege is needed.

set network_queue priority = *priority queue*

Set network queue priority *priority* in *queue*.

Operator privilege is needed.

set network_queue run_limit = *run_limit queue*

Change the run limit of network queue specified by *queue* to the number specified by *run_limit*. Run_limit defines the maximum number of requests that are allowed to be routed in network queues simultaneously.

Operator privilege is needed.

set network_queue staging_extended_buffer_size = *size queue*

An extended buffer with the size specified by *size* is used for the staging file transfer of the network_queue specified by *queue*. Size can be specified in the range from 0 to 512.
The unit of size is kilobyte. When 0 is specified in *size*, standard buffer for the staging file transfer is used.

Operator privilege is needed.

set network_queue staging_machine = *host_name queue*

Hosts specified by *host_name* are made client hosts as objects of staging for network queue specified by *queue*.

Operator privilege is needed.

set network_queue staging_method = *method queue*

Set the staging method specified by *method* to the network queue specified by *queue*.

The following can be specified in *method*.

internal	It uses a standard staging method in the batch server.
external	It uses an external staging method of the user definition.

Operator privilege is needed.

set node_group comment = "*comment*" *ngrp_name*

Sets the comment string as the "*comment*" of the node group specified by *ngrp_name*.
If a comment is already set, the new comment will be overwritten.

Operator privilege is needed.

set node_group switch_layer = *layer ngrp_name*

Specify the level of the switch layer in layer. 1 to 4 can be specified.
This subcommand can only be executed for node groups of type nw_topo.

Operator privilege is needed.

set node_group network = *network_name ngrp_name*

Set the information of the cloud bursting node group specified in *ngrp_name*.
Specify the name of the network in the cloud environment where the cloud instance to be started is located in *network_name*. Specify it if you need to distinguish between networks in a cloud environment. You can specify up to 255 characters.
This subcommand can only be executed for node groups of type cloud.

Operator privilege is needed.

set node_group priority= *priority ngrp_name*

Specify the priority of the cloud bursting node group in *priority*. 0 to 63 can be specified.
This subcommand can only be executed for node groups of type cloud.

Operator privilege is needed.

set openstack_template image = *OS_image　template_name*
set openstack_template flavor = *flavor_name　template_name*
set openstack_template cpu = *cpunum　template_name*
set openstack_template memsz = *memory_size　template_name*
set openstack_template gpu = *gpunum　template_name*
set openstack_template boot_timeout *= timeout　template_name*
set openstack_template stop_timeout *= timeout　template_name*
set openstack_template custom = "*custom_define*"　*template_name*
set openstack_template comment = "*comment*"　*template_name*

Set the parameters of OpenStack template specified by *template_name.*
The template which is used by any request cannot be set.

Specify following parameters for the template.

image　OS disk image name which specified in OpenStack.
flavor　Flavor name which specified in OpenStack.
cpu　Number of CPUs. Specify the integer of one or more.
memsz　Amount of memory size. Specify the integer of one or more with the unit (B, KB, MB, GB, TB, PB,EB).
gpu　Number of GPUs. Specify the integer of one or more.
boot_timeout　Timeout for booting the VM or Baremetal.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 900 seconds.
stop_timeout　Timeout for stopping the VM or Baremetal.
The unit is second. Specify integer value of 1 to 2147483647.
Default is 900 seconds.
custom　The custom value which is specified for the booting environment.
comment　The comment for the template.

Manager privilege is needed.

set routing_queue access *queue*

Enable accessing of routing queue specified by *queue* to users in the user list and groups in the group list.
In case the mode is altered, entire contents of the lists till that time will be deleted.

Manager privilege is needed.

set routing_queue noaccess *queue*

Disable accessing of routing queue specified by *queue* to users in the user list and groups in the group list. In case the mode is altered, entire contents of the lists till that time will be deleted.

Manager privilege is needed.

set routing_queue destination = *destination queue*
set routing_queue destination = (*destination*[,*destination*...]) *queue*

Relates one or more destination queues specified by destination to routing queue specified by *queue*.
It is also possible to move requests between NQSV batch systems by setting routing queues or execution queues on a remote NQSV batch system as a destination queue.

Operator privilege is needed.

set routing_queue group_submit_limit = *limit queue*

Set the number of batch requests which one group can submit to a routing queue.
The number of batch requests that can be submitted means the total number of batch requests existent in a routing queue at the same time.

If this sub-command is used, one group can submit requests to each routing queue only in a number specified by *limit*. An error message will be output if requests in excess of the number specified by *limit* are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of routing queue to be set this limit by *queue*.

This limit is applied to requests of the group which an upper limit by individually specifying group name is not set to.

Operator privilege is needed.

set routing_queue group_submit_limit = *limit* **groups** = *group* *queue*
set routing_queue group_submit_limit = *limit* **groups** = (*group*[, *group* ...]) *queue*

Set the number of batch requests which the group designated in *group* can submit to an routing queue *queue*. Specify group name to *group*. An error message will be output if requests of group *group* in excess of the number specified by limit are submitted at one time. You can specify integer value which is 0 or more to limit.
"0" indicates unlimited.

Operator privilege is needed.

set routing_queue groups = *group queue*
set routing_queue groups =(*group*[,*group* ...]) *queue*

Set groups specified by group in the group list of routing queue specified by *queue*. The previous group list information will be deleted. The mode of the group list needs to be set by sub-command **set routing_queue access | noaccess**.
The following two methods are available to specify *group*.

group-name
[*group-id*]

For example, if the *group-name* is "group1" and group-id is "2001" specify "group1" or "[2001]".

Manager privilege is needed.

set routing_queue hook_function { **on** | **off** } *queue*

Set using or not hook script function to the routing queue specified by *queue*.
The default is **off.**

on	Use hook script function.
off	Do not use hook script function (Default).

Operator privilege is needed.

set routing_queue priority = *priority queue*

Set routing queue priority *priority* in *queue*.

Operator privilege is needed.

set routing_queue refuse_submission = (*from*[,*from*...]) *queue*

Set the rejected route from of submitting request in the routing queue *queue*.

The following can be specified in *from*.

qsub ｜ submit	Refuse to submit via qsub
qmove	Refuse to submit via qmove
local_routing	Refuse to routing via local routing queue
remote_routing	Refuse to routing via remote routing queue

Operator privilege is needed.

set routing_queue run_limit = *run_limit queue*

Change the run limit of routing queue specified by *queue* to the number specified by *run_limit*. Run_limit defines the maximum number of requests that are allowed to be routed in routing queues simultaneously.

Operator privilege is needed.

set routing_queue submit_limit = *limit queue*

Set the number of batch requests which can be submitted to an routing queue.
The number of batch requests that can be submitted means the total number of batch requests existent in a routing queue at the same time.
An error message will be output if requests in excess of the number specified by *limit* are submitted.

A value ranging from 0 to the maximum value of int can be specified in *limit*.

"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of routing queue to be set this limit by *queue*.

Operator privilege is needed.

set routing_queue supplementary_groups_check { **on** | **off** } *queue*

Set using or not the supplementary group id for the access privilege check of the routing queue specified by *queue*. In case of on, the supplementary group id is used for the access privilege check of the routing queue. In case of off, the supplementary group id is not used.

Operator privilege is needed.

set routing_queue user_submit_limit = *limit queue*

Set the number of batch requests which one user can submit to a routing queue.
The number of batch requests that can be submitted means the total number of batch requests existent in a routing queue at the same time.
If this sub-command is used, one user can submit requests to each routing queue only in a number specified by *limit*. An error message will be output if requests in excess of the number specified by *limit* are submitted.
A value ranging from 0 to the maximum value of int can be specified in *limit*.
"0" indicates unlimited. The default is 0 (unlimited).

Specify the name of routing queue to be set this limit by *queue*.

This limit is applied to requests of the group which an upper limit by individually specifying user name is not set to.

Operator privilege is needed.

set routing_queue user_submit_limit = *limit* **users** = *user* *queue*
set routing_queue user_submit_limit = *limit* **users** = (*user* [,*user*...]) *queue*

Set the number of batch requests which the user designated in *user* can submit to an routing queue *queue*. Specify user name to *user*. An error message will be output if

requests of user *user* in excess of the number specified by limit are submitted at one time. You can specify integer value which is 0 or more to limit.
"0" indicates unlimited.

Operator privilege is needed.

set routing_queue users = *user queue*
set routing_queue users = (*user* [,*user* ...]) *queue*

Set a user specified by user in the user list of routing queue specified by *queue*.
The previous user list information will be deleted. The mode of the user list needs to be set by sub-command set routing_queue access | noaccess.
The following two methods are available to specify *user*.

user-name
[*user-id*]

For example, if the *user-name* is "user1" and user-id is "1001" specify "user1" or "[1001]".

Manager privilege is needed.

show all

Displays an overview of information on batch server, queues, jobservers, node groups and scheduler.

show all_queue

Displays states of all queues.

show batch_server

Displays states of batch server.

show execution_queue [= *queue*]

Displays states of all batch queues.
If *queue* is specified, limitation will be made to this queue.

show interactive_queue [= *queue*]

Displays states of all interactive queues.
If *queue* is specified, limitation will be made to this queue.

show job_server

Displays states of job servers.

show limit

Displays status related to the limitation specified by individual designation of the group name and the user name.

show long all_queue

Displays states of all queues in an extended form.

show long execution_queue [= *queue*]

Displays states of all batch queues in an extended form.
If *queue* is specified, limitation will be made to this queue.

show long interactive_queue [= *queue*]

Displays states of all interactive queues in an extended form.
If *queue* is specified, limitation will be made to this queue.

show long network_queue [= *queue*]

Displays states of all network queues in an extended form.
If *queue* is specified, limitation will be made to this queue.

show long routing_queue [= *queue*]

Displays states of all routing queues in an extended form.
If *queue* is specified, limitation will be made to this queue.

show network_queue [= *queue*]

Displays states of all network queues.
If *queue* is specified, limitation will be made to this queue.

show node_group [= *ngrp_name*]

Displays states of all node groups.
If *ngrp_name* is specified, only the information of the specified node group is displayed.

show routing_queue [= *queue*]

Displays states of all routing queues.
If *queue* is specified, limitation will be made to this queue.

show scheduler

Displays states of batch scheduler.

shutdown

Shuts down the NQSV.

Manager privilege is needed.

start all queue

All batch queues, interactive queues, routing queues and network queues on the batch server are activated and execution of the requests in the queues can be started.

Operator privilege is needed.

start execution_queue = *queue*

Execution queue specified by *queue* is activated and execution of the requests in

the execution queue are started.
If specified execution queue is already activated, no execution will take place.

Operator privilege is needed.

start interactive_queue = *queue*

The interactive queue specified by *queue* is activated and execution of the requests in the queue can be started.

Operator privilege is needed.

start job_server execution_host = *host_name* **job_server_id** = *n*
start job_server execution_host = *host_name*
start job_server job_server_id = *n*
start job_server node_group = *ngrp_name*
start job_server all

The job server is started on the execution host specified by *host_name* with the job server number *n*. If the job server is registered to the batch server, the job server number *n* must be match to the registered number.
And also when the job server is registered to the batch server by **attach execution_host** sub-command, it is possible to start job server specifying only *host_name* of the execution host or job server number *n*.

By specifying a node group to *ngrp_name*, all jobservers included in the node group can be started.
And **start job_server all** starts all jobservers registered to the batch server.

To execute this command, the launcher daemon (nqs_lchd) must be running on the execution host.

Manager privilege is needed.

start network_queue = *queue*

Network queue specified by *queue* is activated and execution of the requests in the network queue are started.

If specified network queue is already activated, no execution will take place.

Operator privilege is needed.

start routing_queue = *queue*

Routing queue specified by *queue* is activated and execution of the requests in the routing queue are started.
If specified routing queue is already activated, no execution will take place.

Operator privilege is needed.

stop all queue

All batch queues, interactive queues, routing queues and network queues in the system are stopped and requests in the queues are "frozen" or stopped execution. However, execution of those requests that are currently executed can be continued for completion.
New requests can be input to these queues. Nevertheless, these requests cannot be executed by the same method as that for the requests that resided in the queues.

Operator privilege is needed.

stop execution_queue = *queue*

Batch queue specified by *queue* is stopped and requests in the execution queues are "frozen" or stopped execution. However, execution of those requests that are currently executed can be continued for completion.
New requests can be input to the execution queue. Nevertheless, these requests cannot be executed by the same method as that for the requests that resided in the queue.

Operator privilege is needed.

stop interactive_queue = *queue*

Interactive queue specified by *queue* is stopped and requests in the interactive queues are "frozen" or stopped execution. However, execution of those requests that are currently executed can be continued for completion. New requests can be input to the interactive queue. Nevertheless, these requests cannot be executed by the same method as that for the requests that resided in the queue.

Operator privilege is needed.

stop job_server execution_host = *host_name* [**force**]
stop job_server job_server_id = *n* [**force**]
stop job_server node_group = *ngrp_name* [**force**]
stop job_server all [**force**]

The specified job server will be shutdown. Specify the job server by hostname of the execution_host (*host_name*) or job server number (*n*).
And it is possible to shutdown all jobservers included in the node group specified by *ngrp_name*.
stop job_server all shutdowns all jobservers registered to the batch server.

If not specified **force**, the shutdown of the job server with jobs is rejected. When specified **force**, the shutdown will be executed regardless of jobs. In this case, the executing jobs will abort.

Manager privilege is needed.

stop network_queue = *queue*

Network queue specified by *queue* is stopped and requests in the network queues are "frozen" or stopped execution. However, execution of those requests that are currently executed can be continued for completion.
New requests can be input to the network queue. Nevertheless, these requests cannot be executed by the same method as that for the requests that resided in the network queue.

Operator privilege is needed.

stop routing_queue = *queue*

Routing queue specified by *queue* is stopped and requests in the routing queues

are "frozen" or stopped execution. However, execution of those requests that are currently executed can be continued for completion.
New requests can be input to the routing queue.

Nevertheless, these requests cannot be executed by the same method as that for the requests that resided in the routing queue.

Operator privilege is needed.

unbind execution_queue job_server *queue* **job_server_id** = *n*

An execution queue specified by queue and a job server matching the job server ID specified by *n* is unbound.

Manager privilege is needed.

unbind execution_queue node_group *queue* **node_group** = *ngrp_name*

An execution queue specified by *queue* and a node group whose name is *ngrp_name* is unbound.

Manager privilege is needed.

unbind execution_queue scheduler *queue*

A scheduler bound to an execution queue specified by *queue* is unbound.

Manager privilege is needed.

unbind interactive_queue job_server *queue* **job_server_id** = *n*

The interactive queue specified by *queue* and a job server matching the job server ID specified by *n* is unbound.

Manager privilege is needed.

unbind interactive_queue node_group *queue* **node_group** = *ngrp_name*

The node group whose name is *ngrp_name* is unbound from the interactive queue specified by *queue*.

Manager privilege is needed.

unbind interactive_queue scheduler *queue*

The scheduler bound to the interactive queue specified by *queue* is unbound.

Manager privilege is needed.

unlock cloud_template = *template_name*

Unlock the cloud bursting template specified by *template_name*.

Manager privilege is needed.

unlock container_template = *template_name*

Unlock the container template specified by *template_name*.

Manager privilege is needed.

unlock node_group = *ngrp_name*

Unlock the cloud bursting node group specified by *ngrp_name*.

Manager privilege is needed.

unlock openstack_template = *template_name*

Unlock the OpenStack template specified by *template_name*.

Manager privilege is needed.

update license

Batch server read setting of license in nqsd.conf and get license from license server again.

This is used at changing setting of license in nqsd.conf or at changing of the license number on license server.
When an error occur, please confirm the possession situation of license.

Manager privilege is needed.

RESOURCE LIMIT

The forms of limit values specified by Commands set is as follows:

limit = *max_limit*[,*warn_limit*]

max_limit :	Maximum value of a limit
warn_limit :	Warning value of a limit

1. Time Limit

Limits on time are specified in the following form:

[[*hours*:]*minutes*:]*seconds*[.*fraction*]

Specify the character string "**unlimited**" for unlimited.

[Specification Example]

1234:58:21.29	1234 hours 58 minutes and 21.29 seconds
59:01	59 minutes and 1 second
12345	12345 seconds
121.1	121.1 seconds

2. Size Limit

Specify size limits in the following form:

integer[.*fraction*][*units*]

The following *units* can be specified in "units".

b	Byte
kb	Kilobyte (1024 b)
mb	Megabyte (1024 kb)
gb	Gigabyte (1024 mb)
tb	Terabyte (1024 gb)
pb	Petabyte (1024 tb)

eb	Exabyte (1024 pb)

Omission of units will be construed as "byte".

Specify the character string "**unlimited**" for unlimited.

[Specification Example]

1234	1234Bytes
1234kb	1234Kilobytes
1234.5gb	1234.5Gigabytes

ENVIRONMENT VARIABLES

NQSV_PRIV or NQSII_PRIV

Default access privilege for command execution.

PRIV_MGR	Manager privilege.
PRIV_OPE	Operator privilege.
PRIV_GMGR	Group manager privilege.
PRIV_SPU	Special user privilege.
PRIV_USR	User privilege.

SEE ALSO

nmapmgr(1M), qdel(1), qstat(1), qsub(1), qlogin(1)

NOTES

The settings of kernel paramaters and resource limits for queues are inherited to submitted requests at the submission. Therefore, if the settings of a queue are changed, the attributes of already submitted requests are not affected from that.

NOTICE

2.5. sbuedit(1M)

NAME

sbuedit -- displays and edits the accounting rate

SYNOPSIS

/opt/nec/nqsv/bin/sbuedit {add|upd}
queue_name:*acct_item*=r*ate_in_prime*[/*rate_in_nonprime*][,*acct_item*=...]

/opt/nec/nqsv/bin/sbuedit del *queue_name*

/opt/nec/nqsv/bin/sbuedit {add|upd}
-t *template_name*:*acct_item*=r*ate_in_prime*[/*rate_in_nonprime*][,*acct_item*=...]

/opt/nec/nqsv/bin/sbuedit del -t *template_name*

/opt/nec/nqsv/bin/sbuedit [-N]

DESCRIPTION

The sbuedit command displays, registers, updates, and deletes the accounting rate of the budget management items.

If the command is executed with no option, all the registered accounting rates are displayed for each queue. There is no display for queues that the accounting rate is not set.
In the following example, accounting rates for one queue are displayed.

```
-----------------------------------------------------
QUEUE NAME [quename]
-----------------------------------------------------
CPU             =    10.00/5.00
MEM             =     0.00/0.00
JOB             =     0.00
GPUNUM          =     0.00
REQVE           =     0.00
RSVVE           =     0.00
ELAPSE          =     0.00
RESERVE         =     0.00
DEC             =     0.00
ACT             =     1.00
PRI_MAX         =     1.00
PRI_MIN         =     1.00
```

```
Custom Resource Rate
Power           =   10.00
```

When the custom resource rate isn't set, Custom Resource Rate isn't displayed. The requests that are submitted to a queue to which the accounting rate is not set and the resource reservation sections that are created specifying a queue to which the accounting rate is not set are out of budget checking and calculating charges.

OPTIONS

The sbuedit command specifies the operation, setting target, and applicable items as follows. (template name is NOT available for the environment whose execution host is SX-Aurora TSUBASA system.)

```
# sbuedit add queue1:CPU=5/10,CR_Power=10.0
# sbuedit add -t template1:ELAPSE=10,RESERVE=10.0
```

add Adds a new accounting rate. If the specified queue name or template name already exists, an error message is displayed.

upd Updates the accounting rate. The accounting rate values for the specified queue name or template name is replaced. If the specified queue name or template name does not exist, an error message is displayed.

del Deletes the accounting rate. If the specified node queue name or template name does not exist, an error message is displayed.

-N Displays the sbuedit version information. This option takes priority even when other options are specified, which are then disregarded.

queue_name

Sets an accounting rate for queue which specified by *queue_name*.

-t *template_name*

Sets an accounting rate for queue which specified by *template_name*.

acct_item

Sets an accounting item described below; CPU, MEM, and so on.

rate_in_prime

Sets an accounting rate for prime time.

rate_in_nonprime

Sets an accounting rate for non-prime time. If omitted, the same value as prime time is set.

ACCOUNTING ITEMS

Accounting items are described as follows. When setting the custom resource rate, specify a "CR_" followed by a custom resource name as acct_item.

CPU Accounting rate for CPU usage per second.

This item can be specified for prime time and non-prime time separately.

MEM Accounting rate for unit memory usage (1KB * MIN).

This item can be specified for prime time and non-prime time separately.

JOB Accounting rate per job.

GPUNUM

Accounting rate for the elapsed time of a GPU per second.

REQVE

Accounting rate for the elapsed time of a VE node per second.
Requested VE nodes is used for calculation of the accounting.
This item is available only for the environment whose execution host is SX-Aurora TSUBASA system.

RSVVE

Accounting rate for the elapsed time of a VE node per second.
Reserved VE nodes is used for calculation of the accounting.
This item is available only for the environment whose execution host is SX-Aurora TSUBASA system.

ELAPSE

Accounting rate for the elapsed time of a job per second.

RESERVE

Accounting rate for the resource reservation per node * second.

DEC Weight value for the requested resource.

ACT Weight value for the real resource usage.

PRI_MAX

Weight value when the priority is minimum.

The value must be greater or equal to 1.0.

PRI_MIN

Weight value when the priority is maximum.

The value must be between 0.0 and 1.0.

CR_*crname*

Accounting rate for the specified custom resource. This rate can be specified up to 20 resources. When delete a custom resource rate, specify 0.0 for the custom resource rate value.

SEE ALSO

budgetedit(1M-N), budgetcheck(1-N).

2.6. sceditacct(1M)

NAME

sceditacct -- edit accounting database

SYNOPSIS

/opt/nec/nqsv/bin/sceditacct -j [-s [*date*]] [-e [*date*]] [-S [*date*]] [-E [*date*]]
[-u *user*|-g *group*|-x *acct-code*] [-h *hostname*][-d *database*][-m *database*|-c *database*]

/opt/nec/nqsv/bin/sceditacct -r [-s [*date*]] [-e [*date*]] [-S [*date*]] [-E [*date*]]
[-u *user*|-g *group*|-x *acct-code*] [-f] [-q [date]][-Q [date]]
[-d *database*][-m *database*|-c *database*]

/opt/nec/nqsv/bin/sceditacct -y [-s [*date*]] [-e [*date*]] [-S [*date*]] [-E [*date*]]
[-u *user*|-g *group*|-x *acct-code*] [-d *database*][-m *database*|-c *database*]

date：MM/DD[/YY]

/opt/nec/nqsv/bin/sceditacct -N

DESCRIPTION

sceditacct command will increase the data access performance of the accounting Database by deleting or saving unnecessary parts of request accounting data, job accounting data, or reservation accounting data which are collected in the accounting system.

You have to specify at least one option using this command.
The following options can be specified.

-N Show the version of sceditacct. This option has the highest priority

-j Job accounting data will be edited.

-r Request accounting data will be edited.

-y Resource reservation accounting data will be edited.

-u *user*

The data of specified user is deleted or saved.

-g *group*

The data of specified group is deleted or saved.

-x *acct-code*

The data of specified acct-code is deleted or saved.

-s *date*

Delete or save records of requests, jobs or reservations that finished or after the specified date.

-e *date*

Delete or save records of requests, jobs or reservations that started at or before the specified date.

-S *date*

Delete or save records of requests, jobs or reservations that started at or after the specified date.

-E *date*

Delete or save records of requests, jobs or reservations that finished at or before the specified date.

-q *date*

Delete or save records of requests that queued at or after the specified date and deleted by qdel before running.

-Q *date*

Delete or save records of requests that queued at or before the specified date and deleted by qdel before running.

-h *hostname*

The data of specified hostname is deleted or saved. This option is effective only if -j option is specified.

-f

Delete or save records on conditions indicated by other option include the requests that are not finished. When this option is not indicated, sceditacct doesn't operate the requests that are not finished. This option is effective only if -r option is specified.

-d *database*

Specify the target database to be edited. Specify the file name if the target database is a reservation account database. Specify the directory name if the target database is a request account database or a job account database.

If this option is not specified, the default database is used.
If -j option is specified, /var/opt/nec/nqsv/asv/jacct, jacct_cr and jacct_ps are used.
If -r option is specified, /var/opt/nec/nqsv/asv/racct, racct_tally, racct_cr, racct_cr_tally and racct_ps_tally are used. If -y option is specified, /var/opt/nec/nqsv/asv/rsvacct is used.

The multi database files can be specified by delimiting by comma.

-m *database*

Specify the target database file (-y) or directory (-j, -r) where you want to save the accounting data. The saved data will be deleted on the original database. If this option is specified, the accounting data will be moved from the original database to the target database.

-c *database*

Specify the target database file (-y) or directory (-j, -r) where you want to save the accounting data. The difference from -m option is that the saved data will remain on the original database without being deleted. If this option is specified, the accounting data will be copied from the original database to the target database. If -m and -c option are not specified, the accounting data will be deleted on the original database.

SEE ALSO

scacctjob(1-N), scacctreq(1-N), scacctrsv(1-N).

2.7. scsumacct(1M)

NAME

scsumacct -- displays daily and monthly accounting information

SYNOPSIS

/opt/nec/nqsv/bin/scsumacct [-btv][-s *day_str* | -S *mon_str*][-r *database*][-y *database*]

day_str: MM[/DD[/YY]]

mon_str: MM[/YY]

/opt/nec/nqsv/bin/scsumacct -N

DESCRIPTION

scsumacct outputs the accounting summary information for the specified day or month. Display contents consist of summary information of the request accounting and the resource reservation accounting information. The total of all data for the specified day or month is displayed on the last line.

The following are output as default.

<u>SUMMARY OF REQUEST ACCOUNTING</u>

REQUESTS	Total number of requests
JOBS	Total number of jobs
EHOST NUM	Total number of execution hosts
ELAPSE(H)	Total time of the elapsed time of jobs [HOUR]
CPU(S)	CPU consumption time (system + user) [SECS]
MEM	Total memory consumption (KB * MIN)

<u>SUMMARY OF RESERVATION ACCOUNTING</u>

RESERVES	Total number of reservations
EHOST NUM	Total number of execution hosts
RSVTIME	Total time of the resource reservation [HOUR]
USETIME	Total time of the used reservation time [HOUR]

The following options can also be specified.

-N Shows the scsumacct version information. This option is given priority even when other options are specified, and they are disregarded.

-s *MM/DD*[/ *YY*]

Specifies a day for which summary information is to be output. (If -s and -S are not both specified, summary information for the current day is output.)

-S *MM*[/ *YY*]

Specifies a month for which summary information is to be output.

-t Displays the user CPU time and the system CPU separately.

-b Displays the estimated billings and the actual billings.

-r *database*

Specifies a target data base of the request accounting for calculation. If this option is not specified, the default database (/var/opt/nec/nqsv/asv/racct and racct_tally) is assigned.

-y *database*

Specifies a target database of the reservation accounting for calculation. If this option is not specified, the default database (/var/opt/nec/nqsv/asv/rsvacct) is assigned.

If a future day is specified, an error message may be output.

SEE ALSO

sceditacct(1M-N)

REMARKS

When the accumulated accounting data is remarkably large, it may take time to execute this command. Therefore, it is recommended to delete unnecessary data using the sceditacct command before executing this command.

2.8. scturnacct(1M)

NAME

scturnacct - saves each type of accounting data

SYNOPSIS

/opt/nec/nqsv/bin/scturnacct [-N]

DESCRIPTION

scturnacct is a command used to save the accounting files.

If each accounting file exceeds the specified size, the specified number of generations of the accounting data is saved.

The file size and the number of generations to be saved are specified in /etc/opt/nec/nqsv/acmd.conf. The default values are 500 MB and 10 generations each.

TURN_SIZE=500	The file size at which each accounting file begins saving. (Unit: MB)
TURN_SAVE=10	Number of generations to be saved.

When the -N option is specified, the version of scturnacct is displayed.

SEE ALSO

None.

2.9. smgr(1M)

NAME

smgr -- Configuration and Operation Management of JobManipulator

SYNOPSIS

smgr [**-P** *privilege*] [**-V**] [**-h** *server_host*] [**-s** *scheduler_id*] [--help] [--version]

DESCRIPTION

The *smgr*(1M) command is the program used by the system manager and system operators to control JobManipulator on the *server_host*.

The following options can be specified.

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	(Manager privilege)
o	(Operator privilege)
g	(Group manager privilege)
s	(Special user privilege)
u	(User privilege)

Among access privileges, Manager privilege is the highest and User privilege is the lowest. It is possible to refer the request information of other user by having the operator privilege or higher.

When the **-P** option is not specified, access privilege for command execution is set by Environment Variable **JM_PRIV**. If this variable is not set, user privilege is set up.

-V Display only the version of the command and JobManipulator API.

-h *server_host*

Specify the JobManipulator server host. If this option is omitted, the default JobManipulator server host will be set. (The default host name is specified in

/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf). If the default host is not specified, a local host will be assumed.

-s *scheduler_id*

Specify the scheduler ID of JobManipulator. If this option is omitted, the scheduler ID will be the value of Environment Variable **JM_SCHED_ID**. If this variable is not set, the scheduler ID specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf is used. If none of the above is specified, 0 will be the scheduler ID.

--**help** Display only the usage of smgr.

--**version**

Display only the version of the command and JobManipulator API.

INPUT COMMAND

Activating Command *smgr*, *smgr* returns Prompt **Smgr:** and then enables to input the subcommands described below. Execute Command **exit** to terminate Command *smgr*.

Input a line continuation character (backslash), before line feed when inputting subcommands in several lines.

EDIT COMMANDS IN INPUT COMMAND

The following edit commands can be used in **Smgr:** prompt lines.

Ctrl-f

Moves one character in the forward direction.

Ctrl-b

Moves back one character in the reverse direction.

Ctrl-h

Deletes characters in the front.

Ctrl-p

Displays an input command one before.

Ctrl-n

Displays an input command one after.

<tab>

Displays or complements a subcommand that matches a character string that is input.

ACCESS PRIVILEGE

Four types of access privilege are available with the *smgr* subsystem. The Manager privilege permits use of all commands, but the Operator privilege permits use of only limited commands. User privilege, Special user privilege and Group manager privilege are allowed to use only commands **exit**, **help**, and **list**.

PAGER FUNCTION

Programs for paging can be specified with PAGER Environment Variables. Output of Subcommand **help** can be used at this time as standard input of programs. For example, by setting

```
PAGER=/usr/bin/more
```

output of **help** will be paged and displayed.
If a specified program cannot be executed, the program will be output as is.

SUBCOMMANDS

Each subcommand of *smgr* is described here. Words within square brackets "[]", are optional.

add complex_queue queue = (*queue-name* [,*queue-name...*]) *complex-queue-name*

The executing queue specified by *queue-name* is added to the complex queue specified by *complex-queue-name*.

Manager privilege is needed.

create complex_queue = *complex-queue-name* **queue** = (*queue-name* [,*queue-name...*])

Create the complex queue specified by *complex-queue-name*. Specify the executing queue to *queue-name*. This queue belongs to the complex queue which will be created.

Manager privilege is needed.

create eco_schedule starttime = *start_time* **endtime** = *end_time* **hostname** = *host_name*

Create power-saving schedule.

The start time of the schedule is specified to *start_time*. If a time before the current time is specified, it will not be created. The end time of the schedule is specified to *end_time*. The interval from *start_time* to *end_time* must be larger than or equal to the sum of the margin for stopping a node, the minimum period of DC power off and the margin for starting a node. The target host is specified to *host_name* with the host name.

Operator privilege is needed.

The form for *start_time and end_time are* as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The two digits are expressed as follows:

CC: First two digits of year
YY: Last two digits of year
MM: Month (01-12)
DD: Date (01-31)
hh: Hours (00-23)
mm: Minutes (00-59)
SS: Seconds (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.
If this option is omitted, the current time will be used.
The beginning time is rounded down to the nearest minute.

create resource_reservation starttime = *start_time* **blocktime** = *block_time* **queue** = *queue-name* [**name** = *resource_reservation_name*] [**hostnum** = *hostnum*] [**cpunum** = *cpunum*] [**group** = *group_name*]

create resource_reservation starttime = *start_time* **blocktime** = *block_time* [**name** = *resource_reservation_name*] **hostname** = *host_name* | **node_group** = *node_group_name*

Reserve the resource section. Specifying with a queue, with a hostname, hostnames or a node group can be accepted for reservation. The hostname and node group cannot be specified together.

In case of the queue specification, the host bound to the execution queue specified in *queue-name* will be target of the Resource Reservation Section.

The number of hosts to be reserved is specified to *hostnum*. If omitted, it will be the number of all hosts bound to the queue. Which group can use this reservation is specified to *group-name*.
The number of CPUs per host is specified to *cpunum*. If omitted, it will be the number of all CPUs.

In case of the **hostname** specification, all of the number of CPU specified in *host_name* will be target. The following can be specified in *host_name*

- hostname
- (hostname,hostname,hostname,......)
- ALL_HOSTS (all execution hosts will be the targets)

In the case of the **node_group** specification, the Resource Reservation Section is created for all the execution nodes belonging to the **node_group** specified in *node_group_name*. Note that the Resource Reservation Section will not be created automatically for the execution host that is added to the node group afterward. Additionally, even if the execution host is deleted from the node group, the deleted execution host will not be deleted automatically from the Resource Reservation Section.

The start time of the Resource Reservation Section is specified to *start_time*. If a time before the current time is specified, it will not be reserved. The length of the section is specified to *block_time* by seconds. A name of resource reservation can be specified in *resource_reservation_name*.

The form for *start_time is* as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The two digits are expressed as follows:

CC: First two digits of year
YY: Last two digits of year
MM: Month (01-12)

DD: Date (01-31)

hh: Hours (00-23)

mm: Minutes (00-59)

SS: Seconds (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.
If this option is omitted, the current time will be used.
The beginning time is rounded down to the nearest minute.

[Example]
In case of starting at "17:00", "240 seconds (4 minutes)" for reservation time, "execque1" of the queue name and "1" CPU.

```
create resource_reservation starttime=1700 blocktime=240 queue=execque1 cpunum=1
```

Operator privilege is needed.

create resource_reservation starttime = *start_time* **blocktime** = *block_time* **queue** = *queue-name* **template** = *template_name* **machinenum** = *machine_num* [**name** = *resource_reservation_name*] [**group** = *group_name*]

Reserve the resource section specifying template.

The start time of the Resource Reservation Section is specified to *start_time*. If a time before the current time is specified, it will not be reserved. The length of the section is specified to *block_time* by seconds.

The host bound to the execution queue specified in *queue-name* will be target of the Resource Reservation Section.

Template name is specified to *template_name.*

The number of machine that is started with template *template_name* is specified to machine_num with its value 1 to 10240.

A name of resource reservation can be specified in *resource_reservation_name*.

Which group can use this reservation is specified to *group-name*.

The form for *start_time is* as follows:

[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]

The two digits are expressed as follows:

CC: First two digits of year
YY: Last two digits of year
MM: Month (01-12)
DD: Date (01-31)
hh: Hours (00-23)
mm: Minutes (00-59)
SS: Seconds (00-61)

If the time specified by *hhmm* has already passed, it is set up in the next day.
If this option is omitted, the current time will be used.
The beginning time is rounded down to the nearest minute.

[Example]
In case of starting at "17:00", "240 seconds (4 minutes)" for reservation time, "execque1" of the queue name, template is template1 and "10" CPU.

```
create resource_reservation starttime=1700 blocktime=240 queue=execque1
template=template1 machinenum=10
```

Operator privilege is needed.

delete complex_queue = *complex-queue-name*

Delete the complex queue specified by *complex-queue-name*.

Manager privilege is needed.

delete eco_schedule = *eco_schedule_id*

Delete a power-saving schedule.

The ID of power-saving schedule is specified to *eco_schedule_id*.

Operator privilege is needed.

delete global_group_cpu_run_limit groups = *group-name-list*

Delete limits of the number of CPUs that can be used for each group specified in *group-name-list* in the scheduler.

The following can be specified in *group-name-list*

- group name
- (group name,group name,......)

Operator privilege is needed.

delete global_group_run_limit groups = *group-name-list*

Delete request run limit for each group specified in *group-name-list* in the scheduler.

The following can be specified in *group-name-list*

- group name
- (group name,group name,......)

Operator privilege is needed.

delete global_group_ve_run_limit groups = *group-name-list*

Delete limits of the number of VEs that can be used for each group specified in *group-name-list* in the scheduler.

The following can be specified in *group-name-list*

- group name
- (group name,group name,group name,......)

Operator privilege is needed.

delete global_user_cpu_run_limit users = *user-name-list*

Delete limits of the number of CPUs that can be used for each user specified in *user-name-list* in the scheduler.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete global_user_run_limit users = *user-name-list*

Delete request run limit for each user specified in *user-name-list* in the scheduler.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete global_user_ve_run_limit users = *user-name-list*

Delete limits of the number of VEs that can be used for each user specified in *user-name-list* in the scheduler.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete queue group_cpu_run_limit groups = *group-name-list queue-name*

Delete limits of the number of CPUs that can be used for each group specified in *group-name-list* in the *queue-name*.

The following can be specified in *group-name-list*

- group name
- (group name,group name,group name,......)

Operator privilege is needed.

delete queue group_run_limit groups = *group-name-list queue-name*

Delete request run limit for each group specified in *group-name-list* in the *queue-name*.

The following can be specified in *group-name-list*

- group name
- (group name,group name,group name,......)

Operator privilege is needed.

delete queue group_ve_run_limit groups = *group-name-list queue-name*

Delete limits of the number of VEs that can be used for each group specified in *group-name-list* in the *queue-name*.

The following can be specified in *group-name-list*

- group name
- (group name,group name,group name,......)

Operator privilege is needed.

delete queue user_cpu_run_limit users = *user-name-list queue-name*

Delete limits of the number of CPUs that can be used for each user specified in *user-name-list* in the *queue-name*.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete queue user_run_limit users = *user-name-list queue-name*

Delete request run limit for each user specified in *user-name-list* in the *queue-name*.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete queue user_ve_run_limit users = *user-name-list queue-name*

Delete limits of the number of VEs that can be used for each user specified in *user-name-list* in the *queue-name*.

The following can be specified in *user-name-list*

- user name
- (user name, user name, user name,......)

Operator privilege is needed.

delete resource_reservation = *resource_reservation_id* [**force**]

Delete the Resource Reservation Section. The reservation ID is specified to *resource_reservation_id*. When specified **force**, deletes the Resource Reservation Section even a request exists in the Resource Reservation Section. At this time, request will be deleted too.

Operator privilege is needed. In case a request exists in the Resource Reservation Section, manager privilege is required to delete request by **force** specification.

exit

Exits Command *smgr*.

help [*command*]

Displays help information. Using Command **help** without an argument will display information on commands that can be used. By using Command **help** with an argument, information of specified commands will be displayed. Commands can be specified partially as long as commands are unique. Specification of a command argument in more detail will yield more detailed information that will be received.

remove complex_queue queue =(*queue-name* [*,queue-name...*]) *complex-queue-name*

Delete the executing queue specified by *queue-name* from the complex queue specified by *complex-queue-name*.

Manager privilege is needed.

resume request = *request_id*

Send a resume request for the request specified with *request_id*. The format of *request_id* is as below.

request_number[.*batch_server*]

request_number

Sequence number of the request ID. For a parametric request, specify [] following the sequence number. For a sub-request, specify [*n*] following the sequence number. Specify the sub-request number in *n*.

batch_server

The name of batch server

If *batch_server* is omitted, it will be the name of the batch server of NQSV, which JobManipulator is using.

If a parametric request ID is specified, a resume request will be issued to all sub-requests in the specified parametric request.

The request receiving a resume request becomes the target of rescheduling and will be resumed reaching the scheduled start time.
Only the request in suspended state and suspended by the suspend request subcommand of **smgr** can be resumed by this command.

Operator privilege is needed.

set assign_policy = *assign-policy*

Set request assign policy. The following can be specified in *assign-policy*.

resource_balance

"Resource balanced assignment" is set.

Jobs are assigned to the node whose resource usage is least at the assignment timing.

CPU_concentration

"CPU number concentrated assignment" is set

Jobs are assigned to a node until it is filled to the usable limit of the number of CPUs which is set to each node. It is same for GPU, in other words, the requests requiring GPU are concentrically assigned to a node by GPU until no usable GPU.

The initial value is **resource_balance**.

Operator privilege is needed.

set assign_policy_priority = *priority* **assign_policy** = *assign_policy*

Set the priority or disablement of either following assign policies specified in *assign-policy*.

network_topology

The assignment which is considered about network topology

staging _job

Preferential assignment policy of the node without staging any job whose scheduled start time has been canceled

The following can be set as *priority*.

low	The priority is low.
high	The priority is high.
disable	The assignment policy is disabled.

The initial values are as follows.

network_topology:	**high**
staging_job:	**low**

Operator privilege is needed.

set auto_delete_resource_reservation = on | off

Specifies whether the feature of auto deleting of the reserved section is valid or not. If set to **on**, the feature of auto deleting of the reserved section will be valid. If set to **off**, this feature will be invalid. The initial value is **off**.

Operator privilege is needed.

set cloud_bursting priority_weight_custom_resource = *weight cr_name*

Specifies the weight for custom resource specified by *cr_name* in *weight*. This value is used to calculate the cloud bursting priority. 0 to 10000 can be specified in *weight*.

Operator privilege is needed.

set cloud_bursting priority_weight_resource = *weight*

Specifies the weight for the amount of requested resources (CPU, GPU, Memory) of request in *weight*. This value is used to calculate the cloud bursting priority. 0 to 10000 can be specified in *weight*. The initial value is 0.

Operator privilege is needed.

set cloud_bursting priority_weight_scheduling_priority = *weight*

Specifies the weight for scheduling priority in *weight*. This value is used to calculate the cloud bursting priority. 0 to 10000 can be specified in *weight*. The initial value is 0.

Operator privilege is needed.

set cloud_bursting priority_weight_wait_time = *weight*

Specifies the weight for execution wait time in *weight*. This value is used to calculate the cloud bursting priority. 0 to 10000 can be specified in *weight*. The initial value is 1.

Operator privilege is needed.

set cloud_bursting request_number = *value*

Specifies the upper limit of the requests to be selected for cloud bursting at one time in *value*. 0 to 2147483647 can be specified in *value*. Specifies the weight for execution wait time in *weight*. Since the maximum number of instances defined in the cloud bursting node group is the upper limit of the execution host that can actually be used at one time, specify the number of requests that can be executed within the range of the number of instances. The initial value is 0.

Operator privilege is needed.

set complex_queue run_limit = *run_limit complex-queue-name*

Set the number of requests which can be executed simultaneously in the complex queue specified by *complex-queue-name* to *run-limit*. When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set complex_queue user_assign_limit = *assign_limit complex-queue-name*

Set the number of requests which one user can assign simultaneously in the complex queue specified by *complex-queue-name* to *assign-limit*. When a value of 0 is specified in *assign-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set complex_queue user_run_limit = *run_limit complex-queue-name*

Set the number of requests which one user can execute simultaneously in the complex queue specified by *complex-queue-name* to *run-limit*. When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set dc-off_limit = *number_of_times*

Set DC Power Off Limit by which the number of times of stopping a node by Dynamic Power-saving function per day is limited.

The range of value is 1-200.

The initial value is 5.

Operator privilege is needed.

set device_group_topology = on | off

Set scheduling considering HCA device group topology.

on : The jobs are assigned within the same HCA device node group as much as possible in consideration of the HCA device group topology.

off : it does not consider the HCA Device Group Topology, but assigns jobs in consideration of the concentration or distribution of CPU Resources Policy which is a lower-level policy.

For example, in the case of the concentration policy, the number of available VE nodes in the VI is limited and the job is assigned.

The initial value is **off**.

Operator privilege is needed.

set dynamic_dc_control = on | off

Set whether to start Dynamic Dc Control or stop.

on Start Dynamic Dc Control

off Stop Dynamic Dc Control

When changing it from **on** to **off**, the nodes bound with the queue bound with JobManipulator except the node with HW failure and the node stopped according to Eco Schedule will be started immediately, and then Dynamic Dc Control is stopped.

The initial value is **off**.

Operator privilege is needed.

set escalation_interval = *escalation-interval*

Set the escalation interval by a unit of cell size to *escalation-interval*. The value greater or equal to 1 can be specified to *escalation-interval* (map width/cell size). The initial value is 1.

Operator privilege is needed.

set estimated_dc-off_time = *seconds*

Set the Estimated DC-OFF Time (threshold) to *seconds*. The node can be stopped by Dynamic Power-saving function when it is possible to stop for not less than a certain period of time (Estimated DC-OFF Time), in other words, when there is no job scheduled from current to Estimated DC-OFF Time later.

The unit is second.

The range of value is 2-2147483647. It must be equal to or larger than the sum of the margin for stopping a node and the margin for starting a node

The initial value is 3600.

Operator privilege is needed.

set executionhost cpunum_limit_ratio = *limit-ratio execution-host-name* | **node_group** = *node_group_name*

Set the limit of simultaneous usable number of CPUs on the execution host specified in *execution-host-name* or the execution host in the node group specified in *node_group_name*. Specify the ratio against total number of CPUs on the execution host to *limit-ratio*. For example, all CPUs on the machine are used when 1 (= 100%) is specified to *limit-ratio*. Setting 0 to *limit-ratio*, this limit will be invalid and the number of CPUs is not checked. The default is 1. Note that the result of converting the value specified in the *limit-ratio* will be rounded down to a whole number of CPUs when a fraction arises. For example, the existence one CPU is assumed if the converted value is 1.2.

Operator privilege is needed.

set executionhost rsg_cpunum_limit_ratio = *limit-ratio* **rsg_number** = *rsg-number execution-host-name* | **node_group** = *node_group_name*

Set the limit of simultaneous usable number of CPUs for RSG specified in *rsg-number* on the execution host specified in *execution-host-name* or the execution host in the node group specified in *node_group_name*. Specify the ratio against the number of CPU specified per RSG (= Icpu) to *limit-ratio*. For example, all of Icpu are used when 1 (= 100%) is specified to *limit-ratio*. By setting 0 to *limit-ratio*, this limit will be invalid and the number of CPUs is not checked. The initial value is 1. Note that the result of converting the value specified in the *limit-ratio* will be rounded down to a whole number of CPUs when a fraction arises. For example, if the converted value is 1.2, the number of CPUs is 1.

Operator privilege is needed.

set executionhost memsz_limit_ratio = *limit-ratio execution-host-name* | **node_group** = *node_group_name*

Set the limit of simultaneous usable memory size on the execution host specified in *execution-host-name* or the execution host in the node group specified in *node_group_name*. Specify the ratio against total memory size on the execution host to *limit-ratio*. For example, all memory on the machine are used when 1 (= 100%) is specified to *limit-ratio*. Setting 0 to *limit-ratio*, this limit will be invalid and memory size is not checked. The initial value is invalid.

Operator privilege is needed.

set executionhost rsg_memsz_limit_ratio = *limit-ratio* **rsg_number** = *rsg-number execution-host-name* | **node_group** = *node_group_name*

Set the limit of simultaneous usable memory size for *rsg-number* on the execution host specified in *execution-host-name* or the execution host in the node group specified in *node_group_name*.
Specify the ratio against the large page memory size specified per RSG (Imem) to *limit-ratio*. For example, all of Imem are used when 1 (= 100%) is specified to *limit-ratio*. Setting 0 to *limit-ratio*, this limit will be invalid and memory size is not checked. The initial value is invalid.

Operator privilege is needed.

set forced_rescheduling = **on** | **off** [cloud]

Set the setting of forced rescheduling. If cloud is not specified, set forced rescheduling for regular requests that are not cloud bursted. If cloud is specified, set forced rescheduling for cloud-bursted requests.

on The running request is rescheduled when this request is stalled due to node trouble. The request is rerun.

off The running request is not rescheduled until it is recovered from stall state when this request is stalled due to node trouble.

The initial value is **off**.

Operator privilege is needed.

set global_group_cpu_run_limit = *run-limit* [**groups** = *group-name-list*]

Set the limits of the number of CPUs that can be used in the scheduler to *run-limit* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list*.

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_group_run_limit = *run-limit* [**groups** = *group-name-list*]

Set the number of requests which one group can execute simultaneously in the scheduler to *run-limit* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list* .

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_group_ve_run_limit = *run-limit* [**groups** = *group-name-list*]

Set the limits of the number of VEs that can be used in the scheduler to *run-limit* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list* .

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_run_limit = *run-limit*

Set the number of requests which can be executed simultaneously in the scheduler.
When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_user_assign_limit = *assign-limit*

Set the number of requests which one user can assign simultaneously in the scheduler.
When a value of 0 is specified in *assign-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_user_cpu_run_limit = *run-limit* [**users** = *user-name-list*]

Set the limits of the number of CPUs that can be used in the scheduler to *run-limit* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list.*

- user name

- (user name,user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_user_run_limit = *run-limit* [**users** = *user-name-list*]

Set the number of requests which one user can execute simultaneously in the scheduler to *run-limit* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list*.

- user name
- (user name,user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set global_user_ve_run_limit = *run-limit* [**users** = *user-name-list*]

Set the limits of the number of VEs that can be used in the scheduler to *run-limit* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list*.

- user name
- (user name,user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set half_reduce_period = *half-reduce-period*

Set the half-life decay length to *half-reduce-period*. The initial value is 604800 seconds.

Operator privilege is needed.

set interruption_method = *method*

Set the interruption method for an interrupted request. This option is valid only in case of specifying **current** in the *interrupt_to_where* subcommand. The followings can be specified in *method*.

suspend

Suspend the request.

rerun

Rerun the request.

This option forcibly reruns the request even if rerun is not allowed for the target request.

The initial value is **suspend.**

Operator privilege is needed.

set interrupt_to_where = *where*

Set where high-priority requests interrupt to. The followings can be specified in *where*.

next_run

High-priority requests interrupt to the head of the assigned requests waiting for execution in order not to affect the currently executing requests.

current

High-priority requests interrupt to the current time. When a request is in execution, it is handled according to the interruption method of interrupted requests.

The initial value is **next_run**.

Operator privilege is needed.

set interrupt_assign_block = *on* | *off*

Set block of assignment of newly submitted lower-level type requests to the execution host to which the urgent request waiting to be executed is assigned.

on

Prohibits the assignment of newly submitted lower-level type requests to the execution host to which an urgent request waiting to be executed is assigned.

off

Allows the assignment of newly submitted lower-level type requests to the execution host to which an urgent request waiting to be executed is assigned.

The initial value is **on.**

Operator privilege is needed.

set keep_forward_schedule = *second*

Set the time to determine to maintain the schedules of which requests when node problem (HW failure or only unlink down of the job server) occurred with specifying a time of period in *second* by second. The schedules of the requests whose scheduled start time is [time of HW failure occurrence + *second*] or a later one are maintained.

When 0 is specified in *second*, the schedule is not maintained.

The initial value is 0.

Operator privilege is needed.

set logfile [**file** = *logfile-name*] [**level** = *log-level*] [**size** = (*logfile-size*)] [**save** = *save-num*]

Set JobManipulator log files.

Set a log file name in *logfile-name*. Set the output level of log files in *log-level*. A value from 1 to 5 can be specified. Set a high-limit size of log files in *logfile-size*. Values in 1024 - 2^31-1 bytes can be specified. And when setting the high limit size of log files, unit can be specified by "B, KB, MB and GB". A unit can be specified in size specification. If a unit is not specified, the unit will be byte. 0 indicates "unlimited". Reaching the high-limit value for size, a file will automatically be reset and the previous log file will be saved in the backup file. Set the number of backup files to be saved in *save_num*.

Operator privilege is needed.

set mapsize [**sched_time** = *sched_time*] [**sched_interval** = *sched_interval*]

Set the map width and cell size. The map width is specified with the unit of seconds to *sched_time*. The cell size is specified with the unit of seconds to *sched_interval*. When the map width or cell size is changed, the requests are all rescheduled except the running requests. The initial value of **sched_time** is 86400 seconds. The available value range of **sched_time** is from 3 to 2147483647 seconds. The initial value of **sched_interval** is 60 seconds. The available value range of **sched_interval** is from 2 to 2147483646 seconds. A larger value than **sched_interval** must be set to **sched_time**.

Operator privilege is needed.

set max_operation_hosts = *hostnum*

Set the maximum number of operation nodes in Dynamic DC Control to *hostnum*. The DC power supplies of a part of nodes are turned off so that the nodes in operation are not more than the maximum number of operation nodes.

The range of the value is 0-10240. The initial value is 10240.

Operator privilege is needed.

set min_idle_time = *seconds*

Set the Minimum Idle Time to *seconds*. The node can be stopped by Dynamic Power-saving function after the elapse of this period of time from the following time in order to avoid stopping the node right after it becomes the target of operation or the job in it is finished.

The start of this period is the latest one of following time.

- When there is no running job in the node.
- When the node is started.
- When JobManipulator is started.
- When you enable the Dynamic Power-saving function by smgr.
- When you bind the Job Server to a queue which is bound with JobManipulator.
- When you bind JobManipulator to the queue with the node bound.

The unit is second.

The range of value is 0-2147483647.

The initial value is 300.

Operator privilege is needed.

set overtake allow_small_request = *on* |*off*

Set whether or not to use the overtaking assignment function for small requests.

on

Even if a request with a scheduling priority value that exceeds the threshold for overtaking control exists, a request with a scheduling priority value that does not exceed the threshold for overtaking control and that fits in the free space in front of the request can be assigned.

off

If a request with a scheduling priority value that exceeds the threshold for overtaking control exists, a request with a scheduling priority value that does not exceed the threshold for overtaking control cannot be assigned.

The initial value is off.

Operator privilege is needed.

set overtake_priority = *priority queue-type*

Set the value of scheduling priority to *priority* which disables overtake in a queue of type specified in *queue-type*. The followings can be specified in *queue-type*.

normal normal type

special	special type

Operator privilege is needed.

set peak_cut_urgency = wait_run | right_now

Set whether to power off a node immediately when the node will be powered off by the adjusting maximum operation function of Dynamic DC Control.

wait_run

The node is powered off after the running request is finished.

right_now

The running request is rerun, the assigned requests are rescheduled and then the node is powered off immediately.

The initial value is **wait_run**.

Operator privilege is needed.

set priority baseup_interrupted = *baseup* **processing_pattern** = *pattern*

Set the base-up value for a request suspended by urgent request in "baseup". This value is used to calculate scheduling priority. The following can be specified in *pattern*, and you can set the base-up value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority baseup_reschedule = *baseup* **processing_pattern** = *pattern*

Set the base-up value for a rescheduled request in *baseup*. This value is used to calculate scheduling priority. The following can be specified in *pattern*, and you can set the base-up value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_request_priority = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of request priority to *weight*. This value is used to calculate scheduling priority. The following can be specified in *pattern*, and you can set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_cpu_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of number of CPUs to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and you can set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_elapse_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of elapse time to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and you can set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_memory_size = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of memory size to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_ve_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of number of VEs to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and you can set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority pastusage_weight_userrank_request_priority = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of request priority to *weight*. This value is used to calculate User Rank. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority pastusage_weight_userrank_cpu_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of number of CPUs to *weight*. This value is used to calculate User Rank. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority pastusage_weight_userrank_elapse_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of elapse time to *weight*. This value is used to calculate User Rank. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority pastusage_weight_userrank_memory_size = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of memory size to *weight.* This value is used to calculate User Rank. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority pastusage_weight_userrank_ve_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for usage data of number of VEs to *weight.* This value is used to calculate User Rank. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority weight_user_rank = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for user rank to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_user_share = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for user share to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_request_priority = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for request priority to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 1.

Operator privilege is needed.

set priority weight_cpu_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for the declared number of CPUs to *weight*. This value is used to calculate scheduling priority. The following can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_elapse_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for declared ELAPSED time to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_memory_size = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for declared memory size to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_job_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for the number of jobs to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_ve_number = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for the declared number of VEs to *weight*. This value is used to calculate scheduling priority. The following can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_restart_wait_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for wait time for restart to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_run_wait_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for wait time for execution after submission to the queue to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set priority weight_assign_wait_time = *weight* **processing_pattern** = *pattern*

Set the weight coefficient value for wait time for execution after the request becomes assignable to *weight*. This value is used to calculate scheduling priority. The followings can be specified in *pattern*, and it is possible to set the weight coefficient value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set provisioning_start_retry_time = *value*

Set waiting time of rescheduling of starting of virtual machine (VM),　baremetal server or container which fails in starting by the provisioning environment to *value*. The value is set by second. After changing value, changed value is valid even on host that waiting reschedule.

The initial value is 0 (0 means immediate rescheduling).

Operator privilege is needed.

set queue assign_policy = *assign-policy queue-name*

Set request assign policy to the queue specified with *queue-name*. Following assign policies can be specified to *assign-policy*

resource_balance

Jobs are assigned to the node whose resource usage is least at the assignment timing.

CPU_concentration

Jobs are assigned to a node until it is filled to the usable limit of the number of CPUs which is set to each node.

DEFAULT

Jobs are assigned based on the request assign policy of the scheduler.

The initial value is **DEFAULT**.

Operator privilege is needed.

set queue cloud_bursting display_bursting_priority = **off** | **on** *queue-name*

Specify whether to show of hide the cloud bursting priority for requests submitted to the queue specified by *queue-name*.

off Do not display the cloud bursting priority of the request in the execution result of sstat [-R] -f. Requests submitted to the cloud bursting queue will display the cloud bursting priority even if this setting is off.

on Display the cloud bursting priority of the request in the execution result of sstat [-R] -f. Display the cloud bursting priority of requests submitted to the queue, even if cloud bursting is not possible on the queue specified by *queue-name*.

The initial value is **off**.

Operator privilege is needed.

set queue cloud_bursting occupancy_ratio = *value queue-name*

Specify the scheduler map occupancy threshold that determines whether cloud bursting is performed on the queue specified by *queue-name*. If the value specified in *value* is exceeded, requests submitted to the queue specified by *queue-name* can be cloud bursted. A decimal number in the range of 0 to 1 can be specified. If 0 is specified, requests submitted to the queue can always be cloud bursted. If 1 is specified, requests submitted to the queue will not always be cloud bursted.

The initial value is 1.

Operator privilege is needed.

set queue deadline mode = **off** | **on** *queue-name*

Specify the behavior of deadline scheduling of the queue specified with *queue-name*.

off Deadline scheduling is not performed.

on Deadline scheduling is performed.

The initial value is **off**.

Operator privilege is needed.

set queue deadline reduce rate = (*R1*, *R2*, *R3*) **time** = (*T1*, *T2*, *T3*, *T4*) *queue-name*

Set the parameter for tuning reduce rate of the queue specified with *queue-name*. Following values can be specified to the parameter for tuning reduce rate.

R1 Minimum reduce rate. (The range is from 0 to 1.0 and the initial value is 1.0.)

R2 Ontime reduce rate. (The range is from 0 to 1.0 and the initial value is 1.0.)

R3 Maximum reduce rate. (The range is from 0 to 1.0 and the initial value is 1.0.)

T1 End time of rate decrease. (It is an integer equal or greater than 0 and the initial value is 0.)

T2 Start time of rate decrease. (It is an integer equal or greater than 0 and the initial value is 0.)

T3 Start time of rate increase. (It is an integer equal or greater than 0 and the initial value is 0.)

T4 End time of rate decrease. (It is an integer equal or greater than 0 and the initial value is 0.)

Operator privilege is needed.

set queue dynamic_jsv_priority control = off | on *queue-name*

Specify the behavior of Dynamic JSV Priority of the queue specified with *queue-name*.

off Dynamic JSV Priority is not performed.

on Dynamic JSV Priority is performed.

The initial value is **off.**

Operator privilege is needed.

set queue dynamic_jsv_priority item = *item-name* [**max_item_priority** = *priority_value*] [**max_item_value** = *item_value*] [**base** = *base_num*] [**order** = **asce** | **desc**] *queue-name*

Specify the configures for Dynamic JSV Priority calculation of the queue specified with *queue-name*.

Set the parameter for element specified with *item-name*.

Set the parameter for max item priority specified with *priority_value*. The initial value is 0.

Set the parameter for max item value specified with *item_value*. The initial value is 1.

Set the parameter for base value specified with *base_num*. The initial value is 0.

Set the parameter for **order** with **asce** for ascending order or **desc** for descending order specified with *base_num*. The initial value is **asce**.

The following strings can be specified for *item-name* as elements to set values.

running_ve_job	Whether a VE jobs are running or not.
remain_elapse	Percentage of Elapse time remaining for running VE jobs.
elapse_time_limit	Declared Elapse time length of running VE jobs.
elapsed_time	Elapsed time of running VE jobs.
ve_job_num	Number of VE jobs running.
Custom Resource name	The amount of resources specified in the custom resource for the running VE job. The consumer of the custom resource must be job.

Operator privilege is needed.

set queue elapse_margin = *elapse_margin queue-name*

Set the value of Elapse Margin with *elapse_margin* to a queue specified by *queue-name*. When the setting is done, requests other than running ones will be reassigned to reflect *elapse_margin*. *elapse_margin* is set in seconds and values in 0 to 2147483647 can be specified.

The initial value of *elapse_margin* is 0.

Operator privilege is needed.

set queue escalation_limit [**jobs** = *number*] [**period** = *second*] [**forward** = *second*] *queue-name*

Set the condition of selecting the target request of Side Escalation to the queue specified by *queue-name*. Following conditions can be specified.

jobs

Max Side Escalation Jobs. Specify the maximum number of jobs with execution host change exceeding which escalation will not be allowed. If the number of the jobs that is going to do Side Escalation, is more than the specified number, Side

Escalation will not be performed. This function is enabled when the scheduled start time of the request is within No Escalation Period specified by period. If 0 is specified, this function is disabled.

period

No Escalation Period. Specify a period from current time by seconds. If the scheduled start time of the request is within it, Side Escalation will not be allowed. If 0 is specified, this function is disabled regardless of the value of jobs.

forward

Min Forward Time. Specify a minimum forward limit by seconds. If the difference of scheduled start time between before escalation and after escalation is less than or equal to the limit time, Side Escalation will not be allowed. If 0 is specified, this function is disabled.

The initial values of above parameter are all **0**.

Operator privilege is needed.

set queue group_cpu_run_limit = *run-limit* [**groups** = *group-name-list*] *queue-name*

Set the limits of the number of CPUs that can be used in the queue specified in *queue-name* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list*.

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue group_run_limit = *run-limit* [**groups** = *group-name-list*] *queue-name*

Set the number of requests which one group can execute simultaneously in the queue specified in *queue-name* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list*.

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue group_ve_run_limit = *run-limit* [**groups** = *group-name-list*] *queue-name*

Set the limits of the number of VEs that can be used in the queue specified in *queue-name* for all groups or each group. If **groups** isn't specified, it is for all groups. If **groups** is specified, it is for the specified group/groups.

The following can be specified in *group-name-list*.

- group name
- (group name,group name,group name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue interrupt_to_where = *where queue-name*

Set where high-priority requests interrupt to for the queue specified with *queue-name*. Following value can be specified to *where*.

next_run

High-priority requests interrupt to the head of the assigned requests waiting for execution in order not to affect the currently executing requests.

current

High-priority requests interrupt to the current time. When a request is in execution, it is handled according to the suspend method of interrupted requests.

DEFAULT

Requests are assigned based on the policy set to the scheduler.

The initial value is **DEFAULT**.

Operator privilege is needed.

set queue jsv_assign_priority = *priority* **job_server_id** = *job_server_id* | **node_group** = *node_group_name queue*

To schedule the requests submitted in the queue specified in *queue*, set the JSV assignment priority of the job server specified in *job_server_id* or a job server in the node group specified in *node_group_name* to *priority*. A request is preferentially assigned to a job server whose *priority* is higher. For *priority*, specify an integer within the rage of -2147483648 to 2147483647.

For *queue*, specify the queue that is bound with JobManipulator. For *job_server_id*, specify the job server ID of the job server on the attached execution host.

The initial value is 0.

Operator privilege is needed.

set queue mapsize sched_time = *sched_time queue-name*

Specify the map width of the queue specified by *queue-name* to *sched_time*. In case of specifying with exceeding the map width set by scheduler, it will be an error. The queue without setting above will be the map width set by Scheduler.

Operator privilege is needed.

set queue min_operation_hosts = *hostnum queue-name*

Set the minimum number of operation nodes of the queue specified by *queue-name* to *hostnum*. The DC power of a node can be turned off by Dynamic DC Control so as not to make the number of operation nodes of the queue less than this value.

The range of value is 0-10240. The initial value is 10240.

Operator privilege is needed.

set queue network_topology min_group = **off** | **on** *queue-name*

The minimum network topology node group selection is applied to the queue submission request specified by queue-name.

off Disable the minimum network topology node group selection.

on Enable the minimum network topology node group selection.

The initial value is off.

Operator privilege is needed.

set queue resource_reservation pre-margin = *seconds* | **post-margin** = *seconds queue-name*

Set PRE-MARGIN or POST-MARGIN by seconds to the queue specified in *queue-name*.

This value cannot be changed to a larger one, if there is a reservation of the queue. If it is changed to a smaller one, it will be applied to existing reservations of the queue.

The initial value for each margin is 0.

Operator privilege is needed.

set queue retry_time [**staging** = *staging_retry_time*] [**pre-running** = *pre-running_retry_time*] *queue-name*

Set periods of time *staging_retry_time* and *pre-running_retry_time* by seconds to the queue specified by *queue-name*, which a request must wait to be rescheduled when its Stage-in processing or PRE-RUNNING processing is failed respectively, which are performed after assignment. The initial value for each parameter is 0.

Operator privilege is needed.

set queue run_limit = *run-limit queue-name*

Set the number of requests which can be executed simultaneously in a queue specified by *queue-name*.
When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue schedule_type = *backfill* | *fifo queue-name*

Set the schedule type. If *backfill* is specified, then backfill scheduling is used. If *fifo* is specified, then FIFO scheduling is used.

The initial value is *backfill*.

Operator privilege is needed.

set queue type = *queue-type queue-name*

Set the execution queue type. The followings can be specified to *queue-type*.

normal	normal type
special	special type
urgent	urgent type

The initial value is **normal**.

Operator privilege is needed.

set queue user_assign_limit = *assign-limit queue-name*

Set the number of requests which one user can assign simultaneously in the queue specified by *queue-name*.
When a value of 0 is specified in *assign-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue user_cpu_run_limit = *run-limit* [**users** = user*-name-list*] *queue-name*

Set the limits of the number of CPUs that can be used in the queue specified in *queue-name* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list*.

- user name
- (user name, user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue user_run_limit = *run-limit* [**users** = *user-name-list*] *queue-name*

Set the number of requests which one user can execute simultaneously in the queue specified in *queue-name* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list*.

- user name
- (user name, user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue user_ve_run_limit = *run-limit* [**users** = user*-name-list*] *queue-name*

Set the limits of the number of VEs that can be used in the queue specified in *queue-name* for all users or each user. If **users** isn't specified, it is for all users. If **users** is specified, it is for the specified user/users.

The following can be specified in *user-name-list*.

- user name
- (user name, user name, user name,......)

When a value of 0 is specified in *run-limit*, it is assumed to be unlimited. The initial value is 0 (unlimited).

Operator privilege is needed.

set queue wait_stageout = *wait_stageout queue-name*

For *wait_stageout*, specify the estimated time for staging out the request(the stage-out wait time) in the workflow and the request executed on the baremetal server of the queue specified in *queue-name*. The subsequent requests in the same workflow are assigned after the specified estimated time has elapsed from the scheduled execution end time of the request. When the request is in workflow this setting is applied only to following request in workflow.
The initial value is 0.

Operator privilege is needed.

set request baseup_user_definition = *baseup* **processing_pattern** = *pattern request_id*

Set the base-up value for user definition of request specified by *request_id* in *baseup*. This value is used to calculate scheduling priority. The followings are the format of *request_id*.

request_number[.*batch_server*]

request_number

Sequence number of the request ID. For a parametric request, specify [] following the sequence number. For a sub-request, specify [*n*] following the sequence number. Specify the sub-request number in *n*.

batch_server

The name of batch server

When *batch_server* is omitted, it is assumed as the name of NQSV batch server used by the JobManipulator server.

If a parametric request ID is specified, the user-defined base-up value is set to all sub-requests in the specified parametric request.

The followings can be specified in *pattern*, and it is possible to set the base-up value for each of the following processes.

assign	assign process
escalation	escalation process

The initial value is 0.

Operator privilege is needed.

set realtime_scheduling = on | off

Sets immediate request scheduling.

on : Requests are scheduled without waiting for the scheduling interval and start executing immediately.

off : Requests are scheduled at each scheduling interval and start executing.

The initial value is **on**.

Operator privilege is needed.

set realtime_scheduling mode = default | always

Sets the mode for immediate request scheduling.

default : Requests are scheduled immediately during all times except for the times when events such as JSV's LINKDOWN and LINKUP are processed(event time).

always: Requests are scheduled immediately at all times including the event time.

The initial value is **default**.

Operator privilege is needed.

set scheduling_method assign_mode = continue | reset

Sets the assignment mode of the request.

reset: If scheduling takes a long time, the process will be terminated at a certain time.

continue: This process will not be terminated but will continue.

The initial value is **reset**.

Operator privilege is needed.

set scheduling_method non_scheduled_request_cache = **on**|**off**

Sets the caching of non-schedulable requests .

on: function is enabled.

off: function is disabled.

The initial value is **off**.

Operator privilege is needed.

set scheduling_method ve_degradation = continue | **exclude** | **auto**

Set the scheduling method at VE degradation.

If set to **continue** JobManipulator schedules with the change in the number of VE node.

If set to **exclude** JobManipulator excludes VI with degraded VE from the targets of scheduling.

If set to **auto**, when 1 or more VE nodes are degraded, JobManipulator excludes the VI with degraded VE from scheduling. If all degraded VE nodes recovers, JobManipulator restores immediately the VI with the VE nodes to scheduling.

The initial value is **continue**.

Operator privilege is needed.

set stage-in_margin additional_margin_for_escalation = *additional_margin_for_escalation* | **stage-in_threshold** = *stage-in_threshold* | **first_stage-in_time** = *value*

Set an adjusting time of estimated stage-in time for Side Escalation in *additional_margin_for_escalation*.

Set threshold of stage-in in *stage-in_threshold*.

Set a first stage-in time in *value*.

The unit is second.

The initial value is 0.

Operator privilege is needed.

set standard_cpu_number = *cpu-number*

Set the standard number of CPUs in *cpu-number*. The initial value is 8.

Operator privilege is needed.

set standard_job_number = *job-number*

Set the standard number of jobs in *job-number*. The initial value is 10240.

Operator privilege is needed.

set standard_memory_size = (*memory-size*)

Set the standard memory size in *memory-size*. The initial value is 32 GB.

Operator privilege is needed.

set standard_ve_number = *ve-number*

Set the standard number of VEs in *ve-number*. The initial value is 8.

Operator privilege is needed.

set treat_unbookable_request = *method*

Set the method to request in the case that the request cannot be assigned to the specified time. The followings can be specified in *method*.

delete	Delete request
after	Assign request to the earliest executable time (after the specified time).

The initial value is **after**.

Operator privilege is needed.

set use_elapstim_unlimited = on ｜ off

Set **on** or **off** to schedule requests whose *elapstim_req* is set to UNLIMITED. If **on** is set, the requests are scheduled. If **off** is set, the requests are not scheduled. The initial value is off.

Operator privilege is needed.

set use_escalation = forward ｜ all ｜ off

Set behavior of escalation.

forward

Escalation is done only in the same node.

Forward escalation has higher priority than side escalation.

all

Escalation is done regardless whether the node is changed or not.

Forward escalation has higher priority than side escalation.

off

Escalation is not performed.

The initial value is **off**.

Operator privilege is needed.

set use_overtake_priority = on ｜ off *queue-type*

Set valid or invalid of the scheduling priority which disables overtake in a type of queue specified in *queue-type*. If **on**, the scheduling priority which disables overtake is valid. Setting **off**, it will be invalid. The following can be specified in *queue-type*.

normal	normal type

special	special type

The initial value is **off**.

Operator privilege is needed.

start scheduling

Start scheduling.

Operator privilege is needed.

stop scheduling

Stop scheduling.

Operator privilege is needed.

stop waiting_retry executionhost = *hostname*

Dissolve waiting of rescheduling of starting of virtual machine (VM),　baremetal server or container which fails in starting by the provisioning environment. Name of execution host of provisioning is set to hostname.

Operator privilege is needed.

stop waiting_retry request = *request_id*

Make the request specified by *request_id* stop waiting rescheduling. The followings are the format of *request_id*.

request_number[.*batch_server*]

request_number

Sequence number of the request ID. For a parametric request, specify [] following the sequence number. For a sub-request, specify [*n*] following the sequence number. Specify the sub-request number in *n*.

batch_server

The name of batch server

When *batch_server* is omitted, it is assumed as the name of NQSV batch server used by the JobManipulator server.

If a parametric request ID is specified, make all sub-requests in the specified parametric request stop waiting rescheduling.

Operator privilege is needed.

suspend request = *request_id*

Send a request for suspending with elapsed time stopped for the request specified by *request_id*. The format of *request_id* is as below.

request_number[.*batch_server*]

request_number

Sequence number of the request ID. For a parametric request, specify [] following the sequence number. For a sub-request, specify [*n*] following the sequence number. Specify the sub-request number in *n*.

batch_server

The name of batch server. If *batch_server* is omitted, it will be the name of the batch server of NQSV, which JobManipulator is using.

If a parametric request ID is specified, a suspend request that stops the Elapse time for all sub-requests in the specified parametric request is issued.

The elapsed time of the suspended request is stopped and the request will not be rescheduled until it is resumed by smgr command.
Only RUNNING request can be suspended by this command.
CPU for this suspended request is released, however memory is kept because the process is remained. Therefore, it should be ensured that enough memory or swap can be gotten even if other requests are executed while the request is suspended. If the memory becomes insufficient during execution of other requests, it can lead abort of jobs.
The manager can resume the suspended request by the **qsig** command. Because the resumed

request can be executed immediately, there is a possibility that it competes with other running requests for resources.

Operator privilege is needed.

list all [file=*file_name*]

Save information related to the JobManipulator, the execution host, the execution queue, and the node group currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

list executionhost [file=*file_name*]

Save information related to the execution host used by JobManipulator currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

list node_group [file=*file_name*]

Save information related to the node_group used by JobManipulator currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

list priority [file=*file_name*]

Save information related to the scheduling priority used by JobManipulator currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

list queue [file=*file_name*]

Save information related to the execution queue used by JobManipulator currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

list server [file=*file_name*]

Save information related to the JobManipulator currently in operation in a file specified by *file_name* after converting it into the form of subcommand of *smgr*. The JobManipulator environment can be restored easily by reading the created file in *smgr*.

Omitting *file_name*, output will be made to standard output. If a file of the same name already exists, new data will be forcibly overwritten.

ENVIRONMENT VARIABLES

JM_PRIV

Default access privilege for command execution

JM_PRIV_MGR	(Manager privilege)
JM_PRIV_OPE	(Operator privilege)
JM_PRIV_GMGR	(Group manager privilege
JM_PRIV_SPU	(Special user privilege)
JM_PRIV_USR	(User privilege)

JM_SCHED_ID

Default scheduler ID of JobManipulator

SEE ALSO

sstat(1)

FILE

/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf

NOTICE

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

2.10. sushare(1M)

NAME

sushare -- User Share Management Tool

SYNOPSIS

sushare [**-D**] [**-P** *privilege*] [**-V**] [*user-share-config-file*]

sushare -C [**-P** *privilege*] [**-V**] [*user-share-config-file*]

sushare -U [**-P** *privilege*] [**-h** *server_host*] [**-s** *scheduler_id*] [**-V**] [*user-share-config-file*]

sushare -S|**-L** [**-f**|**-b**] [**-P** *privilege*] [**-h** *server_host*] [**-s** *scheduler_id*] [**-V**] [*user-name*[:*Account-code*] ...]

sushare -I -z | [[**-c** *pastusage_cpunum*] [**-v** *pastusage_venum*] [**-e** *pastusage_elapstim*] [**-m** *pastusage_memsz*] [**-r** *pastusage_reqpri*]] [**-P** *privilege*] [**-h** *server_host*] [**-s** *scheduler_id*] [-V] [*user-name*[:*Account-code*]]

sushare -M [**-P** *privilege*] [**-h** *server_host*] [**-s** *scheduler_id*] [**-l**[*logfile_name*]] [**-V**] [*config-file*]

sushare --help

sushare --version

DESCRIPTION

1. If the option **-D** is specified or if none of the options (**-C**, **-U**, **-S**, **-L**, **-I**, **-M**) is specified.

   Reading the user share config file specified by *user-share-config-file*, the user share value of each user will be displayed.
   Specify *user-share-config-file* with an absolute path.
   If *user-share-config-file* is omitted, the object will be /etc/opt/nec/nqsv/jm_sharedb.conf. Refer to the followings for the way to fix the User Share file.
   NQSV User's Guide [JobManipulator]
   Chapter 3 Operation Management, 3.1.3.2 Formula of the Scheduling Priority

2. If the option **-C** is specified.

   Propriety of the user share config file specified by *user-share-config-file* is checked.

Specify *user-share-config-file* with an absolute path.
If *user-share-config-file* is omitted, the object will be /etc/opt/nec/nqsv/jm_sharedb.conf.

The error messages and meanings are as follows.

Error message	**Description**
Double definition	Double definition of a user in multiple groups.
No entry	No user is defined in the group.
Belong to no group	The group is not defined in TOP_GROUP.

3. If the option **-U** is specified.

The user share value of the user share config file specified by *user-share-config-file* is registered to be used for calculation of scheduling priority.
Specify *user-share-config-file* with an absolute path.
If *user-share-config-file* is omitted, the object will be /etc/opt/nec/nqsv/jm_sharedb.conf. The following options can be specified.

-h server_host

Specify the JobManipulator server host. If this option is omitted, the default JobManipulator server host will be the object. (Hostname specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf) If a default host is not specified, a local host will be assumed.

-s scheduler_id

Specify the scheduler ID of JobManipulator. If this option is omitted, the scheduler ID will be the value of Environment Variable **JM_SCHED_ID**. If this variable is not set, the scheduler ID specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf is used. If none of the above is not specified, 0 will be the scheduler ID.

4. If the option **-S** is specified.
Displays the past usage data value (*i.e.* the "merged" usage data when ShareDB merge feature is used.) for the user specified to *user-name*[:*Account-code*]. If *user-name*[:*Account-code*] is omitted, all users' past usage data and total past usage data of each group will be displayed. If **-f** is not specified, "*****" is displayed at the beginning of group name to show that

the displayed data is the total past usage data of a group.

The following options can be specified.

-b

Display original values which do not convert the units.

-f

Display detailed information.

-h *server_host*

Specify the JobManipulator server host. If this option is omitted, the default JobManipulator server host will be set. (Hostname specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf) If a default host is not specified, a local host will be assumed.

-s *scheduler_id*

Specify the scheduler ID of JobManipulator. If this option is omitted, the scheduler ID will be the value of Environment Variable **JM_SCHED_ID**. If this variable is not set, the scheduler ID specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf is used. If none of the above is not specified, 0 will be the scheduler ID.

Following value is accumulated usage data of each user. (including adjusted past usage data by half decay time)

PU_cpunum :Used Number of CPUs * Elapsed Time
PU_venum :Used Number of VEs * Elapsed Time
PU_memsz :Used Memory Size * Elapsed Time
PU_elapstim:Elapsed Time
PU_reqpri :Request Priority * Elapsed Time

An additional word of displayed value like "B","K","M",etc means the unit.
- The unit displayed with PU_cpunum, PU_venum and PU_reqpri.

"K" = 10^3 (1K = 1000)
"M" = 10^6 (1M = 1000K)
"G" = 10^9 (1G = 1000M)
"T" = 10^12 (1T = 1000G)
"P" = 10^15 (1P = 1000T)
"E" = 10^18 (1E = 1000P)

- The unit displayed with PU_memsz.

"B" = Byte
"K" = Kilobytes (1K = 1024B)
"M" = Megabytes (1M = 1024K)
"G" = Gigabytes (1G = 1024M)
"T" = Terabytes (1T = 1024G)
"P" = Petabytes (1P = 1024T)
"E" = Exabytes (1E = 1024P)

5. If the option **-L** is specified.
   Display the local usage data.

6. If the option **-I** is specified.
   Alters the past usage data value for the user specified to *user-name*[:*Account-code*]. If *user-name*[:*Account-code*] is omitted, all user will be the object.

   The following options can be specified.

   -c *pastusage_cpunum*

   Changes the past usage data of the number of CPUs to the value specified by *pastusage_cpunum*.

   -v *pastusage_venum*

   Changes the past usage data of the number of VEs to the value specified by *pastusage_venum*.

   -e *pastusage_elapstim*

Changes the past usage data of the elapse time to the value specified by *pastusage_elapstim*.

-h *server_host*

Specify the JobManipulator server host. If this option is omitted, the default JobManipulator server host will be set. (Hostname specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf) If a default host is not specified, a local host will be assumed.

-m *pastusage_memsz*

Changes the past usage data of memory size to the value specified by *pastusage_memsz*.

-r *pastusage_reqpri*

Changes the past usage data of request priority to the value specified by *pastusage_reqpri*.

-s *scheduler_id*

Specify the scheduler ID of JobManipulator. If this option is omitted, the scheduler ID will be the value of Environment Variable **JM_SCHED_ID**. If this variable is not set, the scheduler ID specified in /etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf is used. If none of the above is not specified, 0 will be the scheduler ID.

-z

Clear all elements of the past usage data to zero.

7. If the option **-M** is specified.
   Merge the ShareDB files according to the configuration specified in *config-file*. If *config-file* is omitted, /etc/opt/nec/nqsv/jm_merge_sharedb.conf will be the target. Operator privilege or higher is required.

   The following options can be specified.

   -l [*logfile_name*]

Output the detailed result of merge processing to the file path which specified by *logfile_name*. If *logfile_name* omitted, the result is outputted to nqs_jmd_sharedb_merge.log on the current directory. Don't insert the whitespace character between **-l** and *logfile_name*.

8. If the option **--help** is specified.

   Display only the usage of sushare.

9. If the option **--version** is specified.

   Display only the version of the command and JobManipulator API.

The following options can be specified in common:

-P *privilege*

Specify access privilege for command execution. The following can be specified in *privilege*.

m	(Manager privilege)
o	(Operator privilege)
s	(Special user privilege)
u	(User privilege)

Among access privileges, Manager privilege is the highest and User privilege is the lowest.

The information on other user's requests can be referred to by using Special user privilege further to user privilege.

When the **-P** option is not specified, access privilege for command execution is set by Environment Variable JM_PRIV. If this variable is not set, user privilege is set up.

-V

Display only the version of the command and JobManipulator API.

ACCESS PRIVILEGE

Operator privilege or higher is required to execute the *sushare* command.

ENVIRONMENT VARIABLES

JM_PRIV

Default access privilege for command execution

JM_PRIV_MGR (Manager privilege)

JM_PRIV_OPE (Operator privilege)

JM_PRIV_SPU (Special user privilege)

JM_PRIV_USR (User privilege)

JM_SCHED_ID

Default scheduler ID of JobManipulator

FILE

/etc/opt/nec/nqsv/jm_sharedb.conf
/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf
/etc/opt/nec/nqsv/jm_merge_sharedb.conf

Appendix. A Update history

18th edition

- Added a sending signal at exceeding the warning value of the limit by elapsed time (qalter(1), qlogin(1), qrsh(1), qstat(1) and qsub(1))
- Fixed qstat(1) --planned-start-time
- Added --mig (qlogin(1), qrsh(1) and qsub(1))
- Added description of the Multi Instances GPU (qstat(1))

19th edition

- Added a Stdout file size limit and Stderr file size limit. (qalter(1), qlogin(1), qrsh(1), qstat(1), qsub(1), qwait(1) and qmgr(1M))

20th edition

- Fixed 4.21 "set queue network_topology min_group" sub-command of the smgr(1M) command.
- Fix the environment variables that cannot be modified by -v option. (qlogin(1), qrsh(1) and qsub(1))

21st edition

- Improved explanation of the -U, -v and -w options of the qlogin(1), qrsh(1) and qsub(1) command.

22nd edition

- Add a explanation of the –long-request-name options of the scacctjob1) and scacctreq(1).

NEC Network Queuing System V (NQSV) User's Guide [Reference]

March 2024　22nd edition

NEC Corporation