NEC Network Queuing System V (NQSV) 利用の手引
［リファレンス編］

輸出する際の注意事項

本製品（ソフトウェアを含む）は、外国為替および外国貿易法で規定される規制貨物（または役務）に該当することがあります。

その場合、日本国外へ輸出する場合には日本国政府の輸出許可が必要です。

なお、輸出許可申請手続きにあたり資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

はしがき

本書は、NEC Network Queuing System V (NQSV)によるジョブ管理のコマンドリファレンスです。

備考

(1)　本書はNEC Network Queuing System V (NQSV) R1.00以降に対応しています。

(2)　本書に説明しているすべての機能はプログラムプロダクトであり、以下のプロダクト名およびプロダクト番号に対応しています。

プロダクト名	型番
NEC Network Queuing System V (NQSV) /ResourceManager	UWAF00 UWHAF00（サポートパック）
NEC Network Queuing System V (NQSV) /JobServer	UWAG00 UWHAG00 (サポートパック)
NEC Network Queuing System V (NQSV) /JobManipulator	UWAH00 UWHAH00 (サポートパック)

(3)　UNIX は The Open Group の登録商標です。

(4)　Intelは、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

(5)　OpenStackは、アメリカ合衆国およびその他の国における OpenStack Foundation の商標です。

(6)　Red Hat OpenStack Platformは、アメリカ合衆国およびその他の国における Red Hat, Inc. の商標です。

(7)　Linux は Linus Torvalds氏の米国およびその他の国における登録商標あるいは商標です。

(8)　Dockerはアメリカ合衆国およびその他の国におけるDocker, Inc.の商標です。

(9)　InfiniBand は、InfiniBand Trade Associationの商標またはサービスマークです。

(10)　Zabbixはラトビア共和国にあるZabbix LLCの商標です。

(11)　その他、記載されている会社名、製品名は、各社の登録商標または商標です。

本書の対象読者

本書は、システム管理者および一般利用者を対象読者として書かれたものです。
とくに一般利用者が構築済みのシステムを利用する際、および、システム管理者がシステムの設定を行う際には、本書は有効な説明書となります。
本書は、対象読者がLinuxの一般操作を知っていることを前提として記述されています。

本書の読み進め方

本書は、次の構成となっています。章ごとに対象読者の範囲は異なっており、表の一番右の列にその範囲を示しています。記載された対象読者の後に(*)がついている章については、該当する読者は必ずお読みください。

章	タイトル	内容	対象読者
1	ユーザコマンド	ユーザコマンドの説明	システム管理者(*) 一般利用者(*)
2	システム管理コマンド	システム管理コマンドの説明	システム管理者(*)

関連説明書

NEC Network Queuing System V (NQSV)のマニュアルは以下で構成されています。

マニュアル名称	内容
NEC Network Queuing System V (NQSV) 利用の手引［導入編］	システムの全体像および基本的なシステムの構築方法について説明します。
NEC Network Queuing System V (NQSV) 利用の手引［管理編］	NQSV の管理者が実施する各種設定について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引［操作編］	NQSV の一般利用者が使用する各種機能について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引［リファレンス編］	コマンドリファレンスに関して説明しています。
NEC Network Queuing System V (NQSV) 利用の手引［API 編］	NQSV を操作する C 言語のプログラミングインタフェース (API) について説明しています。
NEC Network Queuing System V (NQSV) 利用の手引［JobManipulator 編］	NQSV のスケジューラコンポーネント JobManipulator に関して説明しています。
NEC Network Queuing System V (NQSV) 利用の手引［アカウンティング・予算管理編］	NQSV のアカウンティング機能に関して説明しています。

表記上の約束

本書では次の表記規則を使用しています。

省略記号 ...	前述の項目を繰り返すことができることを表しています。ユーザは同様の項目を任意の数だけ入力することができます。
縦棒 |	オプションまたは必須の選択項目を分割します。
中かっこ {}	1つを選択しなければならない一連パラメータまたはキーワードを表しています。
角かっこ []	省略可能な一連パラメータまたはキーワードを表しています。

用語定義・略語

用語・略語	説　明
ベクトルエンジン (VE、Vector Engine)	SX-Aurora TSUBASAの中核であり、ベクトル演算を行う部分です。複数のコアと共用メモリが実装されています。PCI Expressカードであり、x86サーバに搭載して使用します。
ベクトルホスト (VH、Vector Host)	ベクトルエンジンを保持するサーバ、つまり、ホストコンピュータを指します。
ベクトルアイランド (VI)	VH1台にVEを1枚ないし複数枚組み込んだ単位を指します。
バッチサーバ（BSV）	バッチサーバホスト上に常駐し、NQSV全体を管理します。
ジョブサーバ（JSV）	実行ホスト上に常駐し、ジョブの実行を管理します。
JobManipulator（JM）	NQSVの提供するスケジューラ機能です。計算リソースの空き容量を管理し、ジョブの開始時刻を割り当てます。
アカウンティングサーバ	アカウント情報の収集・管理、予算管理を行います。
リクエスト	NQSVにおけるユーザジョブの単位であり、１つまたは複数のジョブから構成されたものです。バッチサーバによって管理されます。
ジョブ	ユーザジョブの実行単位であり、ジョブサーバによって管理されます。
論理ホスト	実行ホストの資源を論理的（仮想的）に分割し、ジョブに割り当てたものです
キュー	受け取ったリクエストをプールし、管理する機構です。
BMC	Board Management controllerの略語です。IPMI(Intelligent Platform Management Interface)と呼ばれる業界標準のサーバーマネジメントインタフェースに準拠したサーバー管理を行います。
HCA	Host Channel Adapterの略語です。IBネットワークに接続するためにサーバ側に取り付けるPCIeカードです。
IB	InfiniBandの略語です。
MPI	Message Passing Interfaceの略語です。主にノード間で並列コンピューティングを行うための標準規格です。
NIC	Network Interface Cardの略語です。他ノードと通信するためのハードウェアです。

目 次

は し が き i
用語定義・略語 vi
目 次 vii
第１章 ユーザコマンド 1
1.1 budgetcheck/rbudgetcheck(1) 2
1.2 qalter (1) 4
1.3 qattach (1) 14
1.4 qcat (1) 15
1.5 qcmdconf (1) 18
1.6 qdel (1) 21
1.7 qdetach (1) 23
1.8 qhold (1) 25
1.9 qlogin (1) 27
1.10 qmove (1) 40
1.11 qrerun (1) 43
1.12 qrls (1) 45
1.13 qrsh (1) 47
1.14 qsig (1) 60
1.15 qstat (1) 62
1.16 qsub (1) 103
1.17 qwait (1) 132
1.18 qwait2(1) 136
1.19 scacctjob/racctjob(1) 140
1.20 scacctreq/racctreq(1) 151
1.21 scacctrsv/racctrsv(1) 163
1.22 sstat(1) 168
1.23 wdel (1) 196
1.24 wstart (1) 198
1.25 wstat (1) 200
第２章 システム管理コマンド 202
2.1 budgetedit(1M) 203
2.2 nmapmgr (1M) 206

2.3 nqs_ntfr(1M) 208
2.4 qmgr (1M) 210
2.5 sbuedit(1M) 315
2.6 sceditacct(1M) 319
2.7 scsumacct(1M) 322
2.8 scturnacct(1M) 324
2.9 smgr(1M) 325
2.10 sushare(1M) 369
付録 A 発行履歴 375
A.1 発行履歴一覧表 375
A.2 追加・変更点詳細 375
索 引 377

第1章　　ユーザコマンド

ユーザコマンドは、以下の通りです。

コマンド	説明
budgetcheck(1M)	予算実績表示
qalter(1)	リクエストの属性変更
qattach(1)	リクエストへのアタッチ
qcat(1)	リクエストのエラー／入出力ファイルの表示
qcmdconf(1)	リモート実行型会話機能の管理
qdel(1)	リクエストの削除
qdetach(1)	リクエストへのアタッチ解除
qhold(1)	リクエストのホールド
qlogin(1)	会話リクエストの投入
qmove(1)	リクエストのキュー間移動
qrerun(1)	リクエストの再投入
qrls(1)	リクエストのホールド解除
qrsh(1)	リモート実行型会話リクエスト投入
qsig(1)	リクエストへのシグナル送信
qstat(1)	NQSVシステムの各種状態表示
qsub(1)	リクエストの投入
qwait(1)	リクエスト終了の待ち合わせ
qwait2(1)	複数のリクエスト終了の待ち合わせ
scactjob(1)	ジョブアカウンティング情報の出力
scacctreq(1)	リクエストアカウンティング情報の出力
scacctrsv(1)	予約アカウンティング情報の出力
sstat(1)	JobManipulatorの状態表示
wdel(1)	ワークフローの削除
wstart(1)	ワークフローの実行
wstat(1)	ワークフローの参照

1.1　budgetcheck/rbudgetcheck(1)

名前

budgetcheck,rbudgetcheck ― 予算実績表示

形式

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

機能説明

budgetcheck コマンドを実行した、ユーザ、グループ、およびアカウントコードを取得し、予算実績を表示します。

rbudgetcheck コマンドは、アカウンティングサーバにアクセスして、起動ユーザに関する予算実績を表示します。第一引数にアカウンティングサーバのホスト名を指定し、第二引数以降に budgetcheck コマンドと同様なオプションを指定します。

-N　budgetcheck のバージョン情報を表示します。
他のオプションを指定した場合でも、本オプションが優先され、他のオプションはすべて無視されます。

-P *priv*
コマンド実行時のアクセス権を指定します。priv には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数も設定されていない場合は、一般ユーザ権限となります。環境変数には、以下の指定が可能です。ただし、rbudgetcheck は環境変数は使用できません。

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限

PRIV_SPU　　特別利用者権限
PRIV_USR　　一般利用者権限

-u　　ユーザの予算実績を表示します(デフォルト)。

-g　　グループの予算実績を表示します。

-a　　アカウントコードの予算実績を表示します。

ユーザの予算が超過している場合は終了ステータスとして 1 を返却し、
グループの場合は 2、アカウントコードの場合は 4 を返却します。
複数の項目で超過している場合はこれらの論理和が終了ステータスとして
返却されます。

下記はユーザ、グループ、アカウントコードで実行した例です。

```
# budgetcheck -uga
usr1      11223.41    120.00     12245.00
grp4          0.00      0.00 19874344.00
acct2         0.00      0.00 19874344.00
```

ユーザ、グループ、アカウントコードの順に表示されます。

各レコードは左からターゲット、残高、予定課金額、初期値の順に表示されます。

関連項目

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

1.2　qalter (1)

名前

qalter ― リクエストの属性変更

形式

qalter [*options*] [*job_number*:]*request_identifier*

機能説明

バッチリクエストおよび会話リクエストの資源制限、実行環境等の属性を変更します。
job_number はハイブリッドリクエストについて属性を変更するジョブ番号を指定します。
省略した場合は、そのリクエストに属する全てのジョブについて属性を変更します。
options に指定できるオプションを以下に示します。

-a *date_time*

リクエストの開始日時を変更します。*date_time* の形式は次のとおりです。

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

CC、*YY*、*MM*、*DD*、*hh*、*mm*、*SS* は各々数字２桁で以下の値を示します。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-A *account_string*

リクエストのアカウントコードを *account_string* に変更します。*account_string* に指定できる最大の文字数は、15 バイトまでです。

-d [*hostname*:]*path_name*

リクエストログの出力先を *path_name* で指定したファイルに変更します。
path_name に *hostname*: を指定することによってホストを指定できます。 *hostname* を省略した場合は、リクエストを投入したホストとなります。
path_name には、以下の変換文字が使用できます。

%m　リクエスト ID 内のマシン ID に展開されます
%r　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%%　%に展開されます

-D *log_level*

リクエストログのログレベルを変更します。*log_level* には以下の値が指定可能です。

0 リクエストログは出力されません
1 リクエストに関するログを出力します
2 リクエスト、およびジョブに関するログを出力します

-e [*hostname*:]*path_name*

バッチリクエストの標準エラー出力を、*path_name* で指定したファイルに変更します。*hostname*:を指定することによって、*path_name* のファイルを出力するホストを指定できます。 *hostname* を省略した場合、バッチリクエストを投入したホストに出力します。*path_name* には、以下の変換文字が使用できます。

%m　リクエスト ID 内のマシン ID に展開されます
%r　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%[0*n*]j　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します
(例) %j　: 0, 1, 2...
%03j　: 000, 001, 002...
%%　%に展開されます

-H y | n

ホールドの可否を変更します。y はホールド可能リクエスト、n はホールド不可能リクエストとなります。

-j *join_list*

バッチリクエストの標準出力と標準エラー出力をマージします。マージされた結果は、*join_list* で指定された標準出力または標準エラーに出力します。*join_list* には、次の指定ができます。

e　バッチリクエストの標準出力と標準エラー出力をマージした結果を標準エラーに出力します。
o　バッチリクエストの標準出力と標準エラー出力をマージした結果を標準出力に出力します。

-J y | n

ジョブマイグレーションの可否を変更します。y はジョブマイグレーション可能リクエスト、n はジョブマイグレーション不可能リクエストとなります。

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

リクエストのカーネルパラメータ値を変更します。

カーネルパラメータ値の変更には操作員権限以上のアクセス権が必要です。
kernel_param_list に指定できる項目は以下のとおりです。

nice=*value*　　nice 値

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

リクエストやジョブ、プロセスに対する資源制限の最大値や警告値を変更します。カンマで区切って（スペースなしで）複数の *resource_list* を指定することができます。*resource_list* は以下のとおりです。制限の最大値と警告値の両方を指定する場合 " で囲む必要があります(コマンドラインで指定する場合は､"をエスケープしてください)。警告値を省略すると、現在の値を保持します。但し、最大値を変更した結果、最大値が警告値よりも小さくなる場合は、警告値を最大値と同じ値に変更します。

【リクエスト単位】

elapstim_req=["]*max_limit*[,*warn_limit*"]
経過時間制限値

【ジョブ/論理ホスト単位】

cpunum_job=*max_limit*
CPU 台数制限値
SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 台数となります。

cputim_job=["]*max_limit*[,*warn_limit*"]
CPU 使用時間制限値
SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 使用時間となります。

gpunum_job=*max_limit*
GPU 台数制限値

memsz_job=["]*max_limit*[,*warn_limit*"]
メモリサイズ制限値
SX-Aurora TSUBASA の場合、ジョブの VH 側のメモリサイズとなります。

vmemsz_job=["]*max_limit*[,*warn_limit*"]
仮想メモリサイズ制限値
SX-Aurora TSUBASA の場合、ジョブの VH 側の仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

【プロセス単位】

coresz_prc=["]*max_limit*[,*warn_limit*"]
コアファイルサイズ制限値

cputim_prc=["]*max_limit*[,*warn_limit*"]
CPU 使用時間制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスの CPU 使用時間となります。

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

データサイズ制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスのデータサイズとなります。

filenum_prc=*max_limit*

同時オープンファイル数制限値

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

ファイルサイズ制限値

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

スタックサイズ制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスのスタックサイズとなります。

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

仮想メモリサイズ制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスの仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

制限値はマシンにとって都合のよい値に変換される場合があります。

-m *mail_option*

リクエストの状態の変化についてのメールを送るタイミングを変更します。*mail_options* には、次の指定ができます。

a　リクエストが異常終了したときにメールを送ります。

b　リクエストが開始したときにメールを送ります。

e　リクエストが終了したときにメールを送ります。

n　メールを送りません（他の mail_options と同時に指定した場合、エラーとなります)。

-M *mail_list*

メールを送るユーザリストを変更します。*mail_list* の形式は、次のとおりです。

mail_address[,*mail_address, ...*]

指定できる *mail_list* の最大長は 1023 バイトです。

-N *name*

リクエストの名前を変更します。*name* は 63 バイトまで指定できます。

-o [*hostname*:]*path_name*

バッチリクエストの標準出力を、*path_name* で指定したファイルに変更します。*hostname*:を指定することによって、*path_name* のファイルを出力するホストを指定できます。*hostname* を省略した場合、バッチリクエストを投入したホストに出力します。*path_name* には、以下の変換文字が使用できます。

%m　リクエスト ID 内のマシン ID に展開されます
%r　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%[0*n*]j　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。
(例) %j　：0, 1, 2...
%03j　：000, 001, 002...
%%　%に展開されます

-p *priority*

バッチリクエストの優先度を変更します。
priority は、[-1024...1023]の整数でなければなりません。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m　管理者権限
o　操作員権限
g　グループ管理者権限
s　特別利用者権限
u　一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-r y | n

リクエストのリラン可否を変更します。y はリラン可能リクエスト、 n はリラン不可能リクエストとなります。

-S *path_name*

リクエストのシェルスクリプトを実行するシェルを変更します。*path_name* にシェルの

パス名を指定します。

-T *job_topology*

実行されるジョブの形態を変更します。*job_topology* には、次を指定します。

distrib	実行されるジョブは分散ジョブです。 qsub(1)コマンドの-b オプションで指定されるジョブ数が 2 以上の場合は、指定されたスクリプトがジョブ数分、各ジョブサーバで実行されます。
intmpi	実行されるジョブは intmpi ジョブです。
mvapich	実行されるジョブは mvapich ジョブです。
necmpi	実行されるジョブは necmpi ジョブです。
openmpi	実行されるジョブは openmpi ジョブです。
pltmpi	実行されるジョブは pltmpi ジョブです。

-V

コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

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

リクエストで実行されるスクリプトにおいて、SIGTERM の捕捉を有効にするか無効にするかを指定します。本オプションに *yes* を指定した場合、スクリプトにおいて SIGTERM の補足を有効にします。

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

クラウドのコンピューティング資源へリクエストをバースティングして実行するかどうかを指定します。*yes* を指定した場合は、バースティングします。*no* を指定した場合は、バースティングしません。本オプションが未指定の場合、既定値は *no* です。

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

ユーザ PP スクリプトの *location* で指定した実行位置のスクリプトを *path* に指定したファイルに変更します。*path* の指定がない場合、ユーザ PP スクリプトの設定を削除します。

location には、以下の文字列が使用できます。

prr	実行位置が PRE-RUNNING です。
por	実行位置が POST-RUNNING です。

--warning-signal=elapstim:*signal*

経過時間制限の警告値を超過した場合に送信するシグナルを指定します。*signal* は次の形式で指定できます。

・シグナル名（例：SIGHUP）

・シグナル名から SIG を取り除いたもの（例：HUP）
・シグナル番号（例：1）詳細は signal(7)を参照してください。

【論理ホスト毎の資源制限オプション】

論理ホスト毎の資源制限の最大値や警告値を、以下のオプションで変更することができます。警告値を省略すると、現在の値を保持します。但し、最大値を変更した結果、最大値が警告値よりも小さくなる場合は、警告値を最大値と同じ値に変更します。
詳細な書式については**資源制限値の形式**をご覧ください。

--cpunum-lhost=*max_limit*

論理ホスト毎の CPU 台数制限値を変更します（-l cpunum_job=*max_limit*　CPU 台数制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 台数となります。

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

論理ホスト毎の CPU 使用時間制限値を変更します（-l cputim_job=*max_limit*　CPU 使用時間制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 使用時間となります。

--gpunum-lhost=*max_limit*

論理ホスト毎の GPU 台数制限値を変更します（-l gpunum_job=*max_limit* GPU 台数制限値(ジョブ)に相当します)。

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

論理ホスト毎のメモリサイズ制限値を変更します（-l memsz_job=*max_limit* メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、ジョブの VH 側のメモリサイズとなります。

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

論理ホスト毎の標準エラー出力の結果ファイルサイズ制限値を変更します。

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

論理ホスト毎の標準出力の結果ファイルサイズ制限値を変更します。

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

論理ホスト毎の VE CPU 使用時間制限値を変更します。SX-Aurora TSUBASA の場合、ジョブの VE 側の CPU 使用時間となります。

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

論理ホスト毎の VE メモリサイズ制限値を変更します。SX-Aurora TSUBASA の場合、

ジョブの VE 側のメモリサイズとなります。

--venum-lhost =*max_limit*

論理ホスト毎の VE ノード数制限値を変更します。

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

論理ホスト毎の仮想メモリサイズ制限値を変更します (-l vmemsz_job=*max_limit* 仮想メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、ジョブの側の VH 側の仮想メモリサイズとなります。

VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

【VE ノード毎の資源制限オプション】

VE ノード毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。警告値を省略すると、現在の値を保持します。但し、最大値を変更した結果、最大値が警告値よりも小さくなる場合は、警告値を最大値と同じ値に変更します。

詳細な書式については**資源制限値の形式**をご覧ください。

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

VE ノード毎の CPU 使用時間制限値を変更します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

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

VE ノード毎のメモリサイズ制限値を変更します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

資源制限値の形式

時間制限

時間に関する制限は以下の形式で指定します。

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

無制限の場合は、"unlimited"という文字列を指定します。 指定できる最大値は、2^31-1 秒です。

【指定例】

1234:58:21.29	1234 時間 58 分 21.29 秒
59:01	59 分 1 秒
12345	12345 秒
121.1	121.1 秒

サイズ制限

サイズに関する制限は以下の形式で指定します。

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

units に指定できる単位は以下のとおりです。

b	バイト
kb	キロバイト(1024 b)
mb	メガバイト(1024 kb)
gb	ギガバイト(1024 mb)
tb	テラバイト(1024 gb)
pb	ペタバイト(1024 tb)
eb	エクサバイト(1024 pb)

units を指定しなかった場合はバイトと解釈されます。 無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 バイトです。

【指定例】

1234	1234 バイト
1234kb	1234 キロバイト
1234.5gb	1234.5 ギガバイト

環境変数

NQSV_PRIV または NQSII_PRIV

qalter 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qstat(1) , qsub(1)

注意

属性変更に関しては、以下のことに注意してください。

・リクエストがキュー上で実行中の場合は、変更できない属性があります。
・リクエストが転送キュー上で転送中の場合は、変更できない属性があります。
・投入されているキューに設定されている資源制限値を越えるような値への変更はでき

ません。

・各オプションの区切り文字として使用している文字(-I オプションの " 等)を通常文字として指定する場合、エスケープする必要があります。

1.3　qattach (1)

名前

qattach ― リクエストへのアタッチ

形式

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

機能説明

実行中リクエストのジョブ内に *command* で指定したコマンドを実行します。
request_identifier には対象となるリクエストのリクエスト ID を指定し、*jobnums* には対象となるジョブ番号を指定します。*jobnums* を省略した場合は、リクエスト内の
全ジョブに対して指定したコマンドを実行します。

jobnums の形式は以下のとおりです。

job_number[, *job_number* ...]

job_number には、ジョブ番号、または、ジョブ番号の範囲（例 1-5 など）が指定できます。

qattach コマンドは、アタッチを許可するように設定されているキューに投入されたリクエストに対して、投入ユーザでのみ実行可能です。

-V　　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_INTERACTIVE_IP または NQSII_INTERACTIVE_IP
　クライアントホストの IP アドレス（IPv4 の数値とドットによる表記）

関連項目

qsub(1), qmgr(1M)

注意

- リクエストがアタッチ中の間、サスペンドおよびホールドはできません。
- X11 フォワーディングを有効にして実行中の会話リクエストに対してアタッチを行った場合でも、アタッチで実行するプログラムでは、X11 のフォワーディングは行いません。

1.4　qcat (1)

名前

qcat ―ジョブのエラー/入出力ファイルの表示

形式

qcat [-P *privilege*] [-V] [-e|-i|-o|-p|-d|--userpp-script=<*args*>] [-b|-f] [-n *number*]
[-h 1|2] *job_identifier* ...

【ジョブ指定】

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

【ランク指定】

機能説明

ジョブ指定の場合、*job_identifier* で指定されたジョブの入力ファイル、標準出力ファイル、エラー出力ファイル、ジョブ内で実行中のプロセスリスト、親リクエストのリクエストログ、もしくは、ユーザ PP スクリプトの標準出力、エラー出力のいずれかの内容を表示します。*job_identifier* は、 [*job_number*:]*request_identifier* の形式で指定します。*job_number* が省略された場合は、*request_identifier* で指定されたリクエストのジョブ番号 0 のジョブとなります。

NEC MPI プログラムを実行するリクエストにおいて、ランク指定が行えます。*request_identifier* に necmpi タイプのリクエストのリクエスト ID を指定しその前に@をはさんで、ユニバース番号（*universe*）、ランク番号（*rank*）を指定した場合（ランク指定）は、指定したリクエスト内での指定したユニバース番号、ランク番号を持つ MPI ジョブの標準出力、または、標準エラー出力ファイルの内容を表示することができます。
universe を省略した場合は、ユニバース番号 0 の情報を表示します。
ランク指定ができるのは、指定したリクエストが necmpi タイプのリクエストで、かつ、投入時に qsub コマンドの-f オプションでランク別出力を指定した場合のみです。
qsub -f でランク別出力を指定せずに投入した場合は、上記のジョブ指定の方法での出力が行えます。

以下のオプションが指定できます。

-b　ファイルの先頭行から表示します。
-d　親バッチリクエストのリクエストログを表示します。
-e　エラー出力ファイルがあれば表示します。
-f　ファイルの内容が増え続ける時、追加されたデータを出力します。
　-e, -o と同時に使用した場合に有効です。
-i　入力ファイル(スクリプトファイル)を表示します。
-o　標準出力ファイルがあれば表示します。
-p　ジョブ内で実行中のプロセスリスト(ps 出力イメージ)を表示します。

-h *type*

ヘッダを表示します。

-d オプションと同時に指定された場合はこのオプションは無視されます。*type* には、次の指定ができます。

【ジョブ指定の場合】

1. 各行の先頭に［*job_identifier*］を表示します。

2. ==> batch_job_identifier <== を表示し、次行からファイルの内容を表示します。

【ランク指定の場合】

1. 各行の先頭に［*applnum.universe.rank@request_identifier*］を表示します。

2. ==> *applnum.universe.rank@request_identifier* <== を表示し、次行からファイルの内容を表示します。

※*applnum* は、MPI プログラムの番号（ジョブ内の mpirun 実行毎に異なる値）を示します。

-n *number*

number で指定された行数分表示します。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

--userpp-script=<*args*>

ユーザ PP スクリプトの標準出力、または標準エラー出力を表示します。

<*args*>には以下の文字列のいずれかが指定できます。

文字列	動作
prr:o	PRE-RUNNING時のユーザPPスクリプトの標準出力を表示
prr:e	PRE-RUNNING時のユーザPPスクリプトの標準エラー出力を表示
por:o	POST-RUNNING時のユーザPPスクリプトの標準出力を表示
por:e	POST-RUNNING時のユーザPPスクリプトの標準エラー出力を表示

-e, -i, -o, -p, -d のいずれのオプションも指定されなければ、入力ファイルの表示を行います。
-b オプションが指定されなければ、ファイルの最終行から表示します。
-n オプションが省略された場合は、10 行分表示します。

環境変数

NQSV_PRIV または NQSII_PRIV

qcat 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

NQSV_INTERACTIVE_IP または NQSII_INTERACTIVE_IP

クライアントホストの IP アドレス（IPv4 の数値とドットによる表記）

関連項目

qstat(1), qsub(1)

1.5　qcmdconf (1)

名前

qcmdconf ― リモート実行型会話機能の管理

形式

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]

機能説明

1.　-a オプション

リモート実行したいプログラムのコマンドラインと、それを会話リクエストとして投入するための qrsh 用のオプションを登録し、登録内容どおりに qrsh コマンドでリモート実行を行うリモート実行型プログラムを作成します。

--cmd=*remote_cmd-line*
　リモートで実行したいプログラムのコマンドラインを指定します。指定可能なコマンドラインの文字数は最大 1023 バイトです。
　コマンドライン内に、オプションや引数の複数指定等で、スペースが入る場合は、シングルコート（’）またはダブルコート（”）で囲んでください。また、リモートへ制御文字等を引き渡したい場合は¥でエスケープしてください。

-f *file-name*
　リモート実行型プログラムのファイル名を、*file-name* にします。

--name=*entry-name*
　リモート実行したいプログラムのコマンドラインの登録名を指定します。
　下記の-f オプションを指定しない場合、この登録名がリモート実行型プログラムのファイル名となります。
　既に同じ登録名で登録されている名前を指定した場合にはエラーとなります。

--opt=*qrsh_options*
　qrsh コマンドに指定するオプション（資源制限等）を指定します。
　オプションや引数の複数指定等で間にスペースが入る場合は、シングルコート（’）またはダブルコート（”）で囲んでください。

--queue = *queue-name*
　qrsh コマンドに指定する投入先の会話キュー名を指定します。

-S
　管理者権限で登録を実行します。本オプション指定可能であるのは、root ユーザのみです。
　管理者権限で登録を行うことにより、全ユーザから利用可能なシステム共通のリモート

実行型プログラムを登録します。
-S オプションなしで登録を行った場合、qcmdconf コマンドを実行したユーザの個人用リモート実行型プログラムを登録します（ユーザのホームディレクトリ配下に個人用登録データファイルを作成します)。

2.　-d オプション

登録名を指定してそれに対応するリモート実行型プログラムの登録情報を削除します。登録情報を削除した後、リモート実行型プログラムのファイルは、rm コマンドで削除してください。

entry-name
削除したいリモート実行型プログラムの登録名を指定します。

-S
管理者権限で登録の削除を行います。管理者権限で登録したシステム共通の登録情報から、リモート実行型プログラムの登録削除を行います。
本オプションを指定しない場合は、起動ユーザの個人用登録データから、リモート実行型プログラムの登録削除を行います。

3.　-l オプション

利用できるリモート実行型プログラムの登録情報の一覧を標準出力に表示します。
システム共通の登録情報と、起動ユーザの個人用登録データを参照して、リモート実行型プログラム登録情報の一覧を表示します。

-S
管理者権限で登録情報の表示を行います。管理者権限で実行する場合、システム共通の登録情報のみを参照して、リモート実行型プログラム登録情報の一覧を表示します。

4.　-o オプション

既に登録されている登録名を指定してそれに対応するリモート実行型プログラムを作成します。リモート実行型プログラムのファイルを間違って消してしまった等、登録情報はあるが対応するリモート実行型プログラムがない場合に使用します。

entry-name
作成したいリモート実行型プログラムの登録名を指定します。下記の-f オプションでリモート実行型プログラムのファイル名を指定しない場合、この登録名がリモート実行型プログラムのファイル名となります。
entry-name の名前で登録されている情報が存在しない場合にはエラーとなります。

-f *file-name*
作成するリモート実行型プログラムのファイル名を、*file-name* にします。

-S

管理者権限でファイルの作成を行います。管理者権限で実行することにより、システム共通の登録情報を参照して、リモート実行型プログラムファイルを作成します。
本オプションを指定しない場合は、起動ユーザの個人用登録データを参照して、リモート実行型プログラムファイルを作成します。

ファイル

$HOME/.nqsv/qcmd_list	ユーザの個人用登録データ
/etc/opt/nec/nqsv/qcmd_list	システム共通の登録データ

関連項目

qrsh(1)

1.6　qdel (1)

名前

qdel ―リクエストの削除

形式

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

機能説明

request_identifier で指定されたリクエストを削除します。

以下のオプションが指定できます。

-g *grace*

実行ホスト上に存在するジョブへ、まず SIGTERM を送り、*grace* で指定された秒数を待った後 SIGKILL を送ります。*grace* が 0 以下の場合は SIGKILL のみを直ちに送ります。 本オプションが指定されない場合、5 秒待った後に SIGKILL を送ります。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

qdel 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限

PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

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

注意

結果ファイル転送中のリクエストを指定すると、そのリクエストを親とするネットワークリクエストも削除されます。

1.7　qdetach (1)

名前

qdetach ― リクエストへのアタッチ解除

形式

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

機能説明

request_identifier で指定したリクエストに対して qattach コマンドによりリクエストのジョブ内で実行しているプログラムを強制終了(デタッチ)します。
qdetach コマンドによりデタッチを行うには、指定したリクエストの所有者であるか、または操作員権限以上が必要です。

以下のオプションが指定できます。

-g *grace*

grace で指定した秒数を待った後デタッチを実行します。*grace* に 0 を指定した場合は、ただちにデタッチを実行します。-g オプション省略時の待ち時間の既定値は、5 秒です。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

qdetach 実行時に使用するアクセス権の既定値

PRIV_MGR　　管理者権限

PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qattach(1)

1.8　qhold (1)

名前

qhold ―リクエストのホールド

形式

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

機能説明

request_identifier で指定されたリクエストをホールドします。 この状態のとき、リクエストは実行されません。
本コマンドによってホールドされたリクエストは、qrls(1)コマンドによってホールドを解除できます。
リクエストをホールドした状態で NQSV がシャットダウンされた場合は、次回立ち上げ時にその状態が引き継がれます。

リクエストのホールドは実行待ち(QUEUED)、実行遅延(WAITING)状態でのみ可能です。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-s　-w オプションによるホールド処理完了待ち合わせ中に"." (ピリオド) を表示しません。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

-w　ホールド処理の完了まで qhold コマンドを終了せずに待ち合わせます。ホールドが成功した場合、Request <リクエスト ID> holding complete. が表示され、失敗した場合は

Request <リクエスト ID> holding failed. が表示されます。

環境変数

NQSV_PRIV または NQSII_PRIV

qhold 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

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

注意

1. 実行中のリクエストはホールドできません。
2. -w オプションを使用して本コマンドを実行した場合、待ち合わせ中に本コマンドを中断してもホールド処理は継続されます。また、待ち合わせ中に本コマンドがエラーで終了した場合もホールド処理は継続されます。ホールド処理の結果は qstat -f コマンドで確認できます。

1.9　qlogin (1)

名前

qlogin ― 会話リクエストの投入

形式

qlogin [*options*]

機能説明

qlogin は、NQSV へ会話リクエストを投入し、スケジューラによって割り当てられた実行ホスト上で会話的にコマンドの実行が行える環境を提供します。qlogin の実行により、qlogin を起動した端末上に実行ホストのシェルプロンプトが表示され、そのプロンプト上で、実行ホストのプログラム実行が行えます。

options に指定できるオプションは以下のとおりです。

-a *date_time*

date_time で指定された日時になるまで会話リクエストのスケジューリングを行いません。 *date_time* の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC	西暦年のはじめの 2 桁
YY	西暦年の後ろの 2 桁
MM	月(01-12)
DD	日(01-31)
hh	時(00-23)
mm	分(00-59)
SS	秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-A *account_string*

会話リクエストのアカウントコードを *account_string* に指定します。*account_string* に指定できる最大の文字数は、15 バイトまでです。

-A オプションが省略された場合、$HOME/.acct ファイルに記述されているアカウントコード名を使用します。$HOME/.acct の形式は次のとおりです。

- アカウントコード名として認識する文字列は、何らかの英字が現れてから空白，タブ，改行が現れるまでです（ファイルの先頭からの空白，タブ，改行は無視されます)。
- $HOME/.acct に記述できるアカウントコード名長は、15 バイトまでです。

本オプションが指定されず、$HOME/.acct も存在しない場合は、アカウントコードは設定されません。

上記のいずれかでアカウントコードが設定された場合、その内容については qlogin でチェック(アカウントコードの存在等)を行いません。

-b *job_count*

実行されるジョブ数を指定します。本オプションが指定されない場合、実行されるジョブ数は、1 となります。また、リクエスト形態が分散実行(-T distrib)の場合は、-b に指定できるのは 1 のみです。

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

スケジューラが、ジョブをジョブサーバに割り当てるための条件(ジョブコンディション)を指定します。本指定によるスケジューリングの詳細は JobManipulator のマニュアルを参照してください。

job_condition の形式は、以下のとおりです。

[*job_number*:]*condition*

job_number はリクエスト内のジョブ番号を 0 以上で指定します。

job_number の指定が省略もしくは ALL が指定された場合は、全ジョブが対象となります。

job_number には以下の指定が可能です。

(1) 数字の単独指定　　0:

(2) "ALL"による全ジョブ指定　ALL:

(3) ","で区切って複数の数字指定　　0,2,5:

(4) "-"による連続の番号指定　0-4:

(5) (3),(4)の組み合わせ　　0,2,4-6:

job_number の後ろには ":" が入り、それ以降に各スケジューラで定義された条件文(*condition*)を指定します。

job_number の指定条件に一致しない文字列は、*condition* として扱われ、*job_number* は ALL:となります。

-b オプションと同時に指定された場合、ジョブ数は-b で指定された数となります。

-b で指定されたジョブ数内で該当する *job_number* がない場合、そのジョブは条件なしとなります。*job_number* が-b で指定されたジョブ数を越えた場合は、エラーとなります。

job_condition に指定できる文字列は 255 バイトまでです。

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

リクエストやジョブ、プロセスに対する資源制限の最大値や警告値を設定します。カンマで区切って（スペースなしで）複数の *resource_list* を指定することができます。

resource_list は以下のとおりです。制限の最大値と警告値の両方を指定する場合 " で囲む必要があります。警告値を省略した場合は、最大値と同じ値に設定されます。

【リクエスト単位】

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

経過時間制限値

【ジョブ/論理ホスト単位】

cpunum_job=*max_limit*

CPU 台数制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 台数となります。

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

CPU 使用時間制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 使用時間となります。

gpunum_job=*max_limit*

GPU 台数制限値

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

メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のメモリサイズとなります。

socknum_job=*max_limit*

同時使用ソケット数制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のソケット数となります。

キューの設定がソケットスケジューリング機能 ON で、且つ、CPU 数指定方法"any"または"socket"の場合のみ指定可

cpunum_job との同時指定不可

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

仮想メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

上記は、ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定するためのレガシーの設定オプションであり、今後拡張しません。ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定する場合は、後述する「論理ホスト毎の資源制限オプション」(例：--cpunum-lhost)の使用を推奨します。

【プロセス単位】

coresz_prc=["]*max_limit*[,*warn_limit*"]
コアファイルサイズ制限値

cputim_prc=["]*max_limit*[,*warn_limit*"]
CPU 使用時間制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスの CPU 使用時間となります。

datasz_prc=["]*max_limit*[,*warn_limit*"]
データサイズ制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスのデータサイズとなります。

filenum_prc=*max_limit*
同時オープンファイル数制限値

filesz_prc=["]*max_limit*[,*warn_limit*"]
ファイルサイズ制限値

stacksz_prc=["]*max_limit*[,*warn_limit*"]
スタックサイズ制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスのスタックサイズとなります。

vecputim_prc=["]*max_limit*[,*warn_limit*"]
VE CPU 時間制限値
SX-Aurora TSUBASA の場合、VE 側のプロセスの CPU 時間制限値となります。

vememsz_prc=["]*max_limit*[,*warn_limit*"]
VE メモリサイズ制限値
SX-Aurora TSUBASA の場合、VE 側のプロセスのメモリサイズとなります。

vmemsz_prc=["]*max_limit*[,*warn_limit*"]
仮想メモリサイズ制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスの仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

また、会話リクエストの実行時制限として以下の設定が可能です。

idle_timer=*max_limit*
会話リクエスト実行時のアイドルタイマー時間（分）

制限値はマシンにとって都合のよい値に変換される場合があります。

-m *mail_option*
会話リクエストの状態の変化についてのメールを送ります。*mail_options* には、次の指

定ができます。

a　会話リクエストが異常終了したときにメールを送ります。
b　会話リクエストが開始したときにメールを送ります。
e　会話リクエストが終了したときにメールを送ります。
n　メールを送りません（他の mail_options と同時に指定した場合、エラーとなります)。

-M *mail_list*

メールを送るユーザリストを指定します。*mail_list* の形式は、次のとおりです。

mail_address[,*mail_address, ...*]

指定できる *mail_list* の最大長は 1023 バイトです。
本オプションが指定されない場合、*mail_list* にはジョブを投入したホスト上での投入ユーザを設定します。

-N *name*

会話リクエストの名前を指定します。*name* は 63 バイトまで指定できます。
本オプションが指定されない場合、"QLOGIN"となります。

-q *destination*

会話リクエストを投入する会話キューを指定します。 本オプションが指定されない場合の投入先会話キューの既定値は、環境変数 PBS_INTERACTIVE_QUEUE で指定することができます。

-R　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します

-s *request_start_time*

本オプションは、JobManipulator の実行開始時刻指定を利用する場合の専用オプションです。
request_start_time で指定された日時に会話リクエストを実行開始します。詳細は JobManipulator のマニュアルを参照してください。
request_start_time の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)

DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-S *path_name*

会話リクエストにおいてリモート実行するシェルを指定します。 - S オプション省略時のデフォルトのシェルは、ユーザのログインシェルとなります。

-T *job_topology*

実行されるジョブの形態を指定します。*job_topology* には、次の指定ができます。

distrib	実行されるジョブは分散ジョブです。 -b オプションによるジョブ数指定にかかわりなく、常に実行ジョブ数は 1 となります。
intmpi	実行されるジョブは intmpi ジョブです。ジョブ内で IntelMPI プログラムを実行する場合は、本オプションを指定する必要があります。
mvapich	実行されるジョブは mvapich ジョブです。ジョブ内で MVAPICH プログラムを実行する場合は、本オプションを指定する必要があります。
necmpi	実行されるジョブは necmpi ジョブです。ジョブ内で NEC MPI プログラムを実行する場合は、本オプションを指定する必要があります。
openmpi	実行されるジョブは openmpi ジョブです。ジョブ内で OpenMPI プログラムを実行する場合は、本オプションを指定する必要があります。
pltmpi	実行されるジョブは pltmpi ジョブです。ジョブ内で Platform MPI プログラムを実行する場合は、本オプションを指定する必要があります。

本オプションが指定されない場合、実行されるジョブの形態は、distrib となります。

-U *user_attribute_list*

会話リクエストに対してユーザが任意の文字列を指定します。*user_attribute_list* の形式は、次のとおりです。

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

user_attribute と *value* の文字列長の合計は 4000 バイト以下でなければなりません。すでに存在する *user_attribute* が指定された場合、値は上書きされます。 *value* が省略

された場合は NULL 文字となります。
value に複数の値を指定する場合、" で囲む必要があります。

user_attribute="value1,value2"

本オプションで指定された文字列は、リクエストの動作に影響を与えません。

-v *variable_list*

リクエストを実行するときに使用される環境変数を指定します。*variable_list* の形式は、次のとおりです。

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

variable と *value* の文字列長の合計は 4000 バイト以下でなければなりません。
すでに存在する環境変数名が *variable* に指定された場合、値は上書きされます。
value に複数の値を指定する場合、" で囲む必要があります。

variable="value1,value2"

リクエスト実行時、以下の環境変数が既定値として設定されています。PBS_ENVIRONMENT 、 PBS_JOBID 、 PBS_SUBREQNO 、 PBS_JOBNAME 、 PBS_NODEFILE 以外の環境変数の値は本オプションによって変更することができます。

環境変数名	説明
PBS_ENVIRONMENT	リクエストが会話処理であることを示す、"PBS_INTERACTIVE"を設定します。
PBS_JOBID	会話リクエストのジョブIDが設定されます。
PBS_JOBNAME	会話リクエスト名が設定されます。
PBS_NODEFILE	リクエストを構成するジョブが実行されているホスト一覧が記載されたファイルへのパスが設定されます。
PBS_O_HOME	クライアントホスト上の環境変数 HOME が設定されます。
PBS_O_HOST	クライアントホスト名が設定されます。
PBS_O_LANG	クライアントホスト上の環境変数 LANG が設定されます。
PBS_O_LOGNAME	クライアントホスト上の環境変数 LOGNAME が設定されます。
PBS_O_MAIL	クライアントホスト上の環境変数 MAIL が設定されます。
PBS_O_PATH	クライアントホスト上の環境変数 PATH が設定されます。
PBS_O_SHELL	クライアントホスト上の環境変数 SHELL が設定されます。

PBS_O_TZ	クライアントホスト上の環境変数 TZ が設定されます。
PBS_O_WORKDIR	クライアントホスト上の作業用ディレクトリを設定します。

-V　会話リクエスト投入時のすべての環境変数を会話リクエスト実行環境にエクスポートします。

-W　即時実行を行うための実行ホストの割り当てが行えない場合に、実行ホスト割り当てを待ち合わせます。-W オプションを指定しない場合、実行ホストの割り当てが行えないときは、qlogin を終了します。
待ち合わせ中に、ctrl-C 等により qlogin コマンドを中断した場合は、会話リクエストを削除し終了します。
但し、この-W 指定は、会話キューの属性 Real Time Scheduling の設定が"manual"に設定されている場合のみ有効です。キューの設定が"manual"以外の場合には、キューの設定に従って、実行ホストの割り当てが行えないときに待ち合わせるか終了するかが決定されます。

-X　X11 フォワーディングを有効にします。

--custom *cr_name*=*n*[,*cr_name*=*n*...]
カスタムリソース *cr_name* の利用量 *n* を指定します。カンマで区切って（スペースなしで）複数のカスタムリソースの利用量を指定することができます。
n には 1～2147483647 の整数、または unused (0) が指定できます。投入先のキューで利用量制御対象外指定不可となっている場合、unused (0) の指定はできません。

--enable-cloud-bursting = {*yes* | *no*}
クラウドのコンピューティング資源へリクエストをバースティングして実行するかどうかを指定します。*yes* を指定した場合は、バースティングします。*no* を指定した場合は、バースティングしません。本オプションが未指定の場合、既定値は *no* です。

--exclusive {= *host*}
1 ホストあたり 1 ジョブで実行すること（排他実行）を指定してリクエストを投入します。
オプションの引数には"*host*"を指定します。引数の"*host*"は省略可能です。
本オプションとプロビジョニング機能のテンプレート指定（--template）は同時に使用することができません。

--group=*group_name*
リクエストのグループを指定します。*group_name* にはグループ名を指定します。
このオプションの指定が無い場合、リクエストのグループはコマンド実行時のグループ

です。
なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無効です。

--job-separator
ジョブグループの区切りを示します。---と略記することも可能です。

--mig=*gi-name*[:*ci-slice-count*[,*ci-slice-count*...]]
マルチインスタンス GPU (MIG)を指定します。
gi-name には GPU インスタンス(GI)プロファイル名を指定します。 指定可能な GI プロファイル名は qstat -Ef で確認できます。
指定した GI を更にコンピュートインスタンス(CI)単位で複数に分割する場合は、*ci-slice-count* を指定します。*ci-slice-count* を指定しない場合は、GI プロファイルと同じサイズの CI を作成します。
[指定例]
--mig=4g.20gb:1,1,2
この場合は、4g.20gb の GI プロファイルを使用し、GI を更に 3 つの CI に分割して割り当てます。--mig オプションを複数指定することで、複数の GI を割り当てることができます。
--gpunum-lhost と同時指定した場合は、--mig オプションのみが適用され、--gpunum-lhost は無視されます。

--template=*template_name*
プロビジョニング環境におけるジョブ実行環境のテンプレートを *template_name* で指定します。
なお、本オプション指定時、同時に-l オプションによる cpunum_job, memsz_job, gpunum_job の指定はできません。リクエストにはテンプレート *template_name* 内で定義された CPU 数、メモリサイズ、GPU 数が適用されます。

--use-hca=*num*
--use-hca=[{io|mpi|all}:*num*][,{io|mpi|all}:*num*...]
同一 PCIeSW 内の VE が使用する、HCA のポート数を *num* で指定します。
または、HCA の種別(io または mpi または all）を先頭にコロンで区切って *num* を指定することで、HCA の種類（ScaTeFS ダイレクト IO 用・MPI 用・ScaTeFS ダイレクト IO 用と MPI 用の両方）別に HCA のポート数を指定することができます。*num* だけを指定した場合は、all が指定されたものと見做します。
本オプションは VE を使用するリクエストに対してのみ有効です。

--userpp-script=*location*:*path*[,*location*:*path*]
location で指定した実行位置で *path* に指定したスクリプトを実行します。

location には、以下の文字列が使用できます。

prr　　実行位置が PRE-RUNNING です。

por　　実行位置が POST-RUNNING です

--venode=*total_venum*

total_venum で指定した数の VE ノードをリクエストに対して割り当てます。投入先キューの既定搭載 VE ノード数に従って論理ホスト数は自動的に決定されます。1 以上の値を指定する必要があります。また、--venum-lhost オプションと同時指定した場合、キューの既定搭載 VE ノード数の代わりに--venum-lhost オプションで指定した値を使用して、論理ホスト数が自動設定されます。本オプションと-b オプションによるジョブ数（論理ホスト数）の指定は同時に指定できません。

--warning-signal=elapstim:*signal*

経過時間制限の警告値を超過した場合に送信するシグナルを指定します。*signal* は次の形式で指定できます。

・シグナル名（例：SIGHUP）

・シグナル名から SIG を取り除いたもの（例：HUP）

・シグナル番号（例：1）詳細は signal(7)を参照してください。

省略した場合、SIGTERM シグナルが送信します。

【論理ホスト毎の資源制限オプション】

リクエスト投入時に -b *job_count* 指定で作成された各ジョブ（論理ホスト）について、論理ホスト毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。警告値を省略すると、最大値と同じ値に設定します。

詳細な書式については**資源制限値の形式**をご覧ください。

-l オプション（例：-l cpunum_job）でジョブ/論理ホスト単位の資源制限の最大値や警告値を設定できますが、-l はレガシーの設定であり今後拡張しないため、下記の設定オプションの使用を推奨します。

--cpunum-lhost=*max_limit*

論理ホスト毎の CPU 台数制限値を設定します（-l cpunum_job=*max_limit* CPU 台数制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 台数となります。

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

論理ホスト毎の CPU 使用時間制限値を設定します（-l cputim_job=*max_limit* CPU 使用時間制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 使用時間となります。

--gpunum-lhost=*max_limit*

論理ホスト毎の GPU 台数制限値を設定します（-l gpunum_job=*max_limit* GPU 台数制限値(ジョブ)に相当します)。

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

論理ホスト毎のメモリサイズ制限値を設定します（-l memsz_job=*max_limit* メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側のメモリサイズとなります。

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

論理ホスト毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

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

論理ホスト毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

--venum-lhost =*max_limit*

論理ホスト毎の VE ノード数制限値を設定します。

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

論理ホスト毎の仮想メモリサイズ制限値を設定します (-l vmemsz_job=*max_limit* 仮想メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の仮想メモリサイズとなります。

VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

【VE ノード毎の資源制限オプション】

VE ノード毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。警告値を省略すると、最大値と同じ値に設定します。

詳細な書式については**資源制限値の形式**をご覧ください。

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

VE ノード毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

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

VE ノード毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

資源制限値の形式

・　時間制限

時間に関する制限は以下の形式で指定します。

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

無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 秒です。

【指定例】

1234:58:21.29	1234 時間 58 分 21.29 秒
59:01	59 分 1 秒
12345	12345 秒
121.1	121.1 秒

・　サイズ制限

サイズに関する制限は以下の形式で指定します。

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

units に指定できる単位は以下のとおりです。

b	バイト
kb	キロバイト(1024 b)
mb	メガバイト(1024 kb)
gb	ギガバイト(1024 mb)
tb	テラバイト(1024 gb)
pb	ペタバイト(1024 tb)
eb	エクサバイト(1024 pb)

units を指定しなかった場合はバイトと解釈されます。無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 バイトです。

【指定例】

1234	1234 バイト
1234kb	1234 キロバイト
1234.5gb	1234.5 ギガバイト

環境変数

PBS_INTERACTIVE_QUEUE

-q オプションを指定しない場合の投入先会話キューの既定値

NQSV_INTERACTIVE_IP または NQSII_INTERACTIVE_IP

クライアントホストの IP アドレス（IPv4 の数値とドットによる表記）

関連項目

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

注意

1. -T オプションにて MPI ジョブを指定した場合には、アイドルタイマーによる自動セッション切断はできません。
2. 各オプションの区切り文字として使用している文字(-I オプションの " 等)を通常文字として指定する場合、エスケープする必要があります。

1.10 qmove (1)

名前

qmove ―リクエストのキュー移動

形式

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

機能説明

request_identifier で指定されたリクエストを *o-destination* で指定されたキューへ移動します。
または、*i-destination* で指定されたキューに存在する全てのリクエストを *o-destination* へ移動します。

ジョブが存在しない、実行待ち(QUEUED)、実行遅延(WAITING)、ホールド(HELD)状態のリクエストが移動できます。
また、パラメトリックリクエストの場合は、サブリクエストが生成されるまでの状態でのみ移動可能です。サブリクエストを移動することはできません。

以下のオプションが指定できます。

-f　移動対象のリクエストがジョブを持っている場合、ジョブを削除した上で移動します。
移動対象のリクエストがジョブを持たない場合、本オプションは無視されます。
移動対象がパラメトリックリクエストで、サブリクエストが既に生成されている場合は、生成されているサブリクエストが削除され、移動後のキューに対して、新たにサブリクエストが生成されます。

-P *privilege*
コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-q *i-destination*

i-destination に存在する全てのリクエストを移動します。 本オプションを省略すると環境変数 PBS_QUEUE で指定されたキューに存在する全てのリクエストを移動します。

-s　-w オプションと同時に使用し、-w オプションによる移動処理完了待ち合わせ中に"."（ピリオド）を表示しません。-w 以外のオプションと同時に使用した場合は無視されます。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

-w　-f オプションと同時に使用し、移動処理の完了までコマンドを終了せずに待ち合わせます。*request_identifier* が複数指定されている場合および-q オプションと組み合わせて使用された場合は、対象となるリクエストの完了を 1 つずつ待ち合わせて順番に移動します。-f 以外のオプションと同時に使用した場合は無視されます。

環境変数

PBS_QUEUE

デフォルトのキュー名

NQSV_PRIV または NQSII_PRIV

qmove 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qstat(1), qsub(1)

注意

1. JobManipulator で管理されているリクエストを -f オプションにより強制移動した場合、リクエストは一旦アサインマップから外され、次のスケジューリングインターバルにおいて再アサインされます。そのため、移動の実施前後で開始予定時刻が変わります。
2. -w オプションによる強制移動完了の待ち受け中に以下の状況が発生すると、リクエストの状態遷移が停止するため、スケジューラ等によって次の状態遷移が発生するまでの間は待ち受け状態が継続されます。
    - リクエストが EXITING 中に他の API クライアントから RERUN 要求を受け付けた場合
    - -f オプションによるリラン実施後の移動処理においてエラーが発生した場合

1.11 qrerun (1)

名前

qrerun ―リクエストのリラン

形式

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

機能説明

request_identifier で指定されたリクエストの処理を中止し、ジョブを持たない QUEUED 状態に戻します。 このとき、実行ホスト上で実行中のジョブはシグナルによってアボートします。 ジョブを持つリクエストであれば、リクエスト状態に関係なく QUEUED 状態へ戻すことが可能です。
リクエストをリランするとそのリクエストの実行が中止され、そのリクエストは投入されているキューに投入し直されます。ただし、リクエスト ID は以前につけられていたものを引き継ぎます。
request_identifier にパラメトリックリクエストを指定した場合、生成済、実行中のすべてのサブリクエストが削除され、新たにパラメトリックリクエストとして再投入されます。

以下のオプションが指定できます。

-g *grace*

実行ホスト上に存在するジョブへはまず SIGTERM を送り、grace で指定された秒数を待った後 SIGKILL を送ります。grace が 0 以下の場合は SIGKILL のみを直ちに送ります。 本オプションが指定されない場合、5 秒待った後に SIGKILL を送ります。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了し

ます。

環境変数

NQSV_PRIV または NQSII_PRIV

qrerun 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qstat(1), qsub(1)

1.12 qrls (1)

名前

qrls —リクエストのホールド解除

形式

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

機能説明

qhold(1)コマンドでホールドされたリクエストのホールドを解除します。
リクエストのホールドが解除できるのは、リクエストがホールド中のときに限ります。

ホールド状態を解除するとリクエストを保留する前の状態に戻されます。つまり、ホールドする前の状態が実行待ち(QUEUED)であれば QUEUED 状態に、実行遅延(WAITING)状態であれば WAITING 状態に戻されます。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

qrls 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限

PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

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

1.13 qrsh (1)

名前

qrsh ― リモート実行型会話リクエストの投入

形式

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

機能説明

queue-name で指定した会話キューへ会話リクエストを投入することにより、*remote_cmd-line* で指定したコマンドラインを実行ホスト上で会話リクエストとしてリモート実行します。標準入力をリモートコマンドに引き渡し、リモートコマンドの標準出力／標準エラー出力を qrsh の標準出力／標準エラー出力に出力します。
リモートコマンドの実行が終了すると、qrsh はそのリモートコマンドの終了コードを引き取り、自身の終了コードとして終了します。

-c *remote_cmd-line*

remote_cmd-line に、リモートで実行したいコマンドラインを指定します。

-q *queue-name*

リモート実行コマンドを会話リクエストとして投入する会話キューを指定します。
本オプションを省略した場合は、環境変数 PBS_INTERACTIVE_QUEUE に設定されている会話キューの既定値を使用します。

その他、以下のオプションが指定できます。

-a *date_time*

date_time で指定された日時になるまで会話リクエストのスケジューリングを行いません。 *date_time* の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC	西暦年のはじめの 2 桁
YY	西暦年の後ろの 2 桁
MM	月(01-12)
DD	日(01-31)
hh	時(00-23)
mm	分(00-59)
SS	秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-A *account_string*

会話リクエストのアカウントコードを *account_string* に指定します。account_string に指定できる最大の文字数は、15 バイトまでです。
-A オプションが省略された場合、$HOME/.acct ファイルに記述されているアカウントコード名を使用します。$HOME/.acct の形式は次のとおりです。

- アカウントコード名として認識する文字列は、何らかの英字が現れてから空白，タブ，改行が現れるまでです（ファイルの先頭からの空白，タブ，改行は無視されます)。
- $HOME/.acct に記述できるアカウントコード名長は、15 バイトまでです。

本オプションが指定されず、$HOME/.acct も存在しない場合は、アカウントコードは設定されません。
上記のいずれかでアカウントコードが設定された場合、その内容については qrsh でチェック(アカウントコードの存在等)を行いません。

-b *job_count*

実行されるジョブ数を指定します。本オプションが指定されない場合、実行されるジョブ数は、1 となります。また、リクエスト形態が分散実行(-T distrib)の場合は、-b に指定できるのは 1 のみです。

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

スケジューラが、ジョブをジョブサーバに割り当てるための条件(ジョブコンディション)を指定します。本指定によるスケジューリングの詳細は JobManipulator のマニュアルを参照してください。
job_condition の形式は、以下のとおりです。

[*job_number*:]*condition*

job_number はリクエスト内のジョブ番号を 0 以上で指定します。
job_number の指定が省略もしくは ALL が指定された場合は、全ジョブが対象となります。
job_number には以下の指定が可能です。

(1) 数字の単独指定　　0:
(2) "ALL"による全ジョブ指定　ALL:
(3) ","で区切って複数の数字指定　　0,2,5:
(4) "-"による連続の番号指定　0-4:
(5) (3),(4)の組み合わせ　　0,2,4-6:

job_number の後ろには ":" が入り、それ以降に各スケジューラで定義された条件文(*condition*)を指定します。
job_number の指定条件に一致しない文字列は、*condition* として扱われ、*job_number* は ALL:となります。
-b オプションと同時に指定された場合、ジョブ数は-b で指定された数となります。
-b で指定されたジョブ数内で該当する *job_number* がない場合、そのジョブは条

件なしとなります。*job_number* が-b で指定されたジョブ数を越えた場合は、エラーとなります。

job_condition に指定できる文字列は 255 バイトまでです。

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

リクエストやジョブ、プロセスに対する資源制限の最大値や警告値を設定します。カンマで区切って（スペースなしで）複数の *resource_list* を指定することができます。*resource_list* は以下のとおりです。制限の最大値と警告値の両方を指定する場合 " で囲む必要があります。警告値を省略した場合は、最大値と同じ値に設定されます。

【リクエスト単位】

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

経過時間制限値

【ジョブ/論理ホスト単位】

cpunum_job=*max_limit*

CPU 台数制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 台数となります。

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

CPU 使用時間制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 使用時間となります。

gpunum_job=*max_limit*

GPU 台数制限値

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

メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のメモリサイズとなります。

socknum_job=*max_limit*

同時使用ソケット数制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のソケット数となります。

キューの設定がソケットスケジューリング機能 ON で、且つ、CPU 数指定方法"any"または"socket"の場合のみ指定可

cpunum_job との同時指定不可

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

仮想メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のプロセスの仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

上記は、ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定するためのレガシーの設定オプションであり、今後拡張しません。ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定する場合は、後述する「論理ホスト毎の資源制限オプション」(例：--cpunum-lhost)の使用を推奨します。

【プロセス単位】

coresz_prc=["]*max_limit*[,*warn_limit*"]
　　コアファイルサイズ制限値

cputim_prc=["]*max_limit*[,*warn_limit*"]
　　CPU 使用時間制限値
　　SX-Aurora TSUBASA の場合、VH 側のプロセスの CPU 使用時間となります。

datasz_prc=["]*max_limit*[,*warn_limit*"]
　　データサイズ制限値
　　SX-Aurora TSUBASA の場合、VH 側のプロセスのデータサイズとなります。

filenum_prc=*max_limit*
　　同時オープンファイル数制限値

filesz_prc=["]*max_limit*[,*warn_limit*"]
　　ファイルサイズ制限値

stacksz_prc=["]*max_limit*[,*warn_limit*"]
　　スタックサイズ制限値
　　SX-Aurora TSUBASA の場合、VH 側のプロセスのスタックサイズとなります。

vecputim_prc=["]*max_limit*[,*warn_limit*"]
　　VE CPU 時間制限値
　　SX-Aurora TSUBASA の場合、VE 側のプロセスの CPU 時間制限値となります。

vememsz_prc=["]*max_limit*[,*warn_limit*"]
　　VE メモリサイズ制限値
　　SX-Aurora TSUBASA の場合、VE 側のプロセスのメモリサイズとなります。

vmemsz_prc=["]*max_limit*[,*warn_limit*"]
　　仮想メモリサイズ制限値
　　SX-Aurora TSUBASA の場合、VH 側のプロセスの仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

制限値はマシンにとって都合のよい値に変換される場合があります。

-m *mail_options*

会話リクエストの状態の変化についてのメールを送ります。*mail_options* には、次の指定ができます。

a	会話リクエストが異常終了したときにメールを送ります。
b	会話リクエストが開始したときにメールを送ります。
e	会話リクエストが終了したときにメールを送ります。
n	メールを送りません（他の mail_options と同時に指定した場合、エラーとなります)。

-M *mail_list*

メールを送るユーザリストを指定します。*mail_list* の形式は、次のとおりです。

mail_address[,*mail_address, ...*]

指定できる *mail_list* の最大長は 1023 バイトです。
本オプションが指定されない場合、*mail_list* にはジョブを投入したホスト上での投入ユーザを設定します。

-N *name*

会話リクエストの名前を指定します。*name* は 63 バイトまで指定できます。
本オプションが指定されない場合、"QRSH"となります。

-R　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します

-s *request_start_time*

本オプションは、JobManipulator の実行開始時刻指定を利用する場合の専用オプションです。
request_start_time で指定された日時に会話リクエストを実行開始します。詳細は JobManipulator のマニュアルを参照してください。
request_start_time の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC	西暦年のはじめの 2 桁
YY	西暦年の後ろの 2 桁
MM	月(01-12)
DD	日(01-31)
hh	時(00-23)

mm　分(00-59)
SS　秒(00-61)
hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-S *path_name*
会話リクエストのリモート実行するシェルを指定します。 - S オプション省略時のデフォルトのシェルは、ユーザのログインシェルとなります。

-T *job_topology*
実行されるジョブの形態を指定します。*job_topology* には、次の指定ができます。

distrib	実行されるジョブは分散ジョブです。-b オプションによるジョブ数指定にかかわりなく、常に実行ジョブ数は 1 となります。
necmpi	実行されるジョブは necmpi ジョブです。ジョブ内で NEC MPI プログラムを実行する場合は、本オプションを指定する必要があります。
openmpi	実行されるジョブは openmpi ジョブです。ジョブ内で OpenMPI プログラムを実行する場合は、本オプションを指定する必要があります。
intmpi	実行されるジョブは intmpi ジョブです。ジョブ内で IntelMPI プログラムを実行する場合は、本オプションを指定する必要があります。
mvapich	実行されるジョブは mvapich ジョブです。ジョブ内で MVAPICH プログラムを実行する場合は、本オプションを指定する必要があります。
pltmpi	実行されるジョブは pltmpi ジョブです。ジョブ内で Platform MPI プログラムを実行する場合は、本オプションを指定する必要があります。

本オプションが指定されない場合、実行されるジョブの形態は、distrib となります。

-U *user_attribute_list*
会話リクエストに対してユーザが任意の文字列を指定します。*user_attribute_list* の形式は、次のとおりです。

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

user_attribute と *value* の文字列長の合計は 4000 バイト以下でなければなりません。すでに存在する *user_attribute* が指定された場合、値は上書きされます。 *value* が省略された場合は NULL 文字となります。
value に複数の値を指定する場合、" で囲む必要があります。

user_attribute="value1,value2"

本オプションで指定された文字列は、リクエストの動作に影響を与えません。

-v *variable_list*

リクエストを実行するときに使用される環境変数を指定します。*variable_list* の形式は、次のとおりです。

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

variable と *value* の文字列長の合計は 4000 バイト以下でなければなりません。
すでに存在する環境変数名が *variable* に指定された場合、値は上書きされます。
value に複数の値を指定する場合、" で囲む必要があります。

variable="value1,value2"

リクエスト実行時、以下の環境変数が既定値として設定されています。PBS_ENVIRONMENT 、 PBS_JOBID 、 PBS_SUBREQNO 、 PBS_JOBNAME 、 PBS_NODEFILE 以外の環境変数の値は本オプションによって変更することができます。

環境変数名	説明
PBS_ENVIRONMENT	リクエストが会話処理であることを設定します。"PBS_INTERACTIVE"を設定します。
PBS_JOBID	会話リクエストのジョブIDが設定されます。
PBS_JOBNAME	会話リクエスト名が設定されます。
PBS_NODEFILE	リクエストを構成するジョブが実行されているホスト一覧が記載されたファイルへのパスが設定されます。
PBS_O_HOME	クライアントホスト上の環境変数 HOME が設定されます。
PBS_O_HOST	クライアントホスト名が設定されます。
PBS_O_LANG	クライアントホスト上の環境変数 LANG が設定されます。
PBS_O_LOGNAME	クライアントホスト上の環境変数 LOGNAME が設定されます。
PBS_O_MAIL	クライアントホスト上の環境変数 MAIL が設定されます。
PBS_O_PATH	クライアントホスト上の環境変数 PATH が設定されます。
PBS_O_SHELL	クライアントホスト上の環境変数 SHELL が設定されます。
PBS_O_TZ	クライアントホスト上の環境変数 TZ が設定されます。
PBS_O_WORKDIR	クライアントホスト上の作業用ディレクトリを設定しま

	す。

-V　会話リクエスト投入時のすべての環境変数を会話リクエスト実行環境にエクスポートします。

-W　即時実行を行うための実行ホストの割り当てが行えない場合に、実行ホスト割り当てを待ち合わせます。-W オプションを指定しない場合、実行ホストの割り当てが行えないときは、qrsh を終了します。
待ち合わせ中に、ctrl-C 等により qrsh コマンドを中断した場合は、会話リクエストを削除し終了します。
但し、この-W 指定は、会話キューの属性 Real Time Scheduling の設定が"manual"に設定されている場合のみ有効です。キューの設定が"manual"以外の場合には、キューの設定に従って、実行ホストの割り当てが行えないときに待ち合わせるか終了するかが決定されます。

-X　X11 フォワーディングを有効にします。

--custom *cr_name*=*n*[,*cr_name*=*n*...]
カスタムリソース *cr_name* の利用量 *n* を指定します。カンマで区切って（スペースなしで）複数のカスタムリソースの利用量を指定することができます。
n には 1～2147483647 の整数、または unused (0) が指定できます。投入先のキューで利用量制御対象外指定不可となっている場合、unused (0) の指定はできません。

--enable-cloud-bursting = {*yes* | *no*}
クラウドのコンピューティング資源へリクエストをバースティングして実行するかどうかを指定します。*yes* を指定した場合は、バースティングします。*no* を指定した場合は、バースティングしません。本オプションが未指定の場合、既定値は *no* です。

--exclusive {= *host*}
1 ホストあたり 1 ジョブで実行すること（排他実行）を指定してリクエストを投入します。
オプションの引数には"*host*"を指定します。引数の"*host*"は省略可能です。
本オプションとプロビジョニング機能のテンプレート指定（--template）は同時に使用することができません。

--group=*group_name*
リクエストのグループを指定します。*group_name* にはグループ名を指定します。
このオプションの指定が無い場合、リクエストのグループはコマンド実行時のグループです。
なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無効で

す。

--job-separator
ジョブグループの区切りを示します。---と略記することも可能です。

--mig=*gi-name*[:*ci-slice-count*[,*ci-slice-count*...]]
マルチインスタンス GPU (MIG)を指定します。
gi-name には GPU インスタンス(GI)プロファイル名を指定します。 指定可能な GI プロファイル名は qstat -Ef で確認できます。
指定した GI を更にコンピュートインスタンス(CI)単位で複数に分割する場合は、*ci-slice-count* を指定します。*ci-slice-count* を指定しない場合は、GI プロファイルと同じサイズの CI を作成します。
[指定例]
--mig=4g.20gb:1,1,2
この場合は、4g.20gb の GI プロファイルを使用し、GI を更に 3 つの CI に分割して割り当てます。--mig オプションを複数指定することで、複数の GI を割り当てることができます。
--gpunum-lhost と同時指定した場合は、--mig オプションのみが適用され、--gpunum-lhost は無視されます。

--template=*template_name*
プロビジョニング環境におけるジョブ実行環境のテンプレートを *template_name* で指定します。
なお、本オプション指定時、同時に-l オプションによる cpunum_job, memsz_job, gpunum_job の指定はできません。リクエストにはテンプレート *template_name* 内で定義された CPU 数、メモリサイズ、GPU 数が適用されます。

--userpp-script=*location*:*path*[,*location*:*path*]
location で指定した実行位置で *path* に指定したスクリプトを実行します。
location には、以下の文字列が使用できます。

prr	実行位置が PRE-RUNNING です。
por	実行位置が POST-RUNNING です。

--use-hca=*num*
--use-hca=[{io|mpi|all}:*num*][,{io|mpi|all}:*num*...]
同一 PCIeSW 内の VE が使用する、HCA のポート数を *num* で指定します。
または、HCA の種別(io または mpi または all）を先頭にコロンで区切って *num* を指定することで、HCA の種類（ScaTeFS ダイレクト IO 用・MPI 用・ScaTeFS ダイレクト IO 用と MPI 用の両方）別に HCA のポート数を指定することができます。*num* だけを指定した場合は、all が指定されたものと見做します。

本オプションは VE を使用するリクエストに対してのみ有効です。

--venode=*total_venum*

total_venum で指定した数の VE ノードをリクエストに対して割り当てます。投入先キューの既定搭載 VE ノード数に従って論理ホスト数は自動的に決定されます。1 以上の値を指定する必要があります。また、--venum-lhost オプションと同時指定した場合、キューの既定搭載 VE ノード数の代わりに--venum-lhost オプションで指定した値を使用して、論理ホスト数が自動設定されます。本オプションと-b オプションによるジョブ数（論理ホスト数）の指定は同時に指定できません。

--warning-signal=elapstim:*signal*

経過時間制限の警告値を超過した場合に送信するシグナルを指定します。*signal* は次の形式で指定できます。

・シグナル名（例：SIGHUP）
・シグナル名から SIG を取り除いたもの（例：HUP）
・シグナル番号（例：1）詳細は signal(7)を参照してください。

省略した場合、SIGTERM シグナルが送信します。

【論理ホスト毎の資源制限オプション】

リクエスト投入時に -b *job_count* 指定で作成された各ジョブ（論理ホスト）について、論理ホスト毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。警告値を省略すると、最大値と同じ値に設定します。
詳細な書式については**資源制限値の形式**をご覧ください。
-l オプション（例：-l cpunum_job）でジョブ/論理ホスト単位の資源制限の最大値や警告値を設定できますが、-l はレガシーの設定であり今後拡張しないため、下記の設定オプションの使用を推奨します。

--cpunum-lhost=*max_limit*

論理ホスト毎の CPU 台数制限値を設定します（-l cpunum_job=*max_limit* CPU 台数制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 台数となります。

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

論理ホスト毎の CPU 使用時間制限値を設定します（-l cputim_job=*max_limit* CPU 使用時間制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 使用時間となります。

--gpunum-lhost=*max_limit*

論理ホスト毎の GPU 台数制限値を設定します（-l gpunum_job=*max_limit* GPU 台数

制限値(ジョブ)に相当します)。

--memsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎のメモリサイズ制限値を設定します（-l memsz_job=*max_limit* メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側のメモリサイズとなります。

--vecputim-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

--vememsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

--venum-lhost =*max_limit*
論理ホスト毎の VE ノード数制限値を設定します。

--vmemsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の仮想メモリサイズ制限値を設定します (-l vmemsz_job=*max_limit* 仮想メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の仮想メモリサイズとなります。
VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

【VE ノード毎の資源制限オプション】
VE ノード毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。
警告値を省略すると、最大値と同じ値に設定します。
詳細な書式については**資源制限値の形式**をご覧ください。

--vecputim-venode=*max_limit*[,*warn_limit*]
VE ノード毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

--vememsz-venode=*max_limit*[,*warn_limit*]
VE ノード毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

資源制限値の形式

- 時間制限

時間に関する制限は以下の形式で指定します。

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

無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 秒です。

【指定例】

1234:58:21.29	1234 時間 58 分 21.29 秒
59:01	59 分 1 秒
12345	12345 秒
121.1	121.1 秒

・ サイズ制限

サイズに関する制限は以下の形式で指定します。

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

units に指定できる単位は以下のとおりです。

b	バイト
kb	キロバイト(1024 b)
mb	メガバイト(1024 kb)
gb	ギガバイト(1024 mb)
tb	テラバイト(1024 gb)
pb	ペタバイト(1024 tb)
eb	エクサバイト(1024 pb)

units を指定しなかった場合はバイトと解釈されます。無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 バイトです。

【指定例】

1234	1234 バイト
1234kb	1234 キロバイト
1234.5gb	1234.5 ギガバイト

環境変数

PBS_INTERACTIVE_QUEUE

-q オプションを指定しない場合の投入先会話キューの既定値

NQSV_INTERACTIVE_IP または NQSII_INTERACTIVE_IP

クライアントホストの IP アドレス（IPv4 の数値とドットによる表記）

関連項目

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

注意

各オプションの区切り文字として使用している文字(**-I** オプションの " 等)を通常文字として指定する場合、エスケープする必要があります。

1.14 qsig (1)

名前

qsig ―リクエストへのシグナル送信

形式

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

機能説明

request_identifier または、*job_identifier* で指定されたバッチリクエスト(ジョブ)にシグナルを送信します。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-s *signal*

リクエスト(ジョブ)に送信するシグナルを指定します。 *signal* は、次の形式で指定できます。

・シグナル名（例：SIGHUP)
・シグナル名から SIG を取り除いたもの（例：HUP)
・シグナル番号（例：1）詳細は signal(7) を参照してください。

-s オプションが省略された場合、SIGTERM シグナルを送信します。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

qsig 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

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

使用例

・　リクエストのサスペンド（一時停止）

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

・　リクエストのレジューム（一時停止解除）

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

注意

JobManipulator によりサスペンドされた以下のリクエストを管理者権限でレジュームした場合、レジュームされたリクエストは直後に実行再開されるため、リソースの競合が発生する場合があります。

・割り込みによりサスペンド
・smgr コマンドからのサスペンド

1.15 qstat (1)

名前

qstat — NQSV の状態表示

形式

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 -Pm [-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] [-n] [-l] [*execution_host* ...]

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

機能説明

qstat は、NQSV システムの各種状態を表示します。表示対象を指定するオプション（表示情報オプション）は、以下の種類があります。

-B　バッチサーバの情報

-D　スケジューラの情報
-E　実行ホストの情報
-G　ノードグループの情報
-J　ジョブの情報
-Q　キューの情報
-R　パラメトリックリクエストのサマリ情報
-S　ジョブサーバの情報
-T　ネットワークリクエストの情報
--custom　カスタムリソースの情報
--limit　グループ毎・ユーザ毎の制限情報
--template　テンプレートの情報
--venode　VE ノードの情報
--cloud_template　クラウドバースティングテンプレートの情報

上記の表示情報オプションを指定しない場合、リクエストの情報を表示します。以下に各種情報の表示機能について説明します。

1.　リクエストの情報

表示情報オプションを指定しない場合、リクエストの情報を表示します。*request_identifier* に、表示対象となるリクエスト ID を指定します。 形式は以下のとおりです。

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

request_number　リクエスト ID のシーケンス番号
パラメトリックリクエストの 場合は、その後に[]または[*n*]の形式で *n* にサブリクエスト番号を指定します。

batch_server　リクエスト ID のバッチサーバ名

batch_server_host　接続するバッチサーバホスト名

request_number が省略された場合は、コマンド実行者が所有する全てのリクエストとなります。また、*batch_server* が省略された場合は、*batch_server_host* 上のバッチサーバとなります。*batch_server_host* が省略された場合はデフォルトのバッチサーバホストに接続します。デフォルトのバッチサーバホストは、/etc/opt/nec/nqsv/api_client.conf で指定されたバッチサーバホスト、指定がない場合はローカルホストとなります。

request_identifier が省略された場合は、デフォルトのバッチサーバホスト上のコマンド実行者が所有する全てのリクエストが表示対象となります。

以下のオプションが指定できます。

-a

リクエストにアタッチ中かどうかの情報を表示します。

-c *cpumode*

CPU 時間を *cpumode* にしたがって表示します。

cpumode には、次の指定ができます。

0 現時点の CPU 時間使用量(CPU)の表示

1 リクエスト実行開始からの積算 CPU 時間使用量(ACCPU)の表示

本オプションが省略された場合は、現時点の CPU 時間使用量表示(CPU)となります。本オプション使用時の *cpumode* は 省略することができません。-f オプション指定時は本オプションの指定に関係なく現時点の CPU 時間使用量、リクエスト実行開始からの積算 CPU 時間使用量が表示されます。

-m

残経過時間を表示します。

本オプションが省略された場合は、現時点の経過時間表示(Elapse)となります。-f オプション指定時は本オプションの指定に関係なく現時点の経過時間が表示されます。

-q *quelist*

quelist で指定されたキューに投入されているリクエストを表示します。

que*list* にはバッチキュー、会話キューのキュー名を","で（スペースなしで）区切って複数指定できます。

-s

パラメトリックリクエストに対して、サブリクエストごとの情報を表示します。

-u *userlist*

userlist で指定されたユーザが所有しているリクエストを表示します。

userlist にはログイン名を","で（スペースなしで）区切って複数指定できます。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

--group[=*group_name*]

サマリ表示の場合はグループ名も合わせて表示します。

group_name でグループ名を指定した場合、指定グループのリクエストのみを表示します。

--planned-start-time

リクエストの実行開始予定時間を表示します。

時間は拡張 ISO8601 規格に準拠した以下のフォーマットで出力されます。

なお、文字列「T」は時間値が続くことを示す文字です。

yyyy-mm-ddThh:mm:ss

qstat で使用可能なオプションのうち、次に示すオプションは併用可能とし、これら以外のオプションは--planned-start-time を無視するものとします。

併用可能オプション:

[-P privilage] [-n] [-l] [-d] [-m] [-s] [-a] [-u userlist] [-q quelist]

[-c cpumode] [-o Item_list] [-O Item list] [--group[=group-name]]

[--adjust-column]

2.　バッチサーバの情報

-B オプションを指定した場合、バッチサーバの情報を表示します。*batch_server_host* には表示対象のバッチサーバホスト名を指定します。*batch_server_host* が省略された場合は、デフォルトのバッチサーバホストの情報を表示します。

以下のオプションが指定できます。

-L

本バッチサーバの NQSV のライセンス設定情報を表示します。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

3.　スケジューラの情報

-D オプションを指定した場合、スケジューラの情報を表示します。*scheduler_identifier* には表示対象となるスケジューラを指定します。形式は以下のとおりです。

[*scheduler_ID*][@*batch_server_host*]

scheduler_ID　スケジューラ ID

batch_server_host　バッチサーバホスト名

scheduler_ID が省略された場合は、*batch_server_host* 上のバッチサーバに接続されている全てのスケジューラが対象となります。

batch_server_host が省略された場合はデフォルトのバッチサーバホスト上のバッチサーバに接続されているスケジューラを表示します。

scheduler_identifier が省略された場合は、デフォルトのバッチサーバホスト上のバッチサーバに接続されている全てのスケジューラを表示します。

以下のオプションが指定できます。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

4.　実行ホストの情報

-E オプションを指定した場合、実行ホストの情報を表示します。*execution_host* には表示対象となる実行ホスト名を指定します。形式は以下のとおりです。

[*execution_host_name*][@*batch_server_host*]

execution_host_name　実行ホスト名

batch_server_host　バッチサーバホスト名

execution_host_name が省略された場合は、*batch_server_host* のバッチサーバが

管理している全ての実行ホストを表示します。
batch_server_host が省略された場合はデフォルトのバッチサーバホスト上のバッチサーバの実行ホストを表示します。

以下のオプションが指定できます。

-g *node_group*

node_group で指定したノードグループに所属するジョブサーバの実行ホストを表示します。

-t

ジョブサーバがリンクダウン状態の実行ホストも含めて、バッチサーバに登録されている全ジョブサーバの実行ホストを表示します。
また、実行ホストの状態等の情報が追加となります。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

5.　ノードグループの情報

-G オプションを指定した場合、ノードグループの情報を表示します。*node_group* には表示対象となるノードグループ名を指定します。形式は以下のとおりです。

node_group_name[@*batch_server_host*]

node_group_name　ノードグループ名
batch_server_host　バッチサーバホスト名

6.　ジョブの情報

-J オプションを指定した場合、ジョブの情報を表示します。*job_identifier* には表示対象となるジョブ ID を指定します。

以下のオプションが指定できます。

-c *cpumode*

CPU 時間を *cpumode* にしたがって表示します。*cpumode* には、次の指定ができます。

0 現時点の CPU 時間使用量(CPU)の表示
1 リクエスト実行開始からの積算 CPU 時間使用量(ACCPU)の表示

本オプションが省略された場合は、現時点の CPU 時間使用量表示(CPU)となります。本オプション使用時の *cpumode* は 省略することができません。-f オプション指定時は本オプションの指定に関係なく現時点の CPU 時間使用量、リクエスト実行開始からの積算 CPU 時間使用量が表示されます。

-e

ジョブ内のプログラムが使用した VE 上の CPU 時間およびメモリ使用量を表示します。-m オプションおよび-c オプションと同時に指定した場合、-m オプションお

よび-c オプションの対象 CPU が VE 上の CPU になります。

-h *execution_host*

表示対象となる実行ホストを指定します。オプションの指定がない場合は、対象となるバッチサーバの全ての実行ホストが対象となります。本オプションと-F オプションを同時に指定する場合は、-F ehost を必ず指定してください。

-m

残 CPU 時間使用量を表示します。

-c オプションと同時に指定することはできません。**-f** オプション指定時は本オプションの指定に関係なく現時点の残 CPU 使用量が表示されます。

-t

ハイブリッドリクエストにおけるジョブ毎に固有な資源量を表示します。

-u *userlist*

userlist で指定されたユーザが所有しているジョブを表示します。*userlist* にはログイン名を","で（スペースなしで）区切って複数指定できます。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

7.　キューの情報

-Q オプションを指定した場合、キューの情報を表示します。*destination* には表示対象となるキューを指定します。形式は以下のとおりです。

[*queue_name*][@*batch_server_host*]

queue_name　　キュー名

batch_server_host　バッチサーバホスト名

queue_name が省略された場合は、*batch_server_host* のバッチサーバに存在する全てのキューを表示します。*batch_server_host* が省略された場合はデフォルトのバッチサーバホストのバッチサーバに存在するキューが対象となります。

destination が省略された場合は、デフォルトのバッチサーバホストのバッチサーバ上の全てのキューを表示します。

一般利用者の場合、アクセス権のあるキューのみの表示となります。「リクエストのグループ指定実行機能」が ON の場合、qstat 実行時のグループまたは--group オプションで指定したグループでアクセス権チェックを行います。
なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無視されます。

以下のオプションが指定できます。

-e

バッチキューの情報を表示します。

-i

会話キューの情報を表示します。

-N

ネットワークキューの情報を表示します。

-r

転送キューの情報を表示します。

-t

ジョブサーバ数の表示において、リンクダウンしているものも含めてバインドされている数を表示します。(既定値では、リンクアップしているジョブサーバ数を表示します。)

--group=*group_name*

「リクエストのグループ指定実行機能」が ON の場合、*group_name* で指定したグループでキューのアクセス権チェックを行います。

なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無効です。

8.　パラメトリックリクエストのサマリ情報

-R オプションを指定した場合、パラメトリックリクエストのサマリ情報を表示します。*parametric_request_identifier* には表示対象となるパラメトリックリクエストのリクエスト ID を指定します。

以下のオプションが指定できます。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

--group[= *group_name*]

サマリ表示の場合はグループ名も合わせて表示します。*group_name* でグループ名を指定した場合、指定グループのリクエストのみを表示します。

9.　ジョブサーバの情報

-S オプションを指定した場合、ジョブサーバの情報を表示します。*job_server* には、表示するジョブサーバを指定します。形式は以下のとおりです。

[*job_server_ID*][@*batch_server_host*]

job_server_ID　ジョブサーバ ID

batch_server_host　バッチサーバホスト名

job_server_ID が省略された場合は、*batch_server_host* のバッチサーバに存在する全てのジョブサーバの情報を表示します。*batch_server_host* が省略された場合はデフォルトのバッチサーバホスト上のバッチサーバに存在するジョブサーバを表示します。

以下のオプションが指定できます。

-g *node_group*

node_gourp で指定したノードグループに所属するジョブサーバを表示します。

-h *execution_host*

表示対象となる実行ホストを指定します。オプションの指定がない場合は、対象となるバッチサーバの全ての実行ホストが対象となります。

-t

リンクダウン状態のジョブサーバも含めて、バッチサーバに登録されている全ジョブサーバを表示します。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

10.　ネットワークリクエストの情報

-T オプションを指定した場合、ネットワークリクエストの情報を表示します。*request_identifier* は表示対象となるネットワークリクエストの親のリクエスト ID を指定します。

以下のオプションが指定できます。

-u *userlist*

userlist で指定されたユーザが所有しているリクエストが対象となります。*userlist* にはログイン名を","で（スペースなしで）区切って複数指定できます。

--adjust-column

サマリ表示で既定の列幅内で表示できない場合に、列幅を自動的に最大長に合わせて整形して表示します。

11.　カスタムリソースの情報

--custom オプションを指定した場合、システムで定義されている、カスタムリソースの情報を表示します。

12.　グループ毎・ユーザ毎の制限情報

--limit オプションを指定した場合、グループ名・ユーザ名を個別指定して設定できる制限に関連する情報を表示します。

なお、グループ管理者権限の場合は、管理するグループに関連する情報のみを表示します。

一般利用者の場合は、qstat コマンド実行時のユーザ・グループに関連する情報のみの表示となります。

以下のオプションが指定できます。

--group=*group_name*

「リクエストのグループ指定実行機能」が ON の場合、qstat コマンドの実行は *group_name* で指定したグループとなります。
なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無視されます。

13.　テンプレートの情報

--template オプションを指定した場合、システムで定義されているテンプレートの情報を表示します。--VE オプションを合わせて指定した場合、GPU 数の代わりに、VE および HCA ポート数の情報を表示します。

14.　VE ノードの情報

--venode オプションを指定した場合、JSV が LINKUP している VI（VH）に搭載されている VE の情報を表示します。

15.　クラウドテンプレートの情報

--cloud_template オプションを指定した場合、クラウドバースティング用テンプレートの情報を表示します。

各表示情報オプション共通で、以下のオプションが指定できます。

-d　時間データの表示フォーマットを d+hh:mm:ss に変更します。
-f　詳細情報を表示します。
-l　出力される内容を規定の長さで切ることなく全て表示します。
-n　ヘッダを出力しません。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次の指定ができます。

m　管理者権限
o　操作員権限
g　グループ管理者権限
s　特別利用者権限
u　一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。
特別利用者権限では、一般利用者権限に加え、他人のリクエスト情報等が参照可能となります。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

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

表示する情報を示すオプション(B,D,E,G,J,Q,R,S,T,オプションなし)のそれぞれに対応したアイテムにしたがって、出力フォーマットをカスタマイズします。

[表示情報オプションなし]:
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 stt 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

それぞれの対応するアイテムの意味は下表のとおりです。

項目	意味	対象オプション
acpu	リクエスト実行開始からの積算 CPU 時間使用量	（無し）
	ジョブ実行開始からの積算 CPU 時間使用量	-J
active	存在するサブリクエスト数	-R
arr	arriving 状態のリクエスト数	-B, -Q -e, -Q -r, -Q -i

項目	意味	対象オプション
att	アタッチ接続の有無	（無し）
attbl	リクエストへのアタッチの可否	-Q -e, -Q -i
avecpu	ジョブ実行開始からの積算 VE CPU 時間使用量	-J
bind	バインド状態	-S
bstt	バッチサーバの状態	-B
bsvhost	バッチサーバホスト名	-B, -E, -G, -S
comment	コメント	-G
cpu	現時点の CPU 時間使用量	（無し）, -J
cpuavg1	過去 1 分間の CPU アベレージ	-E, -S
cpunum	ジョブ当たりの CPU 台数制限値	-J -t
cputim	ジョブ当たりの CPU 時間制限値	-J -t
done	終了したサブリクエスト数	-R
ecode	ジョブ終了コード	-J
ehost	実行ホスト名	（無し）, -B, -J, -Q -e, -Q -i, -S
ejid	実行ジョブ ID	-J
elaps	経過時間	（無し）
euser	ジョブ実行ユーザ	-J
ext	exiting 状態のリクエスト数	-B, -Q -e, -Q -i
fcpu	使用していない CPU 台数	-E
fmem1	空き物理メモリサイズ	-E
fswap1	空きスワップサイズ	-E
gmax	グループごとリクエスト投入数制限値	-B
gpunum	ジョブ当たりの GPU 台数制限値	-J -t
group	グループ	（無し）, -R
hflg	ホールドの可否	（無し）, -R
hld	held 状態のリクエスト数	-B, -Q -e, -Q -r, -Q -i

項目	意味	対象オプション
hol	holding 状態のリクエスト数	-Q -e
hwnm	ハードウェア名	-E
jid	ジョブ ID	-J
jno	ジョブ番号	-J, -J -t
jobs	リクエストのジョブ数	（無し）
	サブリクエストあたりのジョブ数	-R
	ジョブサーバあたりのジョブ数	-S
jsvnm	ジョブサーバ名	-S
jsvno	ジョブサーバ番号	-J, -S
jsvs	ジョブサーバ数	-G, -Q -e, -Q -i
ldavg1	過去 1 分間のロードアベレージ	-E, -S
link	リンク状態	-S
mem	リクエストが使用しているメモリサイズ	（無し）
	ジョブが使用しているメモリサイズ	-J
memsz	ジョブ当たりのメモリ制限値	-J -t
mflg	ジョブマイグレーションの可否	（無し）, -R
mid	マシン ID	-B
mig	migrating 状態のリクエスト数	-Q -e
ngrpnm	ノードグループ名	-G
ngrptype	ノードグループのタイプ	-G
nqrlm	ネットワークキュー単位の同時実行数制限値	-Q -N
nrlm	ネットワークキューの同時実行数制限値	-B
osnm	オペレーティングシステム名	-E
own	所有者	（無し）, -R, -T
pdate	実行開始予定時刻	（無し）
por	post-running 状態のリクエスト数	-Q -e, -Q -i
pri	プライオリティ	（無し）, -R

項目	意味	対象オプション
	キュープライオリティ	-Q -e, -Q -r, -Q -N, -Q -i
	ネットワークリクエストプライオリティ	-T
prr	pre-running 状態のリクエスト数	-Q -e, -Q -i
qdate	キューに投入された時刻 ※リクエストを転送した場合は、転送キューに転送されてきた時刻	（無し）
qtot	キューに登録されているリクエスト数	-Q -e, -Q -r, -Q -N, -Q -i
que	queued 状態のリクエスト数	-B, -Q -e, -Q -r, -Q -N, -Q -i
quenm	リクエストが存在するキュー名	（無し）, -R
	バインドしているキューのリスト	-D, -E, -G, -S
	キュー名	-Q -e, -Q -r, -Q -N, -Q -i
ques	バインドしているキュー数	-D
rcpu	残 CPU 時間使用量	-J
rdate	実行開始時刻	（無し）
rel	OS リリース番号	-E
relaps	残経過時間	（無し）
reqnm	リクエスト名	（無し）, -R
rflg	再実行の可否	（無し）, -R
rid	リクエスト ID	（無し）, -J, -J -t, -R
	親リクエスト ID	-T
rqrlm	転送キュー単位の同時実行数制限値	-Q -r
rrlm	転送キューの同時実行数制限値	-B
run	running 状態のリクエスト数	-B, -Q -e, -Q -r, -Q -N, -Q -i
rvecpu	残 VE CPU 時間使用量	-J
schid	スケジューラ ID	-D, -Q -e, -Q -i
schnm	スケジューラ名	-D
schost	スケジューラ実行ホスト名	-D

項目	意味	対象オプション
sdate	生成時刻	（無し）
stall	ストール状態	（無し）
stg	staging 状態のリクエスト数	-Q -e
stgdir	ステージング方向	-T
stgf	ステージングファイル	-T
stghost	ステージングファイル転送ホスト	-Q -N
stgno	ステージングファイル番号	-T
stt	リクエストの状態	（無し）, -R
	ネットワークリクエスト状態	-T
stt1	キューの状態（投入受付可否状態）	-Q -e, -Q -r, -Q -N, -Q -i
stt2	キューの状態（実行可否状態）	-Q -e, -Q -r, -Q -N, -Q -i
sus	suspending, suspended, resuming 状態のリクエスト数	-B, -Q -e, -Q -i
tot	バッチサーバが管理しているリクエスト数	-B
	全サブリクエスト数	-R
trs	transiting 状態のリクエスト数	-Q -r
ucpu	使用中の CPU 台数	-E
umax	ユーザごとリクエスト投入数制限値	-B
umem1	使用中の物理メモリサイズ	-E
uswap1	使用中のスワップサイズ	-E
vecpu	現時点の VE CPU 時間使用量	-J
venum	ジョブ当たりの VE ノード数制限値	-J -t
vmemsz	ジョブ当たりの仮想メモリ制限値	-J -t
wai	waiting 状態のリクエスト数	-B, -Q -e, -Q -r, -Q -N, -Q -i

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

item で指定された項目をキーとして昇順にソートします。

本オプションは複数回指定することで、指定された順番にソート結果をさらにソー

トします。-O オプション（降順ソート）と組み合わせて指定することも可能です。
指定できる項目は-F と同じです。

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

item で指定された項目をキーとして降順にソートします。本オプションは複数回指定することで、指定された順番にソート結果をさらにソートします。-o オプション（昇順ソート）と組み合わせて指定することも可能です。
指定できる項目は-F と同じです。

標準出力

各情報表示で表示される内容は次のとおりです。

【リクエスト情報】

・ リクエスト ID（RequestID）
・ リクエスト名（ReqName）
・ 所有者（Username）
・ グループ（GrpName）(--group 指定時)
・ リクエストが存在するキュー名（Queue）
・ プライオリティ（Pri）
・ リクエストの状態（STT）

ARI　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　transferring
WAT　waiting

・ ストール状態（S）

Y　　ストール
－ 正常

・ リクエストが使用しているメモリサイズ（Memory）
・ 現時点の CPU 時間使用量（単位：秒）（CPU）（-c 0 指定時）
・ リクエスト実行開始からの積算 CPU 時間使用量（単位：秒）（ACCPU）（-c 1 指定時）

・ 経過時間（単位：秒）（Elapse）
・ 残経過時間（単位：秒）（RElapse）（-m 指定時）
・ 再実行の可否（R）
　Y　　再実行可能
　N　　再実行不可
・ ホールドの可否（H）
　Y　　ホールド可能
　N　　ホールド不可
・ ジョブマイグレーションの可否（M）
　Y　　ジョブマイグレーション可能
　N　　ジョブマイグレーション不可
・ リクエストのジョブ数（Jobs）
・ アタッチ接続の有無（Att）（-a 指定時）
・ 実行開始予定時間（PlannedStartTime）（--planned-start-time 指定時）

【リクエスト情報詳細（-f オプション指定）】
・ リクエスト ID（Request ID）
・ リクエスト名（Request Name）
・ 所有者（User Name）
・ グループ（Group Name）
・ 実行ユーザ ID（User ID）
・ グループ ID（Group ID）
・ 投入ユーザ名（Submitted User）（--user オプション指定時）
・ 現在のリクエストの状態（Current State）
　リクエストがストール状態の時は、状態の後に（Stalled）を表示
・ 一つ前のリクエストの状態（Previous State）
・ 現在の状態に遷移した時刻（State Transition Time）
・ 現在の状態に遷移した理由（State Transition Reason）
・ エラー発生により状態が遷移した場合、そのメッセージ（Error Message）
・ リクエストが存在するキュー名（Queue）
・ ジョブ形態（Job Topology）
・ プロセスマネージャ（Process Manager）
　【バッチリクエストかつジョブ形態が NEC MPI】
・ リクエストプライオリティ（Request Priority）
・ リクエストログの出力レベル（Request Loglevel）【会話リクエスト以外】
・ 再実行の可否（Rerunable）
・ ホールドの可否（Holdable）
・ ホールド実行時のクライアント権限（Hold Type）【会話リクエスト以外】
・ ジョブマイグレーションの可否（Migratable）【会話リクエスト以外】
・ サスペンド実行時のクライアント権限（Suspend Type）

・ アカウントコード（Account Code）
・ 標準出力ファイルのファイル名（Stdout）【会話リクエスト以外】
・ 標準エラー出力ファイルのファイル名（Stderr）【会話リクエスト以外】
・ リクエストログのファイル名（Reqlog）【会話リクエスト以外】
・ 使用シェルのフルパス名（Shell）
・ 強制シェルのパス名（Restrict shell）【会話リクエストのみ】
・ メールアドレス（Mail Address）
・ メールを送信するタイミング（Mail Option）
・ ジョブコンディション（Job Condition）
・ リクエスト内のジョブ数（Number of Jobs）
・ リクエストの生成日時（Created Request Time）
・ スケジューリング開始日時（Entered Queue Time）
・ リクエストの実行開始予定日時（Planned Start Time）
・ リクエスト実行指定日時（Execute Request Time）
・ リクエスト実行開始日時（Started Request Time）
・ リクエスト実行終了日時（Ended Request Time）
・ リクエスト実行開始要求日時（Requested Start Time）
・ デッドライン時刻（Deadline Time）【会話リクエスト以外】
・ User Mask 値（８進表示）（UMASK）
・ 予約区間 ID（Reservation ID）【会話リクエスト以外】
・ リクエスト連携時に所属するリクエストグループ番号（Request Group Number）【会話リクエスト以外】
・ リクエスト連携対象リクエストのうち最初に投入されるリクエストの ID（Lead Request）　【会話リクエスト以外】
・ 会話リクエスト接続ホスト(Interactive Host) 【会話リクエストのみ】
・ 会話リクエスト接続ポート(Interactive Port)　【会話リクエストのみ】
・ 会話リクエストのアイドルタイマー（Idle Timer）【会話リクエストのみ】
・ リクエストのアタッチ可否（qattach command）
・ アタッチ接続の有無(Attach)
・ クラスタタイプの選択（Cluster Type Select）
・ OpenStack テンプレート（OpenStack Template）または Container テンプレート(Container Template)またはクラウドバースティングテンプレート(Cloud Template)
CPU 数（CPU Number）
メモリ量（Memory Size）
GPU 数（GPU Number）
VE 数（VE Number）
HCA ポート数（HCA Number）
・ ユーザ PP スクリプト情報　（UserPP Script）
・ 排他実行（Exclusive）
・ VE ノード総数指定値（VE Node Number）

・ HCA 指定値（HCA Number）
・ SIGTERM 対応オプション（Accept Sigterm）
・ クラウドバースティング可否状態（Enable Cloud Bursting）
・ ユーザレベルチェックポイントスクリプト（Check Point Script）　【会話リクエスト以外チェックポイントスクリプト指定時のみ】
・ ユーザレベルリスタートスクリプト（Restart Script）【会話リクエスト以外チェックポイントスクリプト指定時のみ】
・ VE NUMA モード（VE NUMA Mode）
・ 資源制限の警告値超過時に送信するシグナル（Warning Signal）
・ カスタムリソース情報（Custom Resources）
カスタムリソース名と利用量
・ 実行ホスト情報（Execution Hosts(JSVNO)）
実行ホスト名と JSVNO
・ 資源情報（Resources Information）
リクエストが使用しているメモリサイズ（Memory）
現時点の CPU 時間使用量（単位：秒）（CPU Time）
リクエスト実行開始からの積算 CPU 時間使用量（単位 : 秒）（Accumulated CPU Time）
経過時間（Elapse）
残経過時間（Remaining Elapse）
Linux 上で実行されているリクエストが使用している仮想メモリサイズ（Virtual Memory）
割り当てられた MIG 情報(Multi Instance GPU Resources)
カスタムリソース使用量（カスタムリソース名）
・ 論理ホストの資源制限名と制限値情報（Logical Host Resources）
VE ノード数（VE Node Number）
CPU 数（CPU Number）
GPU 数（GPU Number）
CPU 時間（CPU Time）
メモリサイズ（Memory Size）
仮想メモリサイズ（Virtual Memory Size）
VE CPU 時間（VE CPU Time）
VE メモリサイズ　（VE Memory Size）
標準出力の結果ファイルサイズ（Stdout Size）【バッチリクエストのみ】
標準エラー出力の結果ファイルサイズ（Stderr Size）【バッチリクエストのみ】
・ VE ノード毎の資源制限名と制限値情報（VE Node Resource）
VE CPU 時間（VE CPU Time）
VE メモリサイズ　（VE Memory Size）
・ 資源制限名と制限値情報（Resource Limits）
リクエストごとの資源制限名と制限値（Per_Req）

ジョブごとの資源制限名と制限値（Per_Job）
プロセスごとの資源制限名と制限値（Per_Prc）
・ カーネルパラメータ名と値（Kernel Parameter）
・ スケジューラメッセージ（Scheduler Message）
・ ユーザ設定文字列（User Attributes）
・ ステージングファイル（Staging File）
ステージインファイル番号（Stage-in File Number）
ステージアウトファイル番号（Stage-out File Number）
転送元（From）
転送先（To）
ステージング状況（Status）
・ ジョブマイグレーションファイル（Migration File）
全チェックポイント対象の場合、（chkpnt_files）を表示

【バッチサーバ情報（-B オプション指定）】
・ バッチサーバホスト名（BatchServer）
・ マシン ID（MachineID）
・ リクエスト投入数制限値
UMAX　ユーザごと
GMAX　グループごと
・ 同時実行可能リクエスト数制限値
RRL　転送キューの同時実行数制限
NRL　ネットワークキューの同時実行数制限
・ バッチサーバが管理しているリクエスト数（TOT）
・ 各状態のリクエスト数
ARR　arriving
WAI　waiting
QUE　queued
RUN　running
EXT　exiting
HLD　held
SUS　suspending, suspended, resuming
・ 状態（Status）

【バッチサーバ情報詳細（-B -f オプション指定）】
・ バッチサーバホスト名（Batch Server）
・ NQSV バージョン（NQSV Version）
・ 状態（Batch Server State）
・ マシン ID（Batch Server Machine ID）
・ ログファイルパス（Logfile Path）

・ ログファイル出力レベル（Logfile Level）
・ ログファイル保存数（Logfile Save Count）
・ ログファイルサイズ（Logfile MAX Size）
・ アカウンィングサーバのホスト名（Accounting Server Host Name）
・ アカウンィングサーバのポート番号（Accounting Server Port Number）
・ ジョブアカウントファイルの出力ディレクトリ（Jacct directory）
・ 予算管理機能の ON/OFF の設定（Budget function）
・ リクエスト･ジョブアカウント情報の出力の ON/OFF の設定（Request Accounting）
・ リクエストアカウントファイルのパス名（Request Accounting File Path）
・ 予約アカウントの ON/OFF の設定（Reservation Accounting）
・ 予約アカウントファイルのパス名（Reservation Accounting File）
・ リクエストのグループ指定実行機能（Specify Group for Request）
・ ステージングファイルの絶対パス指定機能（Allow Absolute Exepath）
・ 投入失敗連携リクエスト自動削除機能の ON/OFF の設定（Auto Delete Failed Request）
・ バッチサーバが管理しているリクエスト数（Total Request）
・ 各状態のリクエスト数（xxx Request）
・ ハートビート間隔（Heart beat Interval）
・ 負荷情報やハードウェア搭載量情報採取間隔（Load Interval）
・ ジョブ資源量採取間隔（Get Resource Interval）
・ 同時生成サブリクエスト数制限値（Max Subrequest Entry Limit）
・ 転送キュー同時実行数制限値（Max Run Limit of Routing Queue）
・ 転送キューリトライ間隔（Retry Interval of Routing Queue）
・ 転送キューリトライ可能期間（Retry Span of Routing Queue）
・ ネットワークキュー同時実行数制限値（Max Run Limit of Network Queue）
・ ネットワークキューリトライ間隔（Retry Interval of Network Queue）
・ ネットワークキューリトライ可能期間（Retry Span of Network Queue）
・ リクエスト同時投入数制限値（Submit Number Limitation Value）
・ ユーザ投入数制限値（Submit User Number Limitation Value）
・ グループ投入数制限値（Submit Group Number Limitation Value）
・ ユーザ名／グループ名を指定しての制限値がある場合 （The other Limitation Values are setting.）
・ ライセンス情報（Use License）
使用しているライセンス数および最大使用可能なライセンス数（License (xxx)）

【バッチサーバ　ライセンス情報（-B -L オプション指定)】
・ システム名（Sysname）
・ 製品名（Product）
・ バージョン（Ver）
・ 有効期限（Expiration）

・ ライセンス数（Num）

【スケジューラ情報（-D オプション指定)】
・ スケジューラ ID（SCHID）
・ スケジューラ名（SCHName）
・ 実行ホスト（ExecutionHost）
・ バインドしているキュー数（BindQueue）

【スケジューラ情報詳細（-D -f オプション指定)】
・ スケジューラ ID（Scheduler ID）
・ スケジューラ名（Scheduler Name）
・ 実行ホスト（Execution Host）
・ バインドしているキューのリスト（Bind Queue list）
・ スケジューラバージョン（Scheduler Version）
・ スケジューラメッセージ（Scheduler Message）

【実行ホスト情報（-E オプション指定)】
・ 実行ホスト（ExecutionHost）
ベアメタルサーバの場合、先頭に[B]を表示
クラウドインスタンスの場合、先頭に[C]を表示
・ バッチサーバホスト名（BatchServer）
・ オペレーティングシステム名（OS）
・ OS リリース番号（Release）
・ ハードウェア名（Hardware）
・ VE 搭載ノード数（VE）
・ 過去 1 分間のロードアベレージ（Load）
・ 過去 1 分間の CPU アベレージ（Cpu）
-t オプション指定時には以下の情報も表示します。
・ ジョブサーバ番号（JSVNO）
・ ジョブサーバのリンク状態（JSV）
・ 実行ホストの状態（STT）
ACT　　稼働
INA　　停止
・ （S）　※未使用

【実行ホスト情報詳細（-E -f オプション指定)】
・ 実行ホスト（Execution Host）
ベアメタルサーバの場合、後ろに[Baremetal]を表示
クラウドインスタンスの場合、後ろに[Cloud]を表示
・ バッチサーバホスト名（Batch Server）

・ オペレーティングシステム名（Operating System）
・ OS バージョン番号（Version）
・ OS リリース番号（Release）
・ ハードウェア名（Hardware）
・ 使用している OpenStack テンプレート（OpenStack Template）
・ 使用しているクラウドバースティングテンプレート（Cloud Template）
・ ベアメタルサーバの登録資源量（Defined Baremetal Resouces）
（ベアメタルサーバの場合のみ）
・ 搭載 VE の詳細情報（Vector Engine Information）　（VH の場合）
VE Node　VE ノード番号
Cores　搭載コア数
Memory　搭載メモリ量
Status　VE のステータス
OS_Status　VEOS のステータス
・ ホスト負荷情報（Resource Information）
Memory　実行ホスト単位の物理メモリサイズに関する情報（単位：4KB）
Swap　実行ホスト単位のスワップサイズに関する情報（単位：4KB）
Number of Cpus　実行ホスト単位の CPU 実装台数に関する情報
・ Multi Instances GPU　GPU ごとの MIG 構成情報（MIG 対応の実行ホストのみ）
GPU 情報（GPU Information）(GPU 対応の実行ホストのみ)
・ ホストアベレージ情報（Average Information）
LOAD　実行ホスト単位のロードアベレージに関する情報
(Latest 1 minute)　過去 1 分間のロードアベレージ
(Latest 5 minutes)　過去 5 分間のロードアベレージ
(Latest 15 minutes)　過去 15 分間のロードアベレージ
CPU　実行ホスト単位の CPU アベレージに関する情報
(Latest 1 minute)　過去 1 分間の CPU アベレージ
(Latest 5 minutes)　過去 5 分間の CPU アベレージ
(Latest 15 minutes)　過去 15 分間の CPU アベレージ
・ CPUSET 情報（Cpuset Information）
（ソケットスケジューリング機能利用時）
リソースシェアリンググループ（Resource Sharing Group）
RSG NumberRSG 番号と CPUSET 情報
(none)　実行ホストで CPUSET が利用できない
・ ソケットリソース使用量（Socket Resource Usage）
（ソケットスケジューリング機能利用時）
NUMA ノード（NUMA Nodes）
Socket　ソケット番号とソケットリソース使用量／搭載量
(none)　実行ホストで NUMA アーキテクチャが利用できない
・ デバイストポロジ情報（Device Topology）

（HCA 割り当て機能、GPU-CPU Affinity 機能利用時）

Socket　　ソケットに接続されているデバイス情報

PCIeSW　　PCIe スイッチに接続されているデバイス情報

（none）　　実行ホストでデバイスリソースが定義されていない

-t オプション指定時には以下の情報も表示します。

・ 実行ホストの現在の状態（Current State）

Active　稼働状態

Inactive　停止状態

・ 現在の状態に遷移した時刻（State Transition Time）

・ 現在の状態に遷移した理由（State Transition Reason）

・ ジョブサーバ番号（Job Server Number）

・ ジョブサーバのリンク状態（LINK Batch Server）

UP　　ジョブサーバがバッチサーバとリンクされている状態

DOWN　ジョブサーバがバッチサーバとリンクされていない状態

・ ノード管理エージェント情報（Node Agent）

【ノードグループ情報（-G オプション指定）】

・ ノードグループ名（NodeGroup）

・ ノードグループのタイプ（Type）

・ バッチサーバホスト名（BatchServer）

・ コメント（Comment）

・ ジョブサーバ数（JSVs）

・ バインドしているキュー（BindQueue）

【ノードグループ情報詳細（-G -f オプション指定）】

・ ノードグループ名（Node Group）

・ ノードグループのタイプ（Type）

・ コメント（Comment）

・ バインドしているキュー（Bind Queue list）

・ ジョブサーバのリスト（Job Server number list）

以下はネットワークトポロジーノードグループの場合のみ表示

・ スイッチレイヤーのレベル（Switch Layer）

以下はクラウドバースティングノードグループの場合のみ表示

・ ロック状態（Lock State）

・ プライオリティ（Priority）

・ インスタンス数（Instances）

Live に当該ノードグループで起動しているクラウドインスタンス数を表示

Max に当該ノードグループで起動できる最大のインスタンス数を表示

・ ネットワーク名（Network Name）

・ テンプレート（Template）

【ジョブ情報（-J オプション指定)】

・ ジョブ番号（JNO）
・ リクエスト ID（RequestID）
・ 実行ジョブ ID（EJID）
・ ジョブが使用しているメモリサイズ（Memory）
・ ジョブが使用している VE のメモリサイズ（VEMemory）(-e 指定時)
・ 現時点の CPU 時間使用量（単位：秒）（CPU）(-c 0 指定時)
・ 現時点の VE の CPU 時間使用量（単位：秒）（VECPU）(-c 0 および-e 指定時)
・ ジョブ実行開始からの積算 CPU 時間使用量（単位 : 秒）（ACCPU）(-c 1 指定時)
・ ジョブ実行開始からの VE の積算 CPU 時間使用量（単位：秒）（AVECPU）(-c 1 および-e 指定時)
・ 残 CPU 時間使用量（単位：秒）（RCPU）(-m 指定時)
・ VE の残 CPU 時間使用量（単位：秒）（RVECPU）(-m および-e 指定時)
・ ジョブサーバ番号（JSVNO）
・ 実行ホスト（ExecutionHost）
　実行ホストの仮想マシン上でジョブが実行している場合、先頭に*を表示
　実行ホストがベアメタルサーバの場合、先頭に[B]を表示
　実行ホストのコンテナ内でジョブが実行している場合、先頭に*を表示
　実行ホストがクラウドインスタンスの場合、先頭に[C]を表示
・ ジョブ実行ユーザ（UserName）
・ ジョブ終了コード（16 進表示）（Exit）
　0xf*　：JSV ダウンにより終了コードを採取できなかった
　その他：wait(2)で採取する終了ステータス

【ジョブ毎に固有な資源量（-J -t オプション指定)】

・ ジョブ番号
・ リクエスト ID
・ ジョブ当たりの VE ノード数制限値
・ ジョブ当たりの CPU 台数制限値
・ ジョブ当たりの GPU 台数制限値
・ ジョブ当たりの CPU 時間制限値
・ ジョブ当たりのメモリ制限値
・ ジョブ当たりの仮想メモリ制限値

【ジョブ情報詳細（-J -f オプション指定)】

・ リクエスト ID（Request ID）
・ ジョブ番号（Batch Job Number）
・ 実行ジョブ ID（Execution Job ID）
・ ジョブ実行ユーザ（User Name）

・ ジョブ実行ユーザ ID（User ID）
・ ジョブ実行グループ ID（Group ID）
・ ジョブサーバ番号（Job Server Number）
・ ジョブサーバ名（Job Server name）
・ 実行ホスト（Execution Host）
　実行ホストの仮想マシン上でジョブが実行している場合、後ろに仮想マシンのホスト名を表示
　実行ホストがベアメタルサーバの場合、後ろに[Baremetal]を表示
　実行ホストのコンテナ内でジョブが実行している場合、後ろにコンテナのホスト名を表示
　実行ホストがクラウドインスタンスの場合、後ろに[Cloud]を表示
・ 実行ホスト（Vector Island）（-e 指定時）
・ 割り当てられた VE ノード番号（VE Node）（-e 指定時）
・ ジョブ終了コード（16 進表示）（Exit Code）
　0xf*　：JSV ダウンにより終了コードを採取できなかった
　その他：wait(2)で採取する終了ステータス
・ 資源情報（Resource Information）
　ジョブが使用しているメモリサイズ（Memory）
　ジョブが使用しているメモリ cgroup サイズ（Memory Cgroup Resources）
　ジョブが使用している VE のメモリサイズ（VEMemory）（-e 指定時）
　現時点の CPU 時間使用量（単位：秒）（CPU Time）
　現時点の VE の CPU 時間使用量（単位：秒）（VECPU Time）（-e 指定時）
　ジョブ実行開始からの積算 CPU 時間使用量（単位：秒）（Accumulated CPU Time）
　ジョブ実行開始からの VE の積算 CPU 時間使用量（単位：秒）(Accimulated VECPU Time)（-e 指定時）
　残 CPU 時間使用量（単位：秒）（Remaining CPU Time）
　VE の残 CPU 時間使用量（単位：秒）(Remaining VECPU Time)（-e 指定時）
　Linux 上で実行されているジョブが使用している仮想メモリサイズ（Virtual Memory）
　割り当てソケット番号（Assigned Sockets）
　カスタムリソース使用量（カスタムリソース名）

【ジョブ毎に固有な資源量詳細（-J -f -t オプション指定）】
・ リクエスト ID
・ ジョブ番号
・ HCA ポート数(HCA Number)
・ 予約区間 ID (Reservation ID)
・ 論理ホストごとの資源制限値情報（Resources Limits per Logical Host）
　VE ノード数（Number of VE Node）

CPU 台数（Number of CPU）
GPU 台数（Number of GPU）
CPU 時間（CPU Time）
メモリサイズ（Memory Size）
仮想メモリサイズ（Virtual Memory Size）
VE CPU 時間（VE CPU Time）
VE メモリサイズ （VE Memory Size）
標準出力の結果ファイルサイズ（Stdout Size）【バッチリクエストのみ】
標準エラー出力の結果ファイルサイズ（Stderr Size）【バッチリクエストのみ】

・ VE ノード毎の資源制限名と制限値情報（VE Node Resource）
VE CPU 時間（VE CPU Time）
VE メモリサイズ （VE Memory Size）
プロセスごとの資源制限値情報（Resources Limits per Process）
CPU 時間（CPU Time）
同時オープンファイル数（Number Open File）
メモリサイズ（Memory Size）
仮想メモリサイズ（Virtual Memory Size）
データセグメントサイズ（Data Segment Size）
スタックサイズ（Stack Segment Size）
コアファイルサイズ（Core File Size）
生成ファイルサイズ（Permanent File Size）
VE CPU 時間（VE CPU Time）
VE メモリサイズ （VE Memory Size）

【キュー情報（-Q オプション指定）】
・ キューのタイプ
EXECUTION QUEUE　バッチキュー
INTERACTIVE QUEUE 会話キュー
ROUTING QUEUE 転送キュー
NETWORK QUEUE　　ネットワークキュー
・ バッチサーバホスト名（Batch Server Host）
・ キュー名（QueueName）
・ スケジューラ ID（SCH）
・ バインドされているジョブサーバ数（JSVs）
・ ステージングファイル転送ホスト（StagingMachine）
・ キューの状態（投入受付可否状態）（ENA）
ENA　enabled
DIS　disabled
・ キューの状態（実行可否状態）（STS）
ACT　active

INA　inactive

・ キュープライオリティ（PRI）

・ 転送キュー、ネットワークキュー単位の同時実行数制限値（RLM）

<TOTAL>には、システム全体の同時実行数制限値を表示します。

・ キューに登録されているリクエスト数（TOT）

・ 各状態のリクエスト数

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> 各状態の合計

【キュー情報詳細（-Q -f オプション指定)】

・ キューのタイプ（Execution Queue, Interactive Queue, Routing Queue, Network Queue）

・ キュー名

・ キューの状態（実行可否、投入受付可否）（Run State, Submit State）

・ スケジューラ ID（Scheduler ID）

・ スケジューラ名（Scheduler Name）

・バインドされているジョブサーバ番号（Job Server Number）

・バインドされているノードグループ名（Node Group Name）

・ 補助グループ ID によるアクセス権チェック（Use Supplementary Groups ID for Access Privileges Check）

ON　キューのアクセス権チェックに補助グループ ID を使用する
OFF　キューのアクセス権チェックに補助グループ ID を使用しない

・ アクセス制限状況

Access User List　ユーザに関するアクセス制限状況
アクセスを許可されているユーザのリストを表示
Noaccess User List　ユーザに関するアクセス制限状況
アクセスが禁止されているユーザのリストを表示
Access Group List　グループに関するアクセス制限状況

アクセスを許可されているグループのリストを表示

Noaccess Group List グループに関するアクセス制限状況

アクセスが禁止されているグループのリストを表示

・ リクエスト投入経路制限（Refuse submission）

qsub | submit　　qsub, qlogin 等リクエスト投入コマンドからの投入を拒否

qmove　　qmove からのリクエスト投入を拒否

local_routing　ローカルからのリクエスト転送を拒否

remote_routing リモートからのリクエスト転送を拒否

(none)　　リクエスト投入経路に制限なし

・ ソケットスケジューリング機能（NUMA Control）

ON　ソケットスケジューリング機能を使用する

OFF　ソケットスケジューリング機能を使用しない

・ ソケット選択ポリシー（NUMA option）

コアバインドポリシー（Core Bind Policy）

concentration　　ソケット集中ポリシー

balance　　ソケット分散ポリシー

メモリアロケーションポリシー（Memory Allocation Policy）

membind　　membind ポリシー

localalloc　　localalloc ポリシー

interleave　　interleave ポリシー

・ CPU 数指定方法（Submit CPU Unit）

・ ジョブ毎の CPU 台数とメモリ量の比率チェック機能（NUMA Unit Check）

・ ホールド制限（Hold Privilege）　　【会話キューを除く】

User　　ホールド要求には、一般利用者権限以上が必要

SpecialUser　　ホールド要求には、特別利用者権限以上が必要

GroupManager ホールド要求には、グループ管理者権限以上が必要

Operator　　ホールド要求には、操作員権限以上が必要

Manager　　ホールド要求には、管理者権限以上が必要

Scheduler　　ホールド要求には、スケジューラ権限以上が必要

(none)　　ホールド要求に権限は必要なし

・ サスペンド制限（Suspend Privilege）

User　　サスペンド要求には、一般利用者権限以上が必要

SpecialUser　　サスペンド要求には、特別利用者権限以上が必要

GroupManager サスペンド要求には、グループ管理者権限以上が必要

Operator　　サスペンド要求には、操作員権限以上が必要

Manager　　サスペンド要求には、管理者権限以上が必要

Scheduler　　サスペンド要求には、スケジューラ権限以上が必要

(none)　　サスペンド要求に権限は必要なし

・ キュープライオリティ（Queue Priority）

・ キュー単位のリクエスト投入数制限値（Submit Number Limit）

・ キュー単位のユーザ別リクエスト投入数制限値（Submit User Number Limit）
・ キュー単位のグループ別リクエスト投入数制限値（Submit Group Number Limit）
・ パラメトリックリクエストのサブリクエスト数制限値（Subrequest Number Limit）
・ 転送キュー、ネットワークキュー単位の同時実行数制限値（Total Run Limit）
・ ジョブサーバ自動バインド（Auto Bind JobServer）
　ON　　ジョブサーバ起動時に自動的に以前バインドされていたキューにバインドする
　OFF　　ジョブサーバ起動時に自動的にキューにバインドしない
・ リクエストへのアタッチの可否（qattach command）【バッチキューのみ】
・ リクエストのリラン可否（Rerun default）
　Yes　　リラン可能
　No　　リラン不可
・ リクエストのリラン回数制限（Rerun Count Limit）【バッチキューのみ】
・ IntelMPI リクエストのプロセスマネージャの設定（IntelMPI Process Manager）
　hydra　　Hydra プロセスマネージャを使用
・ フックスクリプト機能（Hook Function）
・ 排他実行リクエストの投入可否（Exclusive Submit）
・ ユーザ Exit タイムアウト時間（UserExit Timeout）
・ ユーザ PP スクリプトタイムアウト時間（UserPP Timeout）
・ ステージアウト実施設定（File Stageout）
・ 実行不可緊急リクエストの削除（Delete Failed Urgent Request）【バッチキューのみ】
・ Partial Process Swapping 利用可否（Partial Process Swapping）【バッチキューのみ】
・ GPU-CPU Affinity 機能（GPU-CPU Affinity）
　ON　GPU-CPU Affinity 機能を使用する
　OFF　GPU-CPU Affinity 機能を使用しない
・ GPU あたりの CPU コア数（GPU-CPU Affinity CPU per GPU）
・ NEC MPI リクエストのプロセスマネージャの設定（Process Manager）【バッチキューのみ】
　hydra　Hydra プロセスマネージャを使用
　mpd　　MPD プロセスマネージャを使用
・ リクエストごとのジョブ数の下限値と上限値（Range of Jobs Limit per Batch/Interactive Request (min,max)）
・ アクセス権別に設定可能なリクエストプライオリティの下限値と上限値（Range of Request Priority）　【会話キューを除く】
　manager(min,max)　　管理者権限でリクエスト投入時のリクエストプライオリティ下限値と上限値
　operator(min,max)　　操作員権限でリクエスト投入時のリクエストプ

ライオリティ下限値と上限値

specialuser(min,max)　特別利用者権限でリクエスト投入時のリクエストプライオリティ下限値と上限値

user(min,max)　一般利用者権限でリクエスト投入時のリクエストプライオリティ下限値と上限値

・ 会話リクエストの即時実行不可時の待ち合わせ設定（Real Time Scheduling）【会話キューのみ】

・ 会話キューのアイドルタイマー（Idle Timer）【会話キューのみ】

・ 強制実行シェル（Restrict Shell）　【会話キューのみ】

・ 既定搭載 VE ノード数（Defined VE Number）

・ VE ノード総数の指定可能な範囲（Submit VE Node Range）

・ 各状態のリクエスト数

Total Request　キューに登録されているリクエスト総数

xxx Request　各状態のリクエスト数

・ リクエスト転送先のキューリスト（Destinations）

・ リクエストごとのネットワークリクエスト同時実行数（Run Limit per Batch Request）

・ ステージングファイル転送先ホスト名（Staging Machine）

・ ステージングファイルのステージング方式（Staging Method）

Internal　バッチサーバ内蔵の標準ステージング方式

External　ユーザ定義の外部ステージング方式

・ ステージングファイル転送用拡張バッファサイズ（単位 : KB）（Staging Extended Buffer Size）

(none)　標準のステージングファイル転送用バッファを使用

・ ユーザ Exit スクリプト（UserExit Script）

Pre-running　各ジョブの実行開始前に実行されるスクリプト

Post-running　各ジョブの実行終了直後に実行されるスクリプト

・ カスタムリソース情報（Custom Resources）

リソース名

利用量の下限値と上限値（Range(min,max)）

利用量の既定値（Std）

利用量制御対象外指定の可否（Permit Unused）

Yes　利用量制御対象外指定（unused (0)）を許可（既定値）

No　利用量制御対象外指定は不可

・ 論理ホスト毎の資源制限範囲（Logical Host Resource Ranges）

・ VE ノード毎の資源制限範囲　（VE Node Resource Range）

・ HCA のポート数の資源制限範囲(Logical Host HCA Resource Ranges)

・ 資源制限名と制限値情報（Resource Limits）

リクエストごとの資源制限名と制限値（Per_Req）

ジョブごとの資源制限名と制限値（Per_Job）

プロセスごとの資源制限名と制限値（Per_Prc）

・ カーネルパラメータ名と値（Kernel Parameter）

・ ユーザ名／グループ名を指定しての制限値がある場合　（The other Limitation Values are setting.）

・ スケジューラメッセージ（Scheduler Message）

【パラメトリックリクエストのサマリ情報（-R オプション指定）】

・ パラメトリックリクエスト ID（RequestID）

・ リクエスト名（ReqName）

・ 所有者（Username）

・ グループ（GrpName）(--group 指定時)

・ リクエストが存在するキュー名（Queue）

・ プライオリティ（Pri）

・ パラメトリックリクエストの状態（STT）

ARI　arriving
EXT　exiting
HLD　held
QUE　queued
RUN　running
WAT　waiting

・ 再実行の可否（R）

Y　　再実行可能
N　　再実行不可

・ ホールドの可否（H）

Y　　ホールド可能
N　　ホールド不可

・ ジョブマイグレーションの可否（M）

Y　　ジョブマイグレーション可能
N　　ジョブマイグレーション不可

・ サブリクエストあたりのジョブ数（Jobs）

・ 全サブリクエスト数（TOTAL）

・ 存在するサブリクエスト数（ACTIVE）

・ 終了したサブリクエスト数（DONE）

【パラメトリックリクエストの情報詳細（-R -f オプション指定）】

・ パラメトリックリクエスト ID（Request ID）

・ リクエスト名（Request Name）

・ 所有者（User Name）

・ グループ（Group Name）

・ 実行ユーザ ID（User ID）

・ グループ ID（Group ID）
・ 現在のリクエストの状態（Current State）
・ 一つ前のリクエストの状態（Previous State）
・ 現在の状態に遷移した時刻（State Transition Time）
・ 現在の状態に遷移した理由（State Transition Reason）
・ エラー発生により状態が遷移した場合、そのメッセージ（Error Message）
・ リクエストが存在するキュー名（Queue）
・ ジョブ形態（Job Topology）
・ プロセスマネージャ（Process Manager）
　【バッチリクエストかつジョブ形態が NEC MPI】
・ リクエストプライオリティ（Request Priority）
・ リクエストログの出力レベル（Request Loglevel）
・ 再実行の可否（Rerunable）
・ ホールドの可否（Holdable）
・ ホールド実行時のクライアント権限（Hold Type）
・ ジョブマイグレーションの可否（Migratable）
・ サスペンド実行時のクライアント権限（Suspend Type）
・ アカウントコード（Account Code）
・ 標準出力ファイルのファイル名（Stdout）
・ 標準エラー出力ファイルのファイル名（Stderr）
・ リクエストログのファイル名（Reqlog）
・ 使用シェルのフルパス名（Shell）
・ メールアドレス（Mail Address）
・ メールを送信するタイミング（Mail Option）
・ ジョブコンディション（Job Condition）
・ サブリクエストあたりのジョブ数（Number of Jobs）
・ サブリクエスト番号情報（Sub-request Numbers）
・ サブリクエスト数（Number of Sub-requests）
　　全サブリクエスト数（TOTAL）
　　存在するサブリクエスト数（ACTIVE）
　　終了したサブリクエスト数（DONE）
・ リクエストの生成日時（Created Request Time）
・ スケジューリング開始日時（Entered Queue Time）
・ リクエストの実行開始予定日時（Planned Start Time）
・ リクエスト実行指定日時（Execute Request Time）
・ リクエスト実行開始日時（Started Request Time）
・ リクエスト実行終了日時（Ended Request Time）
・ リクエスト実行開始要求日時（Requested Start Time）
・ デッドライン時刻（Deadline Time）
・ User Mask 値（８進表示）（UMASK）

・ 予約区間 ID（Reservation ID）
・ リクエスト連携時に所属するリクエストグループ番号（Request Group Number）
・ リクエスト連携対象リクエストのうち最初に投入されるリクエストの ID（Lead Request）
・ リクエストのアタッチ可否（qattach command）
・ クラスタタイプの選択（Cluster Type Select）
・ OpenStack テンプレート（OpenStack Template）または Container テンプレート(Container Template)またはクラウドバースティングテンプレート(Cloud Template)
CPU 数（CPU Number）
メモリ量（Memory Size）
GPU 数（GPU Number）
VE 数（VE Number）
HCA ポート数（HCA Number）
・ ユーザ PP スクリプト情報（UserPP Script）
・ VE ノード総数指定値（VE Node Number）【--venode 指定時のみ】
・ HCA 指定値（HCA Number）
・ SIGTERM 対応オプション（Accept Sigterm）
・ クラウドバースティング可否状態（Enable Cloud Bursting）
・ カスタムリソース情報（Custom Resources）
・ ユーザレベルチェックポイントスクリプト（Check Point Script）【チェックポイントスクリプト指定時のみ】
・ ユーザレベルリスタートスクリプト（Restart Script）【チェックポイントスクリプト指定時のみ】
・ VE NUMA モード（VE NUMA Mode）
・ 資源制限の警告値超過時に送信するシグナル（Warning Signal）
・ 論理ホストの資源制限名と制限値情報（Logical Host Resources）
VE ノード数（VE Node Number）
CPU 数（CPU Number）
GPU 数（GPU Number）
CPU 時間（CPU Time）
メモリサイズ（Memory Size）
仮想メモリサイズ（Virtual Memory Size）
VE CPU 時間 （VE CPU Time）
VE メモリサイズ （VE Memory Size）
標準出力の結果ファイルサイズ（Stdout Size）【バッチリクエストのみ】
標準エラー出力の結果ファイルサイズ（Stderr Size）【バッチリクエストのみ】
・ VE ノード毎の資源制限名と制限値情報（VE Node Resource Ranges）
VE CPU 時間 （VE CPU Time）
VE メモリサイズ （VE Memory Size）
・ 資源制限名と制限値情報（Resource Limits）

リクエストごとの資源制限名と制限値（Per_Req）
ジョブごとの資源制限名と制限値（Per_Job）
プロセスごとの資源制限名と制限値（Per_Prc）

・ カーネルパラメータ名と値（Kernel Parameter）
・ ユーザ設定文字列（User Attributes）

【ジョブサーバ情報（-S オプション指定)】

・ ジョブサーバ番号（JSVNO）
・ ジョブサーバ名（JobServerName）
・ バッチサーバホスト名（BatchServer）
・ 実行ホスト（ExecutionHost）
  ベアメタルサーバの場合、先頭に[B]を表示
  クラウドインスタンスの場合、先頭に[C]を表示
・ リンク状態（LINK）
  UP　　ジョブサーバがバッチサーバとリンクされている状態
  DOWN　ジョブサーバがバッチサーバとリンクされていない状態
・ バインド状態（BIND）
  Y　　ジョブサーバがキューにバインドされている状態
  N　　ジョブサーバがキューにバインドされていない状態
  （※-t オプション指定時は、バインドしているキューの数を表示します。）
・ キュー名（Queue）
・ ジョブ数（Jobs）
・ 過去 1 分間のロードアベレージ（Load）
・ 過去 1 分間の CPU アベレージ（Cpu）

【ジョブサーバ情報詳細（-S -f オプション指定)】

・ ジョブサーバ名（Job Server Name）
・ ジョブサーバ番号（Job Server Number）
・ ジョブサーババージョン（Job Server Version）
・ バッチサーバホスト名（Batch Server）
・ 実行ホスト（Execution Host）
  ベアメタルサーバの場合、後ろに[Baremetal]を表示
  クラウドインスタンスの場合、後ろに[Cloud]を表示
・ リンク状態（LINK Batch Server）
  UP　　ジョブサーバがバッチサーバとリンクされている状態
  DOWN　ジョブサーバがバッチサーバとリンクされていない状態
・ バインド状態（BIND Queue）
  BIND　　ジョブサーバがキューにバインドされている状態
  UNBIND ジョブサーバがキューにバインドされていない状態
・ キュー名（Queue）

・ JobManipulator からの利用可否（Assign JobManipulator license）

YES　　JobManipulator から利用可能

NO　　JobManipulator から利用不可

・ ネットワークトポロジーノードグループ名（Network Topology Group）

・ ジョブサーバが属しているクラウドバースティングノードグループ（Cloud Group）

・ 使用している OpenStack テンプレート（OpenStack Template）

・ ベアメタルサーバの登録資源量（Defined Baremetal Resouces）

（ベアメタルサーバの場合のみ）

・ HCA 障害検知時の動作（HCA Failure Check）

OFF　　障害を検知したノードに対して何もしない

UNBIND　　障害を検知したノードの JSV をキューからアンバインドする

DOWN　　障害を検知したノードの JSV を終了させる

・ ノードヘルスチェックで障害を検知した時の動作（Health Check Action）

NOTHING　　障害を検知したノードに対して何もしない

UNBIND　　障害を検知したノードの JSV をキューからアンバインドする

DOWN　　障害を検知したノードの JSV を終了させる

・ ホスト負荷情報（Resource Information）

Memory　　実行ホスト単位の物理メモリサイズに関する情報（単位：4KB）

Swap　　実行ホスト単位のスワップサイズに関する情報（単位：4KB）

Number of Cpus　　実行ホスト単位の CPU 実装台数に関する情報

・ ホストアベレージ情報（Average Information）

LOAD　　実行ホスト単位のロードアベレージに関する情報

(Latest 1 minute)　　過去 1 分間のロードアベレージ

(Latest 5 minutes)　　過去 5 分間のロードアベレージ

(Latest 15 minutes)　　過去 15 分間のロードアベレージ

CPU　　実行ホスト単位の CPU アベレージに関する情報

(Latest 1 minute)　　過去 1 分間の CPU アベレージ

(Latest 5 minutes)　　過去 5 分間の CPU アベレージ

(Latest 15 minutes)　　過去 15 分間の CPU アベレージ

・ ジョブマイグレーションパラメータ情報（Migration_file Transfer Parameter Information）

Interface Hostname　　ファイル転送時の TCP/IP 接続用ネットワークインタフェースのホスト名

Socketbuffer Size　　ソケットバッファサイズ

(OS default)　　OS 既定のソケットバッファサイズ

I/O buffer Size　　ファイル I/O サイズ

・ スケジューラメッセージ（Scheduler Message）

【ネットワークリクエスト情報（-T オプション指定）】

・ 親リクエスト ID（RequestID）
・ 所有者（UserName）
・ ネットワークリクエストプライオリティ（Pri）
・ ステージング方向（Dir）
　IN　Stage-in
　OUT　Stage-out
・ ステージングファイル番号（STGNO）
・ ステージングファイル（StagingFile）
　Dir が IN の場合　転送元ファイル名
　Dir が OUT の場合　転送先ファイル名
・ ネットワークリクエスト状態（STT）
　QUE　queued
　RUN　running
　WAT　waiting

【ネットワークリクエスト情報詳細（-T -f オプション指定)】
・ 親リクエスト ID（Request ID）
・ 所有者（User Name）
・ 実行ユーザ ID（User ID）
・ グループ ID（Group ID）
・ ネットワークリクエストプライオリティ（Network Request Priority）
・ ステージング方向（Staging Direction）
　STAGE_IN　Stage-in
　STAGE_OUT　Stage-out
・ ステージングファイル番号（Staging Number）
・ ステージングファイル（Staging File）
　STAGE_IN の場合　転送元ファイル名
　STAGE_OUT の場合　転送先ファイル名
・ ステージングファイルのステージング方式（Staging Method）
　Internal　バッチサーバ内蔵の標準ステージング方式
　External　ユーザ定義の外部ステージング方式
・ ネットワークリクエスト状態（State）

【カスタムリソース情報（--custom オプション指定）】
・ カスタムリソース名（Custom Resource）
・ 消費単位（Consumer）
・ カスタムリソース利用量制御情報
　対象種別（Type）
　個別対象指定（Target）
　同時に利用可能なリソースの最大値（Available Resource Limit）

・　資源監視制御情報
資源監視及び使用量出力モード(Check Mode)
資源超過時のジョブ終了の可否(Terminate Job)
資源の単位(Unit)

【グループ毎・ユーザ毎の制限情報（--limit オプション指定）】
・ バッチサーバホスト名（BatchServer）
・ リクエスト同時投入数制限値（Submit Number Limitation Value）
・ ユーザ投入数制限値（Submit User Number Limitation Value）
ユーザ名指定による制限値
・ グループ投入数制限値（Submit GroupNumber Limitation Value）
グループ名指定による制限値
・ キューのタイプ（Execution Queue, Interactive Queue, Routing Queue）
・ キュー名
・ アクセス制限状況　【管理者、操作員、グループ管理者（Group List のみ）の場合】
Access User List　ユーザに関するアクセス制限状況
アクセスを許可されているユーザのリストを表示
Noaccess User List　ユーザに関するアクセス制限状況
アクセスが禁止されているユーザのリストを表示
Access Group List　グループに関するアクセス制限状況
アクセスを許可されているグループのリストを表示
Noaccess Group List　グループに関するアクセス制限状況
アクセスが禁止されているグループのリストを表示
・ キュー単位のリクエスト投入数制限値（Submit Number Limit）
・ キュー単位のユーザ名個別リクエスト投入数制限値（Submit User Number Limit）
ユーザ名指定による制限値
・ キュー単位のグループ名個別リクエスト投入数制限値（Submit Group Number Limit）
グループ名指定による制限値
・ リクエストごとのジョブ数の下限値と上限値（Range of Jobs Limit per Batch/Interactive Request (min,max)）
ユーザ名指定による制限値（User Limit）
グループ名指定による制限値（Group Limit）
・ 資源制限名と制限値情報（Resource Limits）
リクエストごとの経過時間制限（(Per-Req) Elapse Time Limit）
ユーザ名指定による制限値（User Limit）
グループ名指定による制限値（Group Limit）

【テンプレート情報（--template オプション指定）】

OpenStack Template：

・ テンプレート名（Template）
・ ロック状態（L）
　- 　　ロック解除
　Y 　　ロック中
・ OS イメージ名（Image）
・ フレーバ名（Flavor）
・ CPU 数（CPU）
・ メモリ量（Memory）
・ GPU 数（GPU）
・ 独自定義（Custom）
・ コメント（Comment）

Container Template：

・ テンプレート名（Template）
・ ロック状態（L）
　- 　　ロック解除
　Y 　　ロック中
・ イメージ名（Image）
・ CPU 数（CPU）
・ メモリ量（Memory）
・ GPU 数（GPU）
・ 独自定義（Custom）
・ コメント（Comment）

Cloud Template：

・ テンプレート名（Template）
・ ロック状態（L）
　- 　　ロック解除
　Y 　　ロック中
・ イメージ名（Image）
・ CPU 数（CPU）
・ メモリ量（Memory）
・ GPU 数（GPU）
・ VE 数（VE）
・ 独自定義（Custom）
・ コメント（Comment）

【VE テンプレート情報（--template --VE オプション指定）】

Container Template：

・ テンプレート名（Template）
・ ロック状態（L）

-　　ロック解除
Y　　ロック中

・ イメージ名（Image）
・ CPU 数（CPU）
・ メモリ量（Memory）
・ VE 数（VE）
・ MPI/IO 共用の HCA ポート数（HCA）
・ 独自定義（Custom）
・ コメント（Comment）

【テンプレート情報詳細（--template -f オプション指定）】

OpenStack Template：

・ テンプレート名
・ ロック状態（Lock State）
・ OS イメージ名（OS Image）
・ フレーバ名（Flavor）
・ CPU 数（CPU Number）
・ メモリ量（Memory Size）
・ GPU 数（GPU Number）
・ 起動見込時間（Boot Timeout）
・ 停止見込み時間（Stop Timeout）
・ 独自定義（Custom）
・ コメント（Comment）
・ 使用中のリクエスト数（Requests）

Container Template：

・ テンプレート名
・ ロック状態（Lock State）
・ イメージ名（Image）
・ CPU 数（CPU Number）
・ メモリ量（Memory Size）
・ GPU 数（GPU Number）
・ VE 数　（VE Number）
・ HCA ポート数（HCA Number）
・ 起動見込時間（Boot Timeout）
・ 停止見込み時間（Stop Timeout）
・ 独自定義（Custom）
・ コメント（Comment）
・ 使用中のリクエスト数（Requests）

Cloud Template：

・ テンプレート名

・ ロック状態（Lock State）
・ イメージ名（OS Image）
・ CPU 数（CPU Number）
・ メモリ量（Memory Size）
・ GPU 数（GPU Number）
・ VE 数　（VE Number）
・ 起動見込時間（Boot Timeout）
・ 停止見込み時間（Stop Timeout）
・ 独自定義（Custom）
・ コメント（Comment）

【VE ノード情報（--venode オプション指定）】
・ VI の実行ホスト名（VectorIsland）
・ VE ノード番号（VE_No）
・ VE のコア数（Cores）
・ VE のメモリサイズ（Memory）
・ VE のステータス（Status）
・ VEOS のステータス（OS_Status）

【テンプレート情報（--cloud_template オプション指定）】
Cloud Template：
・ テンプレート名（Template）
・ ロック状態（L）
　- ロック解除
　Y ロック中
・ イメージ名（Image）
・ CPU 数（CPU）
・ メモリ量（Memory）
・ GPU 数（GPU）
・ VE 数（VE）
・ 独自定義（Custom）
・ コメント（Comment）

【テンプレート情報詳細（--cloud_template -f オプション指定）】
Cloud Template：
・ テンプレート名（Template）
・ ロック状態（Lock State）
・ イメージ名（OS Image）
・ CPU 数（CPU Number）
・ メモリ量（Memory Size）

・ GPU 数（GPU Number）
・ VE 数　（VE Number）
・ 起動見込時間（Boot Timeout）
・ 停止見込み時間（Stop Timeout）
・ 独自定義（Custom）
・ コメント（Comment）

環境変数

NQSV_PRIV または NQSII_PRIV

qstat 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qsub(1)

注意

列幅調整機能（--adjust-column オプション）使用時、表示する情報量が多いと表示までに時間がかかる場合があります。

1.16 qsub (1)

名前

qsub ― バッチリクエストの投入

形式

qsub [*options*] [*script-file* ...]

機能説明

qsub コマンドは、NQSV にバッチリクエストを投入します。

*script-file*には バッチリクエストとして実行するシェルスクリプトファイルのファイル名を指定します。 このスクリプトファイルを指定しなかった場合は、標準入力からスクリプトが読み込まれます。 スクリプトはすぐにファイルとしてスプールされるので、スクリプトファイルを後から変更しても、投入されたバッチリクエストには反映されません。

バッチリクエストが正しく投入されると、バッチリクエストのリクエスト ID を表示します。リクエスト ID は *sequence_number.host-name* の形をとっています。 ここで言う *sequence_number* とは NQSV によりリクエストに割り当てられたシーケンス番号であり、*host-name* はバッチサーバのマシンの名前です。 リクエスト ID によって、リクエストを一意に特定することができます。

script-file は、複数指定できます。この場合、リクエスト間に関連性を持たせて実行順序を制御するリクエスト連携機能が動作します。リクエスト連携には直列連携と並列連携があります。指定可能な連携リクエスト数に制限はありませんが、コマンドラインに指定できる文字数制限は 4095 文字となりますので、その範囲内に指定する必要があります。

直列連携:　先行リクエストの実行完了を待って後続リクエストがスケジューリングされます。qsub コマンドでは、script1 script2 ...のようにスクリプトファイル名をスペースで区切って指定します。

並列連携:　連携している全リクエストが一斉にスケジューリングされます。qsub コマンドでは、script1:script2:... のようにスクリプトファイル名を":"で区切って指定します。

リクエスト連携機能では、投入された全ての連結対象リクエストは PRIV_MGR 権限での HELD 状態となりスケジューリング対象から外されます。全てのリクエストの投入が完了した時点で先頭のリクエストがリリースされます。したがって、リクエスト投入途中にオプション解析エラー等が発生した場合、すでに投入されているリクエストは HELD 状態のまま残ります。但し、バッチサーバによる残存するリクエストを自動で削除する機能もあります。詳細は、［管理編］連携リクエスト投入失敗時の残存リクエストの削除機能 を参照してください。

options には、qsub コマンドの動作やバッチリクエストの特性を指定するさまざまなオプションが指定できます。
リクエスト連携機能の場合、*options* に指定されたオプションは全てのリクエストに対して有効となります。スクリプトファイル内に指定された埋め込みオプションはそのリクエストに対してのみ有効となります。

options に指定できるオプションを以下に示します。

-a *date_time*

date_time で指定された日時になるまでバッチリクエストのスケジューリングを行いません。 *date_time* の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-A *account_string*

バッチリクエストのアカウントコードを *account_string* に指定します。*account_string* に指定できる最大の文字数は、15 バイトまでです。
qsub は、アカウントコード属性に *account_string* で指定された文字列を設定します。
-A オプションが省略された場合、$HOME/.acct ファイルに記述されているアカウントコード名を使用します。$HOME/.acct の形式は次のとおりです。

- アカウントコード名として認識する文字列は、何らかの英字が現れてから空白，タブ，改行が現れるまでです（ファイルの先頭からの空白，タブ，改行は無視されます)。
- $HOME/.acct に記述できるアカウントコード名長は、15 バイトまでです。

本オプションが指定されず、$HOME/.acct も存在しない場合は、アカウントコードは設定されません。
上記のいずれかでアカウントコードが設定された場合、その内容については qsub でチェック(アカウントコードの存在等)を行いません。

-b *batch_job_count*

実行されるジョブ数を指定します。本オプションが指定されない場合、実行されるジョ

ブ数は、1 となります。

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

スケジューラが、ジョブをジョブサーバに割り当てるための条件(ジョブコンディション)を指定します。本指定によるスケジューリングの詳細は JobManipulator のマニュアルを参照してください。

job_condition の形式は、以下のとおりです。

[*job_number*:]*condition*

job_number はリクエスト内のジョブ番号を 0 以上で指定します。

job_number の指定が省略もしくは ALL が指定された場合は、全ジョブが対象となります。

job_number には以下の指定が可能です。

(1) 数字の単独指定　　0:
(2) "ALL"による全ジョブ指定　ALL:
(3) ","で区切って複数の数字指定　　0,2,5:
(4) "-"による連続の番号指定　　0-4:
(5) (3),(4)の組み合わせ　　0,2,4-6:

job_number の後ろには ":" が入り、それ以降に各スケジューラで定義された条件文(*condition*)を指定します。

job_number の指定条件に一致しない文字列は、*condition* として扱われ、*job_number* は ALL:となります。

-b オプションと同時に指定された場合、ジョブ数は-b で指定された数となります。

-b で指定されたジョブ数内で該当する *job_number* がない場合、そのジョブは条件なしとなります。*job_number* が-b で指定されたジョブ数を越えた場合は、エラーとなります。

condition として JobManipulator により、指定可能な項目は以下のとおりです。

JSV (ジョブサーバ番号)	ジョブを実行するジョブサーバ番号を指定
HW (ハードウェア種別)	ジョブを実行するホストのハードウェアを指定
NGRP(ノードグループ名)	ジョブを実行するノードグループ名を指定

job_condition に指定できる文字列は 255 バイトまでです。

-C *directive_prefix*

スクリプトファイル中の埋め込みオプションを指定します。

-C オプションが省略された場合は、環境変数 PBS_DPREFIX の値が使用されます。

環境変数の定義がない場合は、PBS が使用されます。

-d [*hostname*:]*path_name*

バッチリクエストのリクエストログを、*path_name* で指定したファイルに向けます。*path_name* に *hostname*: を指定することによってホストを指定できます。*hostname* を省略した場合、バッチリクエストを投入したホストとなります。
path_name には、以下の変換文字が使用できます。

%m　リクエスト ID 内のマシン ID に展開されます
%r　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%%　%に展開されます

本オプションが指定されなかった場合、リクエストログはクライアントホストへ転送されません。

-D *log_level*

リクエストログのログレベルを指定します。 本オプションが指定されなかった場合、リクエストログは出力されません。*log_level* には以下の値が指定可能です。

0　リクエストログは出力されません(既定値)
1　リクエストに関するログを出力します
2　リクエスト、およびジョブに関するログを出力します

-e [*hostname*:]*path_name*

バッチリクエストの標準エラー出力を、*path_name* で指定したファイルに向けます。*hostname*:を指定することによって、*path_name* のファイルを出力するホストを指定できます。 *hostname* を省略した場合、バッチリクエストを投入したホストに出力します。
path_name には、以下の変換文字が使用できます。

%f　MPI プログラムのランク情報に展開されます。ランク情報の形式は、
applnum.universe.rank
で、*applnum* はアプリケーション番号、*universe* はユニバース番号、*rank* はランク番号です。%f は qsub コマンドで-f オプションを指定してリクエストを投入し、necmpi ジョブとして NEC MPI プログラムを実行する場合のみ有効です。
%m　リクエスト ID 内のマシン ID に展開されます
%r　リクエスト ID に展開されます
%P　リクエスト ID に展開されます（パラメトリックリクエストの場合サブリクエスト番号を[]でなく_を使用して展開します。）
%s　リクエスト ID 内のシーケンス番号に展開されます
%t　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。
%[0*n*]j　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。
（例）%j　：0, 1, 2...

%03j　：000, 001, 002...
%%　%に展開されます

-e オプションで標準エラー出力の出力ファイルを指定しない場合のファイル名は、既定値として以下の形式のファイル名となります。

request_name.e%s

パラメトリックリクエストの場合、サブリクエストごとに標準エラー出力ファイルが生成され、各サブリクエストの標準エラー出力ファイルのファイル名には、上記のファイル名の後に、"."(ピリオド)＋サブリクエスト番号が付加されます。

-f

necmpi ジョブタイプのリクエスト（-T オプション参照）の実行結果の出力を MPI ランク毎に参照できるようにします。
本オプションを指定すると、リクエストの標準出力、標準エラー出力をジョブごとにまとめて出力することはできなくなります。本オプションは、NEC MPI のリクエストでのみ有効です。

-G ["]*migration_file_path*[","*migration_file_path*"...]

ジョブマイグレーション時に、転送先ホストへ同時に転送したいユーザファイルを絶対パスで *migration_file_path* に指定します。*migration_file_path* が '/'で終わる場合はディレクトリとみなされ、そのディレクトリ配下の全ファイルが転送対象となります。また、*migration_file_path* に "chkpnt_files" が指定された場合は、全てのチェックポイントファイルを転送対象とします。
migration_file_path には、以下の変換文字が使用できます。

%m　リクエスト ID 内のマシン ID に展開されます
%P　　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）
%r　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%t　　サブリクエスト番号に展開されます（パラメトリックリクエストの場合）
%%　%に展開されます

-h　バッチリクエストを投入した後に、そのリクエストをホールドします。

-H y | n

ホールドの可否を指定します。y はホールド可能リクエスト、n はホールド不可能リクエストとなります。本オプションが指定されない場合、ホールド可能リクエストとなります。

-I "*stage_in*"[,"*stage_in*"...]

ジョブ実行に先立って、*stage_in* で指定したファイルをクライアントホスト上から実行ホスト上へ転送します。*stage_in* の形式は以下のとおりです。

[*client_host*:]*file_path*,[*job_number*:]*executionhost_file_path*

client_host

転送するファイルが格納されている、クライアントホスト名を指定します。省略時はリクエスト投入ホストとなります。

file_path

転送するファイルの、クライアントホスト上のパス名を指定します。パス名が"/"で始まらない場合は、qsub 実行時のカレントディレクトリからの相対パスとみなします。

file_path には、以下の変換文字が使用できます。

%H　実行ホストのハードウェア名に展開されます

%m　リクエスト ID 内のマシン ID に展開されます

%n　数列を指定します。形式についての詳細は、後述の数列指定を参照願います。

%P　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）

%r　リクエスト ID に展開されます

%R　実行ホストのリリース名に展開されます

%s　リクエスト ID 内のシーケンス番号に展開されます

%S　実行ホストの OS 名に展開されます

%t　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。

%[0*n*]j　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。

(例)　%j　：0, 1, 2...

%03j　：000, 001, 002...

%[0*n*]t　サブリクエスト番号に展開されます。*n* にはサブリクエスト番号の桁数を指定します。

%(*START*-*END*)date

パラメトリックリクエストの場合、各サブリクエストに、*START*-*END* で指定した範囲の日付を対応させて、*YYYYMMDD* の形式の日付に展開します。

詳細は、後述の日付パラメータを参照願います。

%% %に展開されます

job_number

転送対象ファイルが、job_number で指定されたジョブ番号のジョブが実行される実行ホストに転送されます。

job_number には以下の指定が可能です。

(1) 数字の単独指定　　0:
(2) "ALL"による全ジョブ指定　　ALL:
(3) ","で区切って複数の数字指定　0,2,5:
(4) "-"による連続の番号指定 0-4:
(5) (3),(4)の組み合わせ　　0,2,4-6:
(6) %j によるジョブ番号指定%j:
(7) %n による数列指定　　%{0,2,5}n:

job_number の指定が省略もしくは ALL が指定された場合は、全ジョブにファイルが転送されます。
また、*job_number* が**-b** で指定されたジョブ数を越えた場合は、エラーとなります。

executionhost_file_path

転送したファイル（ステージインファイル）が格納される、実行ホスト上の環境変数 STGDIR からの相対パス名を指定します。STGDIR は、ジョブ実行時の環境変数として設定されます。ジョブ内からのステージインファイルへのアクセスは、${STGDIR}/*executionhost_file_path* に対して行ってください。
executionhost_file_path には、以下の変換文字が使用できます。

%m　　リクエスト ID 内のマシン ID に展開されます
%n　　数列を指定します。形式についての詳細は、後述の数列指定を参照願います。
%P　　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）
%r　　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%t　　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。
%[0*n*]j　　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。
(例)　%j　：0, 1, 2...

%03j：000, 001, 002...
%% %に展開されます

パス名にディレクトリを指定する場合は、最後に"/"を付加しなければなりません。
file_path にディレクトリが指定された場合は、そのディレクトリ配下に存在する全てのレギュラーファイルが転送対象となります。ただし、シンボリックリンクの場合は、シンボリック元のファイルが対象となります。ハードリンクの場合は、リンクが外れます。
executionhost_file_path にディレクトリが指定された場合は､そのディレクトリ配下に転送元と同じファイル名でファイルを格納します。
ただし、ディレクトリ指定からファイル指定へのステージングはできません。
また、セキュリティ上の理由により、SETUID/SETGID ビットは保存されません。

-j *join_list*

バッチリクエストの標準出力と標準エラー出力をマージします。マージされた結果は、*join_list* で指定された標準出力または標準エラーに出力します。*join_list* には、次の指定ができます。

e　　バッチリクエストの標準出力と標準エラー出力をマージした結果を標準エラーに出力します。

o　　バッチリクエストの標準出力と標準エラー出力をマージした結果を標準出力に出力します。

-J y | n

ジョブマイグレーションの可否を指定します。y はジョブマイグレーション可能リクエスト、n はジョブマイグレーション不可能リクエストとなります。本オプションが指定されない場合、ジョブマイグレーション可能リクエストとなります。

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

リクエストやジョブ、プロセスに対する資源制限の最大値や警告値を設定します。カンマで区切って（スペースなしで）複数の *resource_list* を指定することができます。*resource_list* は以下のとおりです。制限の最大値と警告値の両方を指定する場合 " で囲む必要があります。警告値を省略した場合は、最大値と同じ値に設定されます。

【リクエスト単位】
elapstim_req=["]*max_limit*[,*warn_limit*"]
経過時間制限値
【ジョブ/論理ホスト単位】
cpunum_job=*max_limit*
CPU 台数制限値
SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 台数となりま

す。

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

CPU 使用時間制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の CPU 使用時間となります。

gpunum_job=*max_limit*

GPU 台数制限値

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

メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のメモリサイズとなります。

socknum_job=*max_limit*

同時使用ソケット数制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側のソケット数となります。

キューの設定がソケットスケジューリング機能 ON で、且つ、CPU 数指定方法"any"または"socket"の場合のみ指定可

cpunum_job との同時指定不可

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

仮想メモリサイズ制限値

SX-Aurora TSUBASA の場合、ジョブの VH 側の仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

上記は、ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定するためのレガシーの設定オプションであり、今後拡張しません。ジョブ/論理ホスト単位の資源制限の最大値や警告値を設定する場合は、後述する「論理ホスト毎の資源制限オプション」(例：--cpunum-lhost)の使用を推奨します。

【プロセス単位】

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

コアファイルサイズ制限値

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

CPU 使用時間制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスの CPU 使用時間となります。

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

データサイズ制限値

SX-Aurora TSUBASA の場合、VH 側のプロセスのデータサイズとなります。

filenum_prc=*max_limit*
同時オープンファイル数制限値
filesz_prc=["]*max_limit*[,*warn_limit*"]
ファイルサイズ制限値
stacksz_prc=["]*max_limit*[,*warn_limit*"]
スタックサイズ制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスのスタックサイズとなります。
vecputim_prc=["]*max_limit*[,*warn_limit*"]
VE CPU 時間制限値
SX-Aurora TSUBASA の場合、VE 側のプロセスの CPU 時間制限値となります。
vememsz_prc=["]*max_limit*[,*warn_limit*"]
VE メモリサイズ制限値
SX-Aurora TSUBASA の場合、VE 側のプロセスのメモリサイズとなります。
vmemsz_prc=["]*max_limit*[,*warn_limit*"]
仮想メモリサイズ制限値
SX-Aurora TSUBASA の場合、VH 側のプロセスの仮想メモリサイズとなります。VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

パラメトリックリクエストの場合、各資源制限値は、サブリクエスト単位での制限値となります。

制限値はマシンにとって都合のよい値に変換される場合があります。

-m *mail_option*
バッチリクエストの状態の変化についてのメールを送ります。*mail_options* には、次の指定ができます。

a　バッチリクエストが異常終了したときにメールを送ります。
パラメトリックリクエストの場合、サブリクエストが異常終了した際に送ります。
b　バッチリクエストが開始したときにメールを送ります。
d　パラメトリックリクエストの各サブリクエストが終了した際に送ります。
e　バッチリクエストが終了したときにメールを送ります。
パラメトリックリクエストの場合、サブリクエストがすべて終了した際に送ります。
n　メールを送りません。

（他の mail_options と同時に指定した場合、エラーとなります。）

s　　パラメトリックリクエストの各サブリクエストが実行開始した際に送ります。

-M *mail_list*

メールを送るユーザリストを指定します。*mail_list* の形式は、次のとおりです。

mail_address[,*mail_address, ...*]

指定できる *mail_list* の最大長は 1023 バイトです。
本オプションが指定されない場合、*mail_list* にはジョブを投入したホスト上での投入ユーザを設定します。

-N *name*

バッチリクエストの名前を指定します。*name* は 63 バイトまで指定できます。
本オプションが指定されない場合、qsub コマンドのコマンド行で与えられたスクリプトファイルの名前をリクエスト名として使用します。qsub でスクリプトを指定しない場合は、"STDIN"となります。

-o [*hostname*:]*path_name*

バッチリクエストの標準出力を、*path_name* で指定したファイルに向けます。
hostname:を指定することによって、*path_name* のファイルを出力するホストを指定できます。*hostname* を省略した場合、バッチリクエストを投入したホストに出力します。
path_name には、以下の変換文字が使用できます。

%f　　MPI プログラムのランク情報に展開されます。ランク情報の形式は、
applnum.universe.rank
で、*applnum* はアプリケーション番号、*universe* はユニバース番号、*rank* はランク番号です。%f は qsub コマンドで-f オプションを指定してリクエストを投入し、necmpi ジョブとして NEC MPI プログラムを実行する場合のみ有効です。

%m　リクエスト ID 内のマシン ID に展開されます

%P　　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）

%r　　リクエスト ID に展開されます

%s　　リクエスト ID 内のシーケンス番号に展開されます

%t　　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。

%[0*n*]j　　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。
(例) %j　：0, 1, 2...
%03j　：000, 001, 002...

%%　%に展開されます

-o オプションで標準出力の出力ファイルを指定しない場合のファイル名は、既定値として以下の形式のファイル名となります。

request_name.o%s

パラメトリックリクエストの場合､サブリクエストごとに標準出力ファイルが生成され、各サブリクエストの標準出力ファイルのファイル名には、上記のファイル名の後に、"."(ピリオド)＋サブリクエスト番号が付加されます。

-O "*stage_out*"[,"*stage_out*"...]

ジョブの出力ファイルを、*stage_out* での指定により、実行ホストからクライアントホストへ転送します。*stage_out* の形式は以下のとおりです。

[*client_host*:]*file_path*,[*job_number*:]*executionhost_file_path*

client_host

転送したファイルを格納するクライアントホスト名を指定します。省略時はリクエスト投入ホストとなります。

file_path

転送したファイルを格納するクライアントホスト上のパス名を指定します。パス名が"/"で始まらない場合は、qsub 実行時のカレントディレクトリからの相対パスとみなします。

file_path には、以下の変換文字が使用できます。

%H　実行ホストのハードウェア名に展開されます
%m　リクエスト ID 内のマシン ID に展開されます
%n　数列を指定します。形式についての詳細は、後述の数列指定を参照願います。
%P　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）
%r　リクエスト ID に展開されます
%R　実行ホストのリリース名に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%S　実行ホストの OS 名に展開されます
%t　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。
%[0*n*]j　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。

(例)　%j　：0, 1, 2...
　　%03j：000, 001, 002...

%[0*n*]t　　サブリクエスト番号に展開されます。*n* にはサブリクエスト番号の桁数を指定します。

%(*START*-*END*)date
　　パラメトリックリクエストの場合、各サブリクエストに、*START*-*END* で指定した範囲の日付を対応させて、*YYYYMMDD* の形式の日付に展開します。
　　詳細は、後述の日付パラメータを参照願います。

%% %に展開されます

job_number

転送対象ファイルが、job_number で指定されたジョブ番号のジョブが実行された実行ホストから転送されます。*job_number* には以下の指定が可能です。

(1) 数字の単独指定　　0:
(2) "ALL"による全ジョブ指定　　ALL:
(3) ","で区切って複数の数字指定　0,2,5:
(4) "-"による連続の番号指定 0-4:
(5) (3),(4)の組み合わせ　　0,2,4-6:
(6) %j によるジョブ番号指定%j:
(7) %n による数列指定　　%{0,2,5}n:

job_number の指定が省略もしくは ALL が指定された場合は、全ジョブからファイルが転送されます。
また、*job_number* が**-b** で指定されたジョブ数を越えた場合は、エラーとなります。

executionhost_file_path

転送するファイル（ステージアウトファイル）が格納されている、実行ホスト上の環境変数 STGDIR からの相対パス名を指定します。STGDIR は、ジョブ実行時の環境変数として設定されます。ジョブ内からのステージアウトファイルへのアクセスは、${STGDIR}/*executionhost_file_path* に対して行ってください。
executionhost_file_path には、以下の変換文字が使用できます。

%m　　リクエスト ID 内のマシン ID に展開されます
%n　　数列を指定します。形式についての詳細は、後述の数列指定を参照願います。
%P　　リクエスト ID に展開されます（パラメトリックリクエストの場合、サブリクエスト番号を[]でなく_を使用して展開します。）

%r　　リクエスト ID に展開されます
%s　リクエスト ID 内のシーケンス番号に展開されます
%t　　パラメトリックリクエストの場合に、サブリクエスト番号に展開されます。
%[0*n*]j　　ジョブ番号に展開されます。*n* にはジョブ番号の桁数を指定します。
(例)　%j　：0, 1, 2...
%03j：000, 001, 002...
%% %に展開されます

パス名にディレクトリを指定する場合は、最後に"/"を付加しなければなりません。
executionhost_file_path にディレクトリが指定された場合は、そのディレクトリ配下に存在する全てのレギュラーファイルが転送対象となります。ただし、シンボリックリンクの場合は、シンボリック元のファイルが対象となります。ハードリンクの場合は、リンクが外れます。
file_path にディレクトリが指定された場合は、そのディレクトリ配下に転送元と同じファイル名でファイルを格納します。
ただし、ディレクトリ指定からファイル指定へのステージングはできません。
また、セキュリティ上の理由により、SETUID/SETGID ビットは保存されません。

-p *priority*

バッチリクエストの優先度を指定します。*priority* は、[-1024...1023]の整数でなければなりません。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次の指定ができます。

m　　管理者権限
o　　操作員権限
g　　グループ管理者権限
s　　特別利用者権限
u　　一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。
特別利用者権限では、一般利用者権限に加え、他人のリクエスト情報等が参照可能となります。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-q *destination*

バッチリクエストを投入するバッチキューを指定します。 本オプションが指定されな

い場合の投入先キューの既定値は、環境変数 PBS_QUEUE で指定することができます。

-r y | n
リクエストのリランの可否を指定します。y はリラン可能リクエスト、n はリラン不可能リクエストとなります。本オプションが指定されない場合、リラン可能リクエストとなります。

-R　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します

-s *request_start_time*
本オプションは、JobManipulator の実行開始時刻指定を利用する場合の専用オプションです。
request_start_time で指定された日時にバッチリクエストを実行開始します。詳細は JobManipulator のマニュアルを参照してください。
request_start_time の形式は次のとおりです。
[[[[*CC*]*YY*]*MM*]*DD*]*hhmm*[.*SS*]
CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。
CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)
hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-S *path_name*
バッチリクエストのシェルスクリプトを実行するシェルを指定します。指定されたシェルはログインシェルとして起動されます。ログインシェル起動時に読み込まれる /etc/profile 等で export した環境変数はジョブスクリプトに引き継がれます。

なお、本オプション指定時、指定したシェルの入力にジョブスクリプトをリダイレクトします。そのため、ジョブスクリプトの#!は解釈されない場合があります。
（指定したシェルの仕様にもよります。例えば、bash を-S に指定した場合#!は解釈されません。）詳細は使用するシェルの仕様を確認してください。

-t *subreqno*
subreqno で指定された範囲のサブリクエスト番号を持つサブリクエストを生成するパ

ラメトリックリクエストの投入を行います。
subreqno の形式は次のとおりです。

N	サブリクエスト番号＝*N* のサブリクエストを生成
N-*M*	*N* から *M* の範囲の整数をサブリクエスト番号として持つサブリクエストを生成（*N*<*M* である必要がある）
N-*M*:*S*	*N* から *M* の範囲で *S* ずつ増加する整数をサブリクエスト番号として持つサブリクエストを生成（*N*<*M* である必要がある）

-T *batch_job_topology*

実行されるジョブの形態を指定します。*batch_job_topology* には、次の指定ができます。

distrib	実行されるジョブは分散ジョブです。 -b オプションで指定された数だけジョブが生成され実行されます。各ジョブでは同一のスクリプトが実行されます。
intmpi	実行されるジョブは intmpi ジョブです。ジョブ内で IntelMPI プログラムを実行する場合は、本オプションを指定する必要があります。
mvapich	実行されるジョブは mvapich ジョブです。ジョブ内で MVAPICH プログラムを実行する場合は、本オプションを指定する必要があります。
necmpi	実行されるジョブは necmpi ジョブです。ジョブ内で NEC MPI プログラムを実行する場合は、本オプションまたは necmpi_hydra を指定する必要があります。 本オプションを指定すると、プロセスマネージャは投入キューに設定されている方式を使用します。
openmpi	実行されるジョブは openmpi ジョブです。ジョブ内で OpenMPI プログラムを実行する場合は、本オプションを指定する必要があります。
pltmpi	実行されるジョブは pltmpi ジョブです。ジョブ内で Platform MPI プログラムを実行する場合は、本オプションを指定する必要があります。
necmpi_hydra	実行されるジョブは necmpi ジョブです。ジョブ内で NEC MPI プログラムを実行する場合は、本オプションまたは necmpi を指定する必要があります。necmpi とは異なり、プロセスマネージャは必ず hydra 方式を使用します。

本オプションが指定されない場合、実行されるジョブの形態は、distrib となります。

-U *user_attribute_list*

バッチリクエストに対してユーザが任意の文字列を指定します。*user_attribute_list* の形式は、次のとおりです。

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

user_attribute と *value* の文字列長の合計は 4000 バイト以下でなければなりません。すでに存在する *user_attribute* が指定された場合、値は上書きされます。 *value* が省略された場合は NULL 文字となります。
value に複数の値を指定する場合、" で囲む必要があります。

user_attribute="value1,value2"

本オプションで指定された文字列は、バッチリクエストの動作に影響を与えません。

-v *variable_list*

リクエストを実行するときに使用される環境変数を指定します。*variable_list* の形式は、次のとおりです。

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

variable と *value* の文字列長の合計は 4000 バイト以下でなければなりません。
すでに存在する環境変数名が *variable* に指定された場合、値は上書きされます。
value に複数の値を指定する場合、" で囲む必要があります。

variable="value1,value2"

リクエスト実行時、以下の環境変数が既定値として設定されています。PBS_ENVIRONMENT 、 PBS_JOBID 、 PBS_SUBREQNO 、 PBS_JOBNAME 、 PBS_NODEFILE 以外の環境変数の値は本オプションによって変更することができます。

環境変数名	説明
PBS_ENVIRONMENT	バッチリクエストがバッチ処理であることを示す、"PBS_BATCH"が設定されます。
PBS_JOBID	バッチリクエストのジョブIDが設定されます。
PBS_SUBREQNO	サブリクエスト番号が設定されます（パラメトリックリクエストの場合）
PBS_JOBNAME	バッチリクエスト名が設定されます。
PBS_NODEFILE	リクエストを構成するジョブが実行されているホスト一覧が記載されたファイルへのパスが設定されます。
PBS_O_HOME	クライアントホスト上の環境変数 HOME が設定されます。
PBS_O_HOST	クライアントホスト名が設定されます。
PBS_O_LANG	クライアントホスト上の環境変数 LANG が設定されま

	す。
PBS_O_LOGNAME	クライアントホスト上の環境変数 LOGNAME が設定されます。
PBS_O_MAIL	クライアントホスト上の環境変数 MAIL が設定されます。
PBS_O_PATH	クライアントホスト上の環境変数 PATH が設定されます。
PBS_O_SHELL	クライアントホスト上の環境変数 SHELL が設定されます。
PBS_O_TZ	クライアントホスト上の環境変数 TZ が設定されます。
PBS_O_WORKDIR	クライアントホスト上の作業用ディレクトリを設定します。

-V　バッチリクエスト投入時のすべての環境変数をバッチリクエスト実行環境にエクスポートします。

-w *script_variable_list*

スクリプトファイルの埋め込みオプション行で使用する変数を指定します。この変数は、リクエスト実行時の環境変数としては設定されません。

script_variable_list の形式は、次のとおりです。

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

variable と *value* の文字列長の合計は 4000 バイト以下でなければなりません。

すでに存在する変数名が *variable* に指定された場合、値は上書きされます。

value に複数の値を指定する場合、" で囲む必要があります。

variable="value1,value2"

後述の埋め込みオプションを参照願います。

-y *reservation_id*

本オプションは、JobManipulator の事前予約機能を利用する場合の専用オプションです。

reservation_id で指定された予約区間を使用してバッチリクエストを実行します。

本オプションをパラメトリックリクエストに対して指定することはできません。

詳細は JobManipulator のマニュアルを参照してください。

-Y *deadline_time*

本オプションは、JobManipulator のデッドラインスケジューリングを利用する場合の専用オプションです。

deadline_time で指定された日時をバッチリクエストのデッドライン時刻として指定します。

本オプションは-s オプションと同時に指定することはできません。
本オプションをパラメトリックリクエストに対して指定することはできません。
詳細は JobManipulator のマニュアルを参照してください。
deadline_time の形式は次のとおりです。

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

CC,*YY*,*MM*,*DD*,*hh*,*mm*,*SS* には、各々以下に示す 2 桁の数字を指定します。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時刻がすでに過ぎている場合は、次の日となります。

-z　バッチリクエストが正しく投入されても、それを示すメッセージを表示しません。本オプションが指定されない場合、標準出力にバッチリクエストのリクエスト ID と投入先のキューを表示します。

-Z　バッチリクエストの投入に成功した場合、リクエスト ID のみを出力します。それ以外のメッセージは表示しません。

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

リクエストで実行されるスクリプトにおいて、SIGTERM の捕捉を有効にするか無効にするかを指定します。本オプションに *yes* を指定した場合、スクリプトにおいて SIGTERM の捕捉を有効にします。
オプションの引数には *yes* または *no* を指定します。引数は省略可能で、引数を省略した場合は *yes* を指定したものとみなします。
本オプションを指定しない場合のデフォルトの動作は *no*（無効）です。

--after *request_identifier* [,*request_identifier* ...]

実行順序を指定して、リクエストを投入します。
request_identifier で指定されたリクエストの実行が終了した後に、投入リクエストが実行を開始するようにスケジューリングします。また、*request_identifier* をカンマで区切って複数記述することにより、複数のリクエストの実行終了を待ち合わせることも可能です。
request_identifier には、通常のリクエスト ID、パラメトリックリクエストのリクエスト ID、および、サブリクエストのリクエスト ID が指定できます。また、ワークフロースクリプト内では、リクエスト名を指定することもできます。
指定可能なリクエスト数に制限はありませんが、コマンドラインに指定できる文字数制

限は 4095 文字となりますので、その範囲内に指定する必要があります。

--cancel-after

リクエストが異常終了した際に、後続リクエストを自動的に削除します。

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

カスタムリソース *cr_name* の利用量 *n* を指定します。カンマで区切って（スペースなしで）複数のカスタムリソースの利用量を指定することができます。

n には 1～2147483647 の整数、または unused (0) が指定できます。投入先のキューで利用量制御対象外指定不可となっている場合、unused (0) の指定はできません。

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

クラウドのコンピューティング資源へリクエストをバースティングして実行するかどうかを指定します。*yes* を指定した場合は、バースティングします。*no* を指定した場合は、バースティングしません。本オプションが未指定の場合、既定値は *no* です。

--exclusive {= *host*}

1 ホストあたり 1 ジョブで実行すること（排他実行）を指定してリクエストを投入します。

オプションの引数には"*host*"を指定します。引数の"*host*"は省略可能です。

本オプションとプロビジョニング機能のテンプレート指定（--template）は同時に使用することができません。

--group=*group_name*

リクエストのグループを指定します。*group_name* にはグループ名を指定します。

このオプションの指定が無い場合、リクエストのグループはコマンド実行時のグループです。

なお、「リクエストのグループ指定実行機能」が OFF の場合、本オプションは無効です。

--job-separator

ジョブグループの区切りを示します。---と略記することも可能です。

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

マルチインスタンス GPU (MIG)を指定します。

gi-name には GPU インスタンス(GI)プロファイル名を指定します。 指定可能な GI プロファイル名は qstat -Ef で確認できます。

指定した GI を更にコンピュートインスタンス(CI)単位で複数に分割する場合は、*ci-slice-count* を指定します。*ci-slice-count* を指定しない場合は、GI プロファイルと同じサイズの CI を作成します。

［指定例］

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

この場合は、4g.20gb の GI プロファイルを使用し、GI を更に 3 つの CI に分割して割り当てます。--mig オプションを複数指定することで、複数の GI を割り当てることができます。

--gpunum-lhost と同時指定した場合は、--mig オプションのみが適用され、--gpunum-lhost は無視されます。

--parallel

リクエストの実行の際に複数のリクエストを同時に実行開始するように指定して、リクエストを投入します。

qsub コマンドの最後の引数として 2 つ以上のジョブスクリプトファイルを指定することで、指定したジョブスクリプトをそれぞれ別のリクエストとして投入し、それらのリクエストを同時に実行開始するようにスケジューリングします。

--parallel オプションを指定した場合は、リクエスト連携機能（直列連携、並列連携）の投入は行えません。

指定可能な同時実行リクエスト数に制限はありませんが、コマンドラインに指定できる文字数制限は 4095 文字となりますので、その範囲内に指定する必要があります。

--restart-script={no|job|*script*}

ユーザレベルリスタートスクリプトを *script* で指定します。

no を指定すると、ユーザレベルチェックポイントを取得したリクエストは終了します。

job を指定すると、最初から再実行します。

script は絶対パスで指定します。また、*script* で指定したファイルが存在しない場合でも qsub はエラーになりません。

--use-custom-suspend=chkpnt:*script* を指定せずに、--restart-script を指定すると、エラーが発生します。

このオプションを指定しない場合、既定値は job です。

--template=*template_name*

プロビジョニング環境におけるジョブ実行環境のテンプレートを *template_name* で指定します。

なお、本オプション指定時、同時に-l オプションによる cpunum_job, memsz_job, gpunum_job の指定はできません。リクエストにはテンプレート *template_name* 内で定義された CPU 数、メモリサイズ、GPU 数が適用されます。

--use-custom-suspend={no | chkpnt:*script*}

ユーザレベルチェックポイントのためのスクリプトを chkpnt:*script* で指定します（キーワード chkpnt:は省略できません）。no を指定すると、ユーザレベルチェックポイントをとりません。*script* は絶対パスで指定します。この場合、ホームディレクトリを示す~表

記は使用できません。また、*script* で指定したファイルが存在しない場合でも qsub はエラーになりません。
このオプションを指定しない場合、既定値は no です。

--use-hca=*num*
--use-hca=[{io | mpi | all}:*num*][,{io:mpi | all}:*num...*]
同一 PCIeSW 内の VE が使用する、HCA のポート数を *num* で指定します。
または、HCA の種別(io または mpi または all）を先頭にコロンで区切って *num* を指定することで、HCA の種類（ScaTeFS ダイレクト IO 用・MPI 用・ScaTeFS ダイレクト IO 用と MPI 用の両方）別に HCA のポート数を指定することができます。*num* だけを指定した場合は、all が指定されたものと見做します。
本オプションは VE を使用するリクエストに対してのみ有効です。

--user= *user_name*
リクエストのオーナーを *user_name* として投入します。*user_name* にはユーザー名を指定します。
このオプションを使用するには、グループ管理者権限が必要です。*user_name* に指定できるユーザー名は、グループ管理者としての管理対象グループをプライマリグループとするユーザーのみです。
また、ユーザーマッピングによりローカルとリモートで異なるユーザーにマッピングされているユーザーを指定することはできません。
このオプションの指定が無い場合、リクエストのオーナーはコマンドの実行ユーザーになります。

--userpp-script=*location*:*path*[,*location*:*path*]
location で指定した実行位置で *path* に指定したスクリプトを実行します。
location には、以下の文字列が使用できます。

prr	実行位置が PRE-RUNNING です。
por	実行位置が POST-RUNNING です。

--venode=*total_venum*
total_venum で指定した数の VE ノードをリクエストに対して割り当てます。投入先キューの既定搭載 VE ノード数に従って論理ホスト数は自動的に決定されます。1 以上の値を指定する必要があります。また、--venum-lhost オプションと同時指定した場合、キューの既定搭載 VE ノード数の代わりに--venum-lhost オプションで指定した値を使用して、論理ホスト数が自動設定されます。本オプションと-b オプションによるジョブ数（論理ホスト数）の指定は同時に指定できません。

--venuma={on | off}
リクエストを実行する VE ノードの NUMA モードを指定します。
パーティショニングモード/NUMA モードの詳細は、「SX-Aurora TSUBASA パーティショニングモードのための VEOS NUMA モードガイド」を参照してください。

on　　VE NUMA モードを ON に設定する。
off　　VE NUMA モードを OFF に設定する。

--warning-signal=elapstim:*signal*
経過時間制限の警告値を超過した場合に送信するシグナルを指定します。*signal* は次の形式で指定できます。
・シグナル名（例：SIGHUP）
・シグナル名から SIG を取り除いたもの（例：HUP）
・シグナル番号（例：1）詳細は signal(7)を参照してください。リクエストを実行する VE ノードの NUMA モードを指定します。

省略した場合、SIGTERM シグナルが送信します。

【論理ホスト毎の資源制限オプション】
リクエスト投入時に -b *job_count* 指定で作成された各ジョブ（論理ホスト）について、論理ホスト毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。警告値を省略すると、最大値と同じ値に設定します。
詳細な書式については**資源制限値の形式**をご覧ください。
-l オプション（例：-l cpunum_job）でジョブ/論理ホスト単位の資源制限の最大値や警告値を設定できますが、-l はレガシーの設定であり今後拡張しないため、下記の設定オプションの使用を推奨します。

--cpunum-lhost=*max_limit*
論理ホスト毎の CPU 台数制限値を設定します（-l cpunum_job=*max_limit* CPU 台数制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 台数となります。

--cputim-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の CPU 使用時間制限値を設定します（-l cputim_job=*max_limit* CPU 使用時間制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の CPU 使用時間となります。

--gpunum-lhost=*max_limit*
論理ホスト毎の GPU 台数制限値を設定します（-l gpunum_job=*max_limit* GPU 台数制限値(ジョブ)に相当します)。

--memsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎のメモリサイズ制限値を設定します（-l memsz_job=*max_limit* メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側のメモリサイズとなります。

--stderrsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の標準エラー出力の結果ファイルサイズ制限値を設定します。

--stdoutsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の標準出力の結果ファイルサイズ制限値を設定します。

--vecputim-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

--vememsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

--venum-lhost =*max_limit*
論理ホスト毎の VE ノード数制限値を設定します。

--vmemsz-lhost=*max_limit*[,*warn_limit*]
論理ホスト毎の仮想メモリサイズ制限値を設定します。(-l vmemsz_job=*max_limit* 仮想メモリサイズ制限値(ジョブ)に相当します)。SX-Aurora TSUBASA の場合、論理ホストの VH 側の仮想メモリサイズとなります。
VE プロセス一つあたり、32TB と数百メガバイトの仮想メモリが VH 側に必要になります。

【VE ノード毎の資源制限オプション】
VE ノード毎に資源制限の最大値や警告値を、以下のオプションで指定することができます。
警告値を省略すると、最大値と同じ値に設定します。
詳細な書式については**資源制限値の形式**をご覧ください。

--vecputim-venode=*max_limit*[,*warn_limit*]
VE ノード毎の VE CPU 使用時間制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側の CPU 使用時間となります。

--vememsz-venode=*max_limit*[,*warn_limit*]
VE ノード毎の VE メモリサイズ制限値を設定します。SX-Aurora TSUBASA の場合、論理ホストの VE 側のメモリサイズとなります。

資源制限値の形式

- 時間制限

時間に関する制限は以下の形式で指定します。

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

無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 秒です。

【指定例】

1234:58:21.29	1234 時間 58 分 21.29 秒
59:01	59 分 1 秒
12345	12345 秒
121.1	121.1 秒

・ サイズ制限

サイズに関する制限は以下の形式で指定します。

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

units に指定できる単位は以下のとおりです。

b	バイト
kb	キロバイト(1024 b)
mb	メガバイト(1024 kb)
gb	ギガバイト(1024 mb)
tb	テラバイト(1024 gb)
pb	ペタバイト(1024 tb)
eb	エクサバイト(1024 pb)

units を指定しなかった場合はバイトと解釈されます。無制限の場合は、"unlimited"という文字列を指定します。指定できる最大値は、2^31-1 バイトです。

【指定例】

1234	1234 バイト
1234kb	1234 キロバイト
1234.5gb	1234.5 ギガバイト

数列指定（%n）

-I,-O オプションのクライアントホストのパス名、ジョブ番号および実行ホストのパス名には数列（%n）の指定が可能です。フォーマットは次のとおりです。

%[0<n>]<l>n

<l>:　　{<w> [, <w> ...]}

<w>: <e> | (<e>) | (<e>)*<n> | (<e>)@<n>
<e> : <n> [, <n> ...] | <n>-<n> | <d>
<d >: N
n:　　非負の数字
N:　　リスト要素は無視されます
-:　　連続する数列
(例)
0-2 : 0,1,2
2-0 : 2,1,0
1-1 : 1
1-z : エラー
*:　　数列全体を繰り返します
(例)
(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 : エラー
繰り返し回数は 1 以上でなければエラーとなります
@:　　数列の要素を繰り返します
(例)
(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 : エラー
繰り返し回数は 1 以上でなければエラーとなります

数列は、複数指定することができます。ただし、数列の要素数が等しくない場合、一番少ない要素数で統一されます。数列指定は、%{(0-2)*2}n のように {}で囲む必要があります。

日付パラメータ（%(*START*-*END*)date）

-I,-O オプションのクライアントホストのステージングファイルのパス名において、%(*YYYYMMDD*-*YYYYMMDD*)date の形式で日付の範囲を指定することにより、パラメトリックリクエストの各サブリクエストに対して、指定した連続した日付内の 1 日を対応させて、*YYYYMMDD* の形式の日付文字列に展開します。
サブリクエストに対して日付を対応付ける方法は、次のとおりです。

(1) *YYYYMMDD*-*YYYYMMDD* の形式で指定された連続する日付に対して、先頭の日付から順番に、0 から始まる番号を割り当てます。
(2) (1)で割り当てた番号に対して、パラメトリックリクエストのサブリクエスト番号と同じ番号が割り当てられた日付をそのサブリクエストに対応する日付とします。

（例）qsub で、-t 2-4　-I "/input/data_%(20150101-20150131)date,infile"
のように指定した場合、実行されるサブリクエストは、サブリクエスト番号が 2,3,4 の 3 つで、それぞれのサブリクエストに対して、以下のようにステージングファイルが対応します。

ファイル	割り当て番号	使用するサブリクエスト番号
/input/data_20150101	0	なし
/input/data_20150102	1	なし
/input/data_20150103	2	2
/input/data_20150104	3	3
/input/data_20150105	4	4
:	:	なし
/input/data_20150131	30	なし

日付パラメータは、パラメトリックリクエストの投入時にのみ使用可能です。
また、指定する日付の範囲(*START*-*END*)は、START≦END となるように指定してください。
START、および、*END* に指定できるのは、1970 年 1 月 1 日以降の日付です。
シェルのコマンドラインで指定する場合は、日付の範囲を指定する際の（および）をエスケープしてください。

埋め込みオプション

qsub のコマンドラインで指定するオプションは、埋め込みオプションとしてスクリプトファイル中に記述できます。
埋め込みオプションは、"#PBS" で始まる行に記述し、"PBS"の後に空白もしくはタブを 1 文字以上置いて、文字"-"の付いたオプション記述を置きます。
この埋め込みオプションを識別する文字列 "PBS" は環境変数 PBS_DPREFIX または、qsub の-C オプションで変更できます。ただし、埋め込みオプションに-C が指定された場合は無視します。
コマンド行に埋め込みオプションで指定したものと同じオプションを指定した場合は、コマンド行のオプションが優先されます。
埋め込みオプションが置ける範囲は、スクリプトファイルの先頭から、コメント行（空白文字を除く先頭文字が＃の行）、文字":"から始まる行、および、空行のいずれでもない行が現れるまでです。

-v で指定された環境変数および-w で指定されたスクリプト変数は、埋め込みオプション行内で、${variable}の形式で参照することができます。同一名の変数が存在する場合の優先順位は以下のとおりです。

1. スクリプト内の埋め込みオプション行で使用できる変数（-w 指定）
2. qsub のコマンドオプションで指定された環境変数(-v 指定)
3. qsub のプロセスに設定されている環境変数

【埋め込みオプションの例】

```
#!/bin/sh
#PBS -N Sample_JOB
#PBS -w ELAPS=600
#PBS -m b -m e
#PBS -q batch1 #default
#PBS -l elapstim_req=${ELAPS}
#
```

環境変数

PBS_QUEUE

デフォルトのバッチキュー名

PBS_DPREFIX

スクリプトファイル中の埋め込みオプション

STGDIR

実行ホスト上のステージングファイルを示すときの起点となるディレクトリ

NQSV_PRIV または NQSII_PRIV

qsub 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qdel(1), qstat(1)

注意

1. コマンドラインでシェルでの特殊文字を指定する場合は、エスケープする必要があります。
2. 各オプションの区切り文字として使用している文字(**-I** オプションの " 等)を通常文字として指定する場合、エスケープする必要があります。
3. ファイル名中に /, null 文字, 改行文字は指定できません。
4. MPI のリクエストは、実行ホスト上に対応する MPI 実行環境が無い場合、PRE-RUNNING でエラーとなります。
5. 以下の環境変数についてはバッチリクエスト実行時にジョブサーバが設定を行うため、-V,-v オプションで同じ環境変数を指定した場合、無効になります。
   - ENVIRONMENT
   - PATH

- HOME
- SHELL
- USER
- LOGNAME
- QUEUENAME
- STGDIR
- PBS_JOBID
- TZ (ジョブサーバの環境変数に設定されている場合)

1.17 qwait (1)

名前

qwait —リクエスト状態遷移の待ち合わせ

形式

qwait [-P *privilege*] [-V] [-t *timeout*] [-w *wait_state*] [-f] *request_identifier*

機能説明

request_identifier で指定されたリクエストの状態が *wait_state* で指定された状態になるまで待ち合わせます。 *wait_state* が省略された場合、qwait は指定されたリクエストの状態が EXITING もしくはリクエストが終了するまで待ち合わせます。 qwait は起動されるとスリープし、リクエストが、指定された状態になるとメッセージを表示して終了します。

request_identifier には、通常のリクエスト ID の他、パラメトリックリクエストのリクエスト ID、および、サブリクエストのリクエスト ID が指定できます。
以下の表は、通常のリクエスト、および、パラメトリックリクエストのサブリクエスト単位での待ち合わせに対する、qwait の終了コードと出力を表したものです。

終了コード	qwaitの出力	意味
0	exited+リクエストの終了コード	リクエストは待ち合わせ状態になり、正常に終了しました。
1	killed by signal+シグナル番号	リクエストはシグナルにより終了しました。
2	deleted in the XXXXXX state	リクエストはデリート要求により削除されました。
3	error	リクエストはNQSV内部エラーにより消滅しました。
4	time out	タイムアウト時間に達しました。
5	not found	リクエストは存在しません。
6	qwait error	qwaitコマンド自体のエラーです。 バッチサーバとの接続が切れた等、NQSV/API レベルでのエラー発生時も本終了コードとなります。
7	rerun	リクエストはリラン要求により終了しました。
8	exited by system failure	リクエストがHW障害、または、ノードダウンにより終了しました。
9	超過資源名 + limit exceeded	リクエストが資源制限超過により終了しました。
10	check point suspend normally	ユーザレベルチェックポイントにより正常終了しました。
11	check point suspend abnormally	ユーザレベルチェックポイントにより異常終了しました。

従来、論理ホストごとの資源制限値(CPU 使用時間等)の超過時や Elapse 時間制限の超過時には終了コード 1 を出力していました。この終了コードは、通常ユーザや qsig 等によってシグナルを送信してリクエストを終了させた場合と同じになり、資源制限値の超過によるリクエストの終了かどうかを判断できません。そのため新しい終了コード 9 を設け、資源制限値の超過時にはこの終了コードを出力します。「qwait の出力」の超過資源名は下表の通りです。

資源名	「qwaitの出力」の超過資源名文字列
論理ホスト毎のCPU使用時間	CPU time
論理ホスト毎のメモリサイズ	MEM size
論理ホスト毎の仮想メモリサイズ	VMEM size
論理ホスト毎のVE CPU時間	VE CPU time
論理ホスト毎のVEメモリサイズ	VE MEM size
ELAPSE時間	ELAPSE time
論理ホスト毎の標準出力の結果ファイルサイズ	STDOUT size limit exceeded
論理ホスト毎の標準エラー出力の結果ファイルサイズ	STDERR size limit exceeded
論理ホスト毎の標準出力＋標準エラー出力の結果ファイルサイズ	STDOUT&STDERR size limit exceeded

以下は、qwait が待ち合わせる状態毎の終了コードの対応表です。

待ち合わせ状態／終了コード	0	1	2	3	4	5	6	7	8	9	10	11
exiting	○	○	○	○	○	○	○	○	○	○	○	○
exited	○	○	○	○	○	○	○	×	○	○	○	○

○：取り得る終了コード
×：取り得ない終了コード

また、パラメトリックリクエスト ID を指定して、パラメトリックリクエスト全体に対して待ち合わせを行う場合の終了コードと、出力イメージは以下の通りです。

終了コード	qwaitの出力	意味
0	exited (*N* subrequests)	全サブリクエストが終了しました。 （*N*は、実行したサブリクエスト数）
2	deleted in the XXXXXX state	パラメトリックリクエストがデリート要求により削除されました。 サブリクエスト単位で削除された場合は含みません。
3	error	パラメトリックリクエストがNQSV内部エラーにより消滅しました。

4	time out	タイムアウト時間に達しました。
5	not found	パラメトリックリクエストが存在しません。
6	qwait error	qwaitコマンド自体のエラーです。 バッチサーバとの接続が切れた等、NQSV/API レベルでのエラー発生時を含みます。
7	rerun	パラメトリックリクエストはリラン要求により終了しました。
9	超過資源名 + limit exceeded ※超過資源名の文字列は、通常のリクエストの場合と同様です。	パラメトリックリクエストが資源制限超過により終了しました。

以下のオプションが指定できます。

-f　まだバッチリクエストが投入されていない場合でも終了を待ち合わせます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m　管理者権限
o　操作員権限
g　グループ管理者権限
s　特別利用者権限
u　一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-t *timeout*

timeout で指定した時間をタイムアウト時間とします。単位は秒です。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

-w *wait_state*

リクエストが *wait_state* で指定した状態になるまで待ち合わせます。*wait_state* には、次の指定ができます。

exiting　リクエストが EXITING 状態もしくはジョブが終了するまで待ち合わせま

す。

exited　　リクエストが EXITED 状態になるまで待ち合わせます。

本オプションが指定されない場合、exiting となります。

環境変数

NQSV_PRIV または NQSII_PRIV

qwait 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

qsub(1)

1.18 qwait2(1)

名前

qwait2 ― 複数リクエストの終了待ち合わせ

形式

qwait2　[-P *privilege*] [-V] [-t *timeout*] *requests*

機能説明

qwait2 は、*requests* で指定されたリクエストが終了、または、削除され、EXITED 状態になるまでを待ち合わせます。

requests にはリクエスト ID、またはリクエスト名を指定します。リクエスト名は、qwait2 をワークフロースクリプト内で実行する場合にのみ、指定できます。また、*requests* には以下のように複数のリクエストを指定し、待ち合わせることができます。

req,req,...	*req* にリクエスト ID またはリクエスト名をカンマで区切って指定します。指定したリクエストがすべて終了するまで待ち合わせます（AND 指定）
req:req:...	*req* にリクエスト ID またはリクエスト名をセミコロンで区切って指定します。指定したリクエストのうちのいずれかのリクエストが終了するまで待ち合わせます（OR 指定）

カンマ区切りとセミコロン区切りの両方が指定された場合は、カンマ区切り（AND 指定）を優先して評価します。

リクエスト ID には、通常のリクエスト ID の他、パラメトリックリクエストのリクエスト ID、および、サブリクエストのリクエスト ID が指定できます。パラメトリックリクエストのリクエスト ID が指定された場合は、すべてのサブリクエストの終了を待ち合わせます。
リクエスト名には、通常のリクエスト名の他、パラメトリックリクエストのリクエスト名が指定できます。

qwait2 は、指定されたリクエストの終了状態を標準出力に出力します。出力する終了状態は、以下のとおりです。

終了状態	意味
数値（10進）	リクエストが終了（exit(2)またはシグナル受信で終了）しました。リクエストのジョブスクリプトを実行していたシェルのプロセスの終了コードの値をそのまま10進表記で表示します。 パラメトリックリクエストの場合は、先頭のサブリクエストの終了コードを表示します。

D	リクエストがqdel(1)等で削除されました。 ELAPSE時間制限超過で削除される場合も含みます。
F	リクエストがH/W障害によって終了しました。
C	ユーザレベルチェックポイントにより正常に終了しました。
E	ユーザレベルチェックポイントにより異常な終了をしました。
－	リクエストは未終了です。

複数の待ち合わせを指定した場合は、リクエストの終了状態は指定された順序でスペースにより区切って表示されます。

例えば、1.bsv と 2.bsv の両方が終了する、もしくは 3.bsv が単独で終了する、のいずれか早い方を待ち合わせる場合、以下のコマンドイメージを実行します。

```
$ qwait2 1.bsv,2.bsv:3.bsv
256 - D
```

上記の場合、qwait2 は、3.bsv が削除されたことによって終了しました。表示結果は、1.bsv は終了ステータス 256 で終了、2.bsv は未終了状態、3.bsv は qdel(1)で削除されたことを示します。

qwait2 をワークフロースクリプト内で実行した場合、指定したリクエストが終了していても、終了状態が表示されます。つまり、上記の例で qwait2 がワークフロースクリプト内で実行された場合は、qwait2 実行時にすでに 1.bsv が終了していても上記と同じように表示されます。

qwait2 をワークフロースクリプト以外で実行した場合、または、ワークフロースクリプト内で自ワークフロースクリプト以外のリクエストを待ち合わせた場合、指定したリクエストがすでに終了していれば、リクエストが存在しないためエラーとなります。

qwait2 は、バッチサーバを再起動した場合もリクエストの終了を待ち合わせます。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m　管理者権限
o　操作員権限
g　グループ管理者権限
s　特別利用者権限
u　一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-t *timeout*

timeout で指定した時間をタイムアウト時間とします。単位は秒です。指定した時間が経過した時点で qwait2 は待ち合わせを終了します。
本オプションが指定されない場合、タイムアウト時間は、環境変数 NQSV_QWAIT_TIMEOUT または NQSII_QWAIT_TIMEOUT で指定された時間になります。環境変数が設定されていない場合のタイムアウト時間は、1 時間です。タイムアウト時間が経過して qwait2 が終了する場合も各リクエストの実行・終了状態を表示します。したがって、*timeout* に 0 が指定された場合は、各リクエストのその時点での実行・終了状態が表示されます。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

qwait2 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

NQSV_QWAIT_TIMEOUT または NQSII_QWAIT_TIMEOUT　　タイムアウト時間

関連項目

qdel(1), qwait(1), wstart(1)

終了コード

qwait2 の終了コードは以下のとおりです。

0	指定されたリクエストの終了、または、削除を待ち合わせて終了しました
1	*requests* で指定したリクエストが存在しない、またはオプション指定誤りなどのエラーで終了しました
2	*timeout* で指定したタイムアウト時間に達したことで終了しました

3　　上記以外のエラーで終了しました

qwait2 がエラーで終了する場合は、待ち合わせを行っていたリクエストの終了状態は表示されません。

1.19 scacctjob/racctjob(1)

名前

scacctjob,racctjob -- ジョブアカウンティング情報の出力

形式

/opt/nec/nqsv/bin/scacctjob [options]
/opt/nec/nqsv/bin/racctjob acct-server [options]

機能説明

scacctjob コマンドは、NQSV のジョブアカウンティング情報を出力します。

racctjob コマンドは、アカウンティングサーバにアクセスして、NQSV のジョブアカウンティング情報を出力します。第一引数にアカウンティングサーバのホスト名を指定し、第二引数以降に scacctjob コマンドと同様なオプションを指定します。

既定値では下記の項目が出力されます。

JOB-ID	ジョブ ID
REQUEST ID	リクエスト ID
REQUEST NAME	リクエスト名
USER NAME	投入ユーザ名
HOST NAME	実行ホスト名
QUEUE NAME	投入キュー名
QUEUED TIME	投入時刻
START TIME	開始時刻
END TIME	終了時刻
CPU(SECS)	CPU 消費時間
REAL(SECS)	ジョブの実時間

オプションの指定によりさらに、下記の項目が出力されます。

GROUP NAME	グループ名
ACCOUNT CODE	アカウントコード
REQUEST PRTY	リクエストの優先度
NICE	ナイス値
KCORE MIN	延べ使用メモリ量（KB*分）
MEAN SIZE(K)	平均使用メモリ
MAXMEM SIZE(K)	最大使用メモリ
V KCORE MIN	延べ使用仮想メモリ量（KB*分）
V MEAN SIZE(K)	平均使用仮想メモリ

V MAXMEM SIZE(K)	最大使用仮想メモリ
REQELAPS TIME(S)	経過時間制限値
REQCPU TIME(S)	ジョブの CPU 使用時間制限値
REQCPU NUM	ジョブの同時実行 CPU 台数制限値
REQMEM SIZE(K)	ジョブのメモリサイズ制限値
REQGPU NUM	ジョブの同時実行 GPU 台数制限値
EXIT STAT	終了ステータス
QUE TYPE	キュータイプ
WAIT TIME	待ち時間（実行開始予定時刻と実行開始時刻の差）
REQVE NUM	ジョブに割り当てた VE ノード数（*1）
RSVVE NUM	ジョブに確保した VE ノード数（*1）
VE CPU(SECS)	VE の CPU 消費時間（ユーザ）（*1）
VE KCORE MIN	VE の延べ使用メモリ（*1）
VE MEAN SIZE(K)	VE の平均使用メモリ（*1）
VE MAXMEM SIZE(K)	VE の最大使用メモリ（*1）
VE REQ NODELIST	ジョブに割り当てた VE ノード番号のリスト（*1）
VE USE NODELIST	ジョブで使用した VE ノード番号のリスト（*1）
VE RSV NODELIST	ジョブに確保した VE ノード番号のリスト（*1）
RESERVATION ID	ジョブのリソース予約区間 ID
crname　（*2）	カスタムリソースの宣言量。 カスタムリソースの消費単位が job の場合のみ値が出力されます
crname TOTAL	カスタムリソースののべ使用量。カスタムリソースの check_mode が moment または integrate の場合に値が出力されます
crname MEAN	カスタムリソースの平均使用量。カスタムリソースの check_mode が moment の場合にのみ値が出力されます
crname MAX	カスタムリソースの最大使用量。カスタムリソースの check_mode が moment の場合にのみ値が出力されます
UNIT	カスタムリソースの単位
HW FAILURE	HW 障害フラグ
VE TIME SLICE	VE のタイムスライス値（*3）
VE MFLOPS	VE の MFLOPS 値（*3）
VE MFLOPS(REAL)	VE の MFLOPS(REAL)値（*3）
VE FPEC(K)	VE の浮動小数点データ実行要素数（*3）
VE SYSTEM CALL	VE のシステムコール回数（*3）
VE-VH DATA TRANS	VE-VH 間データ転送量（*3）
NUMBER OF VE PROCESS	VE の実行プロセス数（*3）
VE INSTRCT(K)	VE の命令実行数（*3）
VEC-INST(K)	ベクトル命令実行数（*3）
VEC-ELMT(K)	ベクトル要素数（*3）
VEC-EXE(SECS)	ベクトル命令実行時間（*3）

VEC-INT LOAD(K)	ベクトルロード実行要素数（*3）
VE L1 CACHE MISS TIME(SECS)	VE の L1 キャッシュミス時間（*3）
VE CPU PORT CONFLICT(SECS)	VE の CPU ポート競合時間（*3）
VEC ARITH EXEC TIME(SECS)	ベクトル演算実行時間（*3）
VEC LOAD EXEC TIME(SECS)	ベクトルロード実行時間（*3）
VE POWER THROTTLING(SECS)	VE の電力要因による HW 停止時間（*3）
VE THERMAL THROTTLING(SECS)	VE の温度要因による HW 停止時間（*3）
VE MAX NTASK	VE の最大タスク数（*3）
VEC-OPERAND RATIO	ベクトル化率（*3）
NUMA NODE	NUMA ノード番号（*3）
GPU NUM PROCS	GPU プロセス数（*4）
GPU ENERGY CONSUMED	GPU 消費電力量（*4）
GPU MEAN POWER USAGE	GPU 平均消費電力（*4）
GPU MAX POWER USAGE	GPU 最大消費電力（*4）
GPU MAX MEM USED	最大使用 GPU メモリ量（*4）
GPU MEAN SM CLOCK	GPU 平均周波数（*4）
GPU MAX SM CLOCK	GPU 最大周波数（*4）
GPU MEAN SM UTIL	GPU 平均使用率（*4）
GPU MAX SM UTIL	GPU 最大使用率（*4）
GPU MEAN MEM UTIL	GPU 平均メモリ使用率（*4）
GPU MAX MEM UTIL	GPU 最大メモリ使用率（*4）

*1 これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

*2 *crname* は、カスタムリソース名になります。

*3 これらの項目は、実行ホストが SX-Aurora TSUBASA システムでかつ VEOS のプロセスアカウント出力が有効な場合のみ有効です。

*4 これらの項目は、実行ホストにおいて GPU アカウント出力が有効な場合のみ有効です。

また、下記のオプションが指定可能です。

-N　　scacctjob のバージョン情報を出力します。
　　他のオプションを指定した場合でも、本オプションが優先され他の
　　オプションはすべて無視されます。

-P *priv*
　　コマンド実行時のアクセス権を指定します。priv には、次を指定します。

　　m 管理者権限
　　o　　操作員権限

g　　グループ管理者権限
s　　特別利用者権限
u　　一般利用者権限

本オプションが指定されない場合、コマンド実行時のアクセス権は、
環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。
環境変数も設定されていない場合は、一般ユーザ権限となります。
環境変数には、以下の指定が可能です。ただし、racctjob は環境変数は
使用できません。

PRIV_MGR　　管理者権限
PRIV_OPE　　操作員権限
PRIV_GMGR　グループ管理者権限
PRIV_SPU　　特別利用者権限
PRIV_USR　　一般利用者権限

-Y　　グループ名を表示します。

-X　　アカウントコードを表示します。

-t　　CPU 消費時間をユーザ/システム別に表示します。

-k　　延べ使用メモリ量（KCORE MIN）を表示します。

-m　　平均使用メモリ（キロバイト）を表示します。

-M　　最大使用メモリ（キロバイト）を表示します。

--vmem 延べ使用仮想メモリ量（KCORE MIN）、平均使用仮想メモリ（キロバイト）、最大使用仮想メモリ（キロバイト）を表示します。

-T　　経過時間制限値（秒）を表示します。

-q　　ジョブの CPU 使用時間制限値（秒）、ジョブの同時実行 CPU 台数制限値、
ジョブのメモリサイズ制限値（キロバイト）を表示します。

-G　　ジョブの同時実行 GPU 台数制限値を表示します。

-p　　リクエストの優先度、ナイス値、タイムスライス値を表示します。

-h　　終了ステータスを表示します。

-2　　時刻表示に日付を付加します。

-3　　人間が読みやすい形式で表示します。

-K　　キュータイプを表示します。

-w　　待ち時間を表示します。リクエストの投入時刻から実行開始時刻までは待ち時間となります。qmove でリクエストを別のキューに移動した場合でもリクエスト投入時点から待ち時間を計算します。なお、qsub で-a オプションをつけてリクエストのスケジューリング時刻を指定してリクエストを投入した場合は、-a で指定した時刻は計算の起点となります。

-V　　ジョブに割り当てた VE ノード数、ジョブに確保した VE ノード数、VE の CPU 消費時間（ユーザ）(秒)、VE の延べ使用メモリ（キロバイト・分)、VE の平均使用メモリ（キロバイト)、VE の最大使用メモリ（キロバイト）を表示します。本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

-Z　　ジョブに割り当てた VE ノード番号のリスト、ジョブで使用した VE ノード番号のリスト、ジョブに確保した VE ノード番号のリストを表示します。本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

-F　　ジョブのカスタムリソースの宣言量を表示します。使用していないカスタムリソースは UNUSED と表示されます。本オプションは、-B オプションとの併用はできません。

-B *crname1[,crname2...]*

指定されたカスタムリソースの宣言量を表示します。使用していないカスタムリソースは UNUSED と表示されます。本オプションは、-F オプションとの併用はできません。

-O　　カスタムリソースののべ使用量、平均使用量および最大使用量を表示します。使用していないカスタムリソースは UNUSED と表示されます。本オプションは、-j オプションとの併用はできません。

-j *crname1[,crname2...]*

指定されたカスタムリソースののべ使用量、平均使用量および最大使用量を表示します。使用していないカスタムリソースは UNUSED と表示されます。本オプションは、-O オプションとの併用はできません。

--veps

VE の CPU 消費時間（ユーザ）（秒）、VE の延べ使用メモリ（キロバイト・分）、VE の平均使用メモリ（キロバイト）を表示します。本オプションは-V オプションとの併用はできません。本オプションで表示する値は VEOS から出力したプロセスアカウントをもとに計算した値です。

--veps-perform

VE のタイムスライス値、MFLOPS 値、MFLOPS(REAL)値を表示します。

--veps-prog

VE の浮動小数点データ実行要素数(K)、システムコール回数、VE-VH 間データ転送量、実行プロセス数を表示します。

--veps-inst

VE の実行命令数(K)、ベクトル命令実行数(K)、ベクトル要素数(K)、ベクトル命令実行時間（秒）、ベクトルロード実行要素数（K）を表示します。

--veps-time

VE の L1 キャッシュミス時間（秒）、CPU ポート競合時間（秒）、ベクトル演算実行時間（秒）、ベクトルロード実行時間（秒）、電力要因による HW 停止時間（秒）、温度要因による HW 停止時間（秒）を表示します。

--veps-task

VE の最大タスク数を表示します。

--veps-opratio

VE のベクトル化率を表示します。

--veps-numanode

VE の NUMA ノード番号を表示します。

--gpu

GPU プロセス数、GPU 消費電力量、GPU 平均消費電力、GPU 最大消費電力、最大使用 GPU メモリ量、GPU 平均周波数、GPU 最大周波数、GPU 平均使用率、GPU 最大使用率、GPU 平均メモリ使用率、GPU 最大メモリ使用率を表示します。

-R　　1 ジョブの情報を複数行で出力します。

-v　　見出し行を表示しません。

-r　　ジョブのリソース予約区間 ID を表示します。リソース予約区間 ID が指定されていない場合は、"-"と表示されます。

-u *user*

指定されたユーザに属するジョブの情報を出力します。ユーザ引数には、ログイン名, ユーザ ID のどちらかが指定可能です。

-g *group*

指定されたグループで実行されたジョブの情報を出力します。

-x *acctcode*

指定されたアカウントコードで実行されたジョブの情報を出力します。

-I *request-ID*

指定されたリクエスト ID を持つジョブの情報を出力します。

-s *time*

指定された時刻あるいはそれ以降に投入されたジョブの情報を出力します。時刻の指定形式は以下の通りです。

hour[:min[:sec[, MM/DD[/YY]]]]

（MM： 月, DD： 日, YY： 西暦の下二桁）

-e *time*

指定された時刻あるいはそれ以前に投入されたジョブの情報を出力します。

-S *time*

指定された時刻あるいはそれ以降に実行を開始したジョブの情報を出力します。

-E *time*

指定された時刻あるいはそれ以前に実行が終了したジョブの情報を出力します。

-A *time*

指定された時刻あるいはそれ以降に実行が終了したジョブの情報を出力します。

-b *time*

指定された時刻あるいはそれ以前に実行を開始したジョブの情報を出力します。

-n *request-name*

指定されたリクエスト名と同じリクエスト名を持つジョブの情報を出力します。

-Q *queue*

指定されたキューに登録されたジョブの情報を出力します。

-W *type*

type に master が指定された時はジョブ番号 0 のジョブのみを表示し、type に slave が指定された時は、ジョブ番号 1 以上のジョブのみを表示します。

-l *hostname*

指定したホスト名で実行されたジョブを表示します。

--hw-failure

HW 障害フラグを表示します。

1　ジョブ実行中に JSV の LINKDOWN が発生した

20　ジョブ終了時のノードヘルスチェックで障害を検知した

21　前述の両方が発生した場合

--long-request-name

出力される内容を規定の長さで切ることなく全て表示します。

-o *sortname*

グループ管理者権限以外で実行した場合は指定された項目でソートします。グループ管理者権限で実行した場合はグループ名毎に指定された項目でソートします。*sortname* には以下の指定が可能です。

JOB_ID	JOB ID でソートします
REQUEST_ID	REQUEST-ID でソートします
NAME	REQUEST-NAME でソートします
USER_NAME	USER NAME でソートします
GROUP_NAME	GROUP NAME でソートします
ACCOUNT_CODE	ACCOUNT CODE でソートします
HOST_NAME	HOST NAME でソートします
QUEUE_NAME	QUEUE NAME でソートします

QUEUED_TIME	QUEUED TIME でソートします
START_TIME	START TIME でソートします
END_TIME	END TIME でソートします
CPU	CPU (SECS) でソートします
CPU_SYS	CPU (SECS) SYS でソートします
CPU_USER	CPU (SECS) USER でソートします
REAL	REAL (SECS) でソートします
REQUEST_PRTY	REQUEST PRTY でソートします
NICE	NICE でソートします
REQELAPS_TIME	REQELAPS TIME(S) でソートします
REQCPU_TIME	REQCPU TIME(S) でソートします
REQCPU_NUM	REQCPU NUM でソートします
REQMEM_SIZE	REQMEM SIZE(K) でソートします
REQGPU_NUM	REQGPU NUM でソートします
EXIT_STAT	EXIT STAT でソートします
KCORE_MIN	KCORE MIN でソートします
MEAN_SIZE	MEAN SIZE(K) でソートします
MAXMEM_SIZE	MAXMEM SIZE(K) でソートします
V_KCORE_MIN	V KCORE MIN でソートします
V_MEAN_SIZE	V MEAN SIZE(K) でソートします
V_MAXMEM_SIZE	V MAXMEM SIZE(K) でソートします
QUE_TYPE	QUE TYPE でソートします
WAIT_TIME	WAIT TIME(SEC) でソートします
REQVE_NUM	REQVE NUM でソートします (*1)
RSVVE_NUM	RSVVE NUM でソートします (*1)
VE_CPU	VE CPU(S)でソートします (*1)(*2)
VE_KCORE_MIN	VE KCORE MIN でソートします (*1)(*2)
VE_MEAN_SIZE	VE MEAN SIZE(K)でソートします (*1)(*2)
VE_MAXMEM_SIZE	VE MAXMEM SIZE(K)でソートします (*1)
RES_ID	RESERVATION ID でソートします
HW_FAILURE	HW FAILURE でソートします
VE_CPU_PS	VE CPU(SECS)でソートします (*1) (*3)
VE_KCORE_MIN_PS	VE KCORE MIN でソートします (*1) (*3)
VE_MEAN_SIZE_PS	VE MEAN SIZE(K)でソートします (*1) (*3)
VE_TIME_SLICE	VE TIME SLICE でソートします (*1)
VE_MFLOPS	VE MFLOPS でソートします (*1)
VE_MFLOPS_REAL	VE MFLOPS(REAL)でソートします (*1)
VE_FPEC	VE FPEC(K)でソートします (*1)
VE_SYSTEM_CALL	VE SYSTEM CALL でソートします (*1)
VE_VH_DATA_TRANS	VE-VH DATA TRANS でソートします (*1)

VE_PROC_NUM	NUMBER OF VE PROCESS でソートします (*1)
VE_INSTRCT	VE INSTRCT(K)でソートします (*1)
VE_INST	VE-INST(K)でソートします (*1)
VEC_ELMT	VEC-ELMT(K)でソートします (*1)
VEC_EXE	VEC-EXE(SECS)でソートします (*1)
VE_INST_LOAD	VEC-INST LOAD(K)でソートします (*1)
VE_L1_CACHE_MISS_TIME	VE L1 CACHE MISS TIME(SECS) でソートします (*1)
VE_CPU_PORT_CONFLICT	VE CPU PORT CONFLICT(SECS) でソートします (*1)
VEC_ARITH_EXEC_TIME	VEC ARITH EXEC TIME(SECS) でソートします (*1)
VEC_LOAD_EXEC_TIME	VEC LOAD EXEC TIME(SECS) でソートします (*1)
VE_PORT_THROTTLING	VE POWER THROTTLING(SECS) でソートします (*1)
VE_THERMAL_THROTTLING	VE THERMAL THROTTLING(SECS) でソートします (*1)
VE_MAX_NTASK	VE MAX NTASK でソートします (*1)
VEC_OP_RATIO	VEC-OPERAND RATIO でソートします (*1)
GPU_NUM_PROCS	GPU プロセス数でソートします (*4)
GPU_ENERGY_CONSUMED	GPU 消費電力量でソートします (*4)
GPU_MEAN_POWER_USAGE	GPU 平均消費電力でソートします (*4)
GPU_MAX_POWER_USAGE	GPU 最大消費電力でソートします (*4)
GPU_MAX_MEM_USED	最大使用 GPU メモリ量 でソートします (*4)
GPU_MEAN_SM_CLOCK	GPU 平均周波数 でソートします (*4)
GPU_MAX_SM_CLOCK	GPU 最大周波数でソートします (*4)
GPU_MEAN_SM_UTIL	GPU 平均使用率でソートします (*4)
GPU_MAX_SM_UTIL	GPU 最大使用率でソートします (*4)
GPU_MEAN_MEM_UTIL	GPU 平均メモリ使用率 でソートします (*4)
GPU_MAX_MEM_UTIL	GPU 最大メモリ使用率 でソートします (*4)

*1 これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

*2 これらの項目は、-V オプションで表示する VE CPU(SECS)、VE KCORE MIN、VE MEAN SIZE(K)の値でソートします。

*3 これらの項目は、--veps オプションで表示する VE CPU(SECS)、VE KCORE MIN、VE MEAN SIZE(K)の値でソートします。

*4 これらの項目は、実行ホストにおいて GPU アカウント出力が有効な場合のみ有効です。

-d *database[,database...]*

出力の対象となるデータベースのファイル名を指定します。このオプションは、レガシーオプションです。NQSV R1.02 以前のデータベースを参照する場合に使用します。NQSV R1.04 以降の場合は、--db-dir オプションを使用してください。

--db-dir *database-dir[,database-dir...]*

出力の対象となるデータベースがあるディレクトリ名を指定します。本オプションおよび-d オプションが指定されない場合、デフォルトのデータベース(/var/opt/nec/nqsv/asv/配下の jacct および jacct_cr)となります。

関連項目

scacctreq(1-N), scacctrsv(1-N), racctreq(1-N), racctrsv(1-N).

1.20 scacctreq/racctreq(1)

名前

scacctreq,racctreq - リクエストアカウンティング情報の出力

形式

/opt/nec/nqsv/bin/scacctreq [options]
/opt/nec/nqsv/bin/racctreq acct-server [options]

機能説明

scacctreq コマンドは、NQSV のリクエストアカウンティング情報を出力します。

racctreq コマンドは、アカウンティングサーバにアクセスして、NQSV の
リクエストアカウンティング情報を出力します。第一引数にアカウンティング
サーバのホスト名を指定し、第二引数以降に scacctreq コマンドと同様なオプションを
指定します。

既定値では下記の項目が出力されます。

REQUEST ID	リクエスト ID
REQUEST NAME	リクエスト名
USER NAME	投入ユーザ名
QUEUE NAME	投入キュー名
QUEUED TIME	投入時刻
START TIME	開始時刻
END TIME	終了時刻
CPU(SECS)	CPU 消費時間
REAL(SECS)	リクエストの実時間
STATUS	終了ステータス（簡易表示）

オプションの指定によりさらに、下記の項目が出力されます。

PRERUN COUNT	親リクエストの RERUN 回数(サブリクエストのみ有効)
RERUN COUNT	リクエストの RERUN 回数
GROUP NAME	グループ名
ACCOUNT CODE	アカウントコード
REQUEST PRTY	リクエストの優先度
NICE	ナイス値
KCORE MIN	延べ使用メモリ量（KB*分）

MEAN SIZE(K)	平均使用メモリ
MAXMEM SIZE(K)	最大使用メモリ
V KCORE MIN	延べ使用仮想メモリ量（KB*分）
V MEAN SIZE(K)	平均使用仮想メモリ
V MAXMEM SIZE(K)	最大使用仮想メモリ
REQELAPS TIME(S)	経過時間制限値
REQCPU TIME(S)	ジョブ毎の CPU 使用時間制限値。 ハイブリッドリクエストの場合、リクエストの各ジョブの CPU 使用時間制限値の平均値
REQCPU NUM	ジョブ毎の同時実行 CPU 台数制限値。 ハイブリッドリクエストの場合、リクエストの各ジョブの同時実行 CPU 台数制限値の平均値
REQMEM SIZE(K)	ジョブ毎のメモリサイズ制限値。 ハイブリッドリクエストの場合、リクエストの各ジョブのメモリサイズ制限値の平均値
REQGPU NUM	ジョブ毎の同時実行 GPU 台数制限値。 ハイブリッドリクエストの場合、リクエストの各ジョブの同時実行 GPU 台数制限値の平均値
EXIT STAT	終了ステータス
QUE TYPE	キュータイプ
JOBS	ジョブ数
SUBREQ	サブリクエスト数
EHOST NUM	実行ホスト数
TEMPLATE NAME	テンプレート名（*2)
REQVE NUM	要求した VE ノード数（*1)
RSVVE NUM	確保した VE ノード数（*1)
VE CPU(S)	VE の CPU 消費時間（ユーザ）(*1)
VE KCORE MIN	VE の延べ使用メモリ（*1)
VE MEAN SIZE(K)	VE の平均使用メモリ（*1)
VE MAXMEM SIZE(K)	VE の最大使用メモリ（*1)
REQCPUTIME SUM(S)	リクエストの各ジョブの CPU 使用時間制限値の合計
REQCPUNUM SUM	リクエストの各ジョブの同時実行 CPU 台数制限値の合計
REQMEMSIZE SUM(M)	リクエストの各ジョブのメモリサイズ制限値の合計
REQGPUNUM SUM	リクエストの各ジョブの同時実行 GPU 台数制限値の合計
crname　(*3)	カスタムリソースの宣言量。カスタムリソースの消費単位が request の場合は、リクエスト全体のカスタムリソースの宣言量、消費単位が job の場合は、ジョブ毎のカスタムリソースの宣言量
crname TOTAL	カスタムリソースののべ使用量。カスタムリソースの check_mode が moment または integrate の場合に値が

	出力されます
crname MEAN	カスタムリソースの平均使用量。カスタムリソースの check_mode が moment の場合にのみ値が出力されます
crname MAX	カスタムリソースの最大使用量。カスタムリソースの check_mode が moment の場合にのみ値が出力されます
UNIT	カスタムリソースの単位
HW FAILURE	HW 障害フラグ
VE TIME SLICE	VE のタイムスライス値（*4）
VE MFLOPS	VE の MFLOPS 値（*4）
VE MFLOPS(REAL)	VE の MFLOPS(REAL)値（*4）
VE FPEC(K)	VE の浮動小数点データ実行要素数（*4）
VE SYSTEM CALL	VE のシステムコード回数（*4）
VE-VH DATA TRANS	VE-VH 間データ転送量（*4）
NUMBER OF VE PROCESS	VE の実行プロセス数（*4）
VE INSTRCT(K)	VE の命令実行数（*4）
VEC-INST(K)	ベクトル命令実行数（*4）
VEC-ELMT(K)	ベクトル要素数（*4）
VEC-EXE(SECS)	ベクトル命令実行時間（*4）
VEC-INT LOAD(K)	ベクトルロード実行要素数（*4）
VE L1 CACHE MISS TIME(SECS)	VE の L1 キャッシュミス時間（*4）
VE CPU PORT CONFLICT(SECS)	VE の CPU ポート競合時間（*4）
VEC ARITH EXEC TIME(SECS)	ベクトル演算実行時間（*4）
VEC LOAD EXEC TIME(SECS)	ベクトルロード実行時間（*4）
VE POWER THROTTLING(SECS)	VE の電力要因による HW 停止時間（*4）
VE THERMAL THROTTLING(SECS)	VE の温度要因による HW 停止時間（*4）
VE MAX NTASK	VE の最大タスク数（*4）
VEC-OPERAND RATIO	ベクトル化率（*4）
GPU NUM PROCS	GPU プロセス数（*5）
GPU ENERGY CONSUMED	GPU 消費電力量（*5）
GPU MEAN POWER USAGE	GPU 平均消費電力（*5）
GPU MAX POWER USAGE	GPU 最大消費電力（*5）
GPU MAX MEM USED	最大使用 GPU メモリ量（*5）
GPU MEAN SM CLOCK	GPU 平均周波数（*5）
GPU MAX SM CLOCK	GPU 最大周波数（*5）
GPU MEAN SM UTIL	GPU 平均使用率（*5）
GPU MAX SM UTIL	GPU 最大使用率（*5）
GPU MEAN MEM UTIL	GPU 平均メモリ使用率（*5）
GPU MAX MEM UTIL	GPU 最大メモリ使用率（*5）

*1 これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

*2 これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。
*3 *crname* はカスタムリソース名です。
*4 これらの項目は、実行ホストが SX-Aurora TSUBASA システムでかつ VEOS のプロセスアカウント出力が有効な場合のみ有効です。
*5 これらの項目は、実行ホストにおいて GPU アカウント出力が有効な場合のみ有効です。

また、下記のオプションが指定可能です。

-N　scacctreq のバージョン情報を出力します。
他のオプションを指定した場合でも、本オプションが優先され他のオプションはすべて無視されます。

-P *priv*
コマンド実行時のアクセス権を指定します。priv には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数も設定されていない場合は、一般ユーザ権限となります。
環境変数には、以下の指定が可能です。ただし、racctreq は環境変数は使用できません。

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

-r　リクエストの RERUN 回数およびパラメトリックリクエストの RERUN 回数を表示します。パラメトリックリクエストの場合、サブリクエスト単位での RERUN 回数が RERUN COUNT に、パラメトリックリクエスト全体の RERUN 回数が PRERUN COUNT に表示されます。パラメトリックリクエストでない場合、PRERUN COUNT は 0 になります。

-Y　グループ名を表示します。

-X　　アカウントコードを表示します。

-t　　CPU 消費時間をユーザ/システム別に表示します。

-k　　延べ使用メモリ（KCORE MIN）を表示します。

-m　　平均使用メモリサイズ（キロバイト）を表示します。

-M　　最大使用メモリサイズ（キロバイト）を表示します。

--vmem 延べ使用仮想メモリ量（KCORE MIN）、平均使用仮想メモリ（キロバイト）、最大使用仮想メモリ（キロバイト）を表示します。

-T　　経過時間制限値（秒）を表示します。

-q　　ジョブ毎の CPU 使用時間制限値（秒）、ジョブ毎の同時実行 CPU 台数制限値、メモリサイズ制限値（キロバイト）、ジョブ毎の同時実行 GPU 台数制限値を表示します。ハイブリッドリクエストの場合は、各リソースのリクエストの各ジョブの平均値を表示します。

-z　　リクエストの各ジョブの CPU 使用時間制限値（秒）の合計、同時実行 CPU 台数制限値の合計、メモリサイズ制限値の合計（キロバイト）、同時実行 GPU 台数制限値の合計を表示します。

-p　　リクエストの優先度、ナイス値を表示します。

-h　　終了ステータスをより詳細に表示します。

-2　　時刻表示に日付を付加します。

-3　　人間が読みやすい形式で表示します。

-K　　キュータイプを表示します。

-C　　実行ノード数、ジョブ数、サブリクエスト数を表示します。

-L　　テンプレート名を表示します。本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

-V　　要求した VE ノード数、確保した VE ノード数、VE の CPU 消費時間（ユーザ）（秒）、VE の延べ使用メモリ（キロバイト・分）、VE の平均使用メモリ（キロバイト）、VE の最大使用メモリ（キロバイト）を表示します。本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

-F　　カスタムリソースの宣言量を表示します。使用していないカスタムリソースは UNUSED と表示されます。本オプションは、-G オプションとの併用はできません。

-G *crname1[,crname2...]*

指定されたカスタムリソースの宣言量を表示します。使用していないカスタムリソースは UNUSED と表示されます。
本オプションは、-F オプションとの併用はできません。

-O　　カスタムリソースののべ使用量、平均使用量および最大使用量を表示します。
使用していないカスタムリソースは UNUSED と表示されます。
本オプションは、-j オプションとの併用はできません。

-j *crname1[,crname2...]*

指定されたカスタムリソースののべ使用量、平均使用量および最大使用量を表示します。使用していないカスタムリソースは UNUSED と表示されます。
本オプションは、-O オプションとの併用はできません。

--veps

VE の CPU 消費時間（ユーザ）（秒）、VE の延べ使用メモリ（キロバイト・分）、VE の平均使用メモリ（キロバイト）を表示します。
本オプションは-V オプションとの併用はできません。本オプションで表示する値は VEOS から出力したプロセスアカウントをもとに計算した値です。

--veps-perform

VE のタイムスライス値、MFLOPS 値、MFLOPS(REAL)値を表示します。

--veps-prog

VE の浮動小数点データ実行要素数(K)、システムコード回数、VE-VH 間データ転送量、実行プロセス数を表示します。

--veps-inst

VE の実行命令数(K)、ベクトル命令実行数(K)、ベクトル要素数(K)、ベクトル命令実行時間（秒）、ベクトルロード実行要素数（K）を表示します。

--veps-time

VE の L1 キャッシュミス時間（秒）、CPU ポート競合時間（秒）、ベクトル演算実行時間（秒）、ベクトルロード実行時間（秒）、電力要因による HW 停止時間（秒）、温度要因による HW 停止時間（秒）を表示します。

--veps-task
VE の最大タスク数を表示します。

--veps-opratio
VE のベクトル化率を表示します。

--gpu
GPU プロセス数、GPU 消費電力量、GPU 平均消費電力、GPU 最大消費電力、最大使用 GPU メモリ量、GPU 平均周波数、GPU 最大周波数、GPU 平均使用率、GPU 最大使用率、GPU 平均メモリ使用率、GPU 最大メモリ使用率を表示します。

-R　　1 リクエストの情報を複数行で出力します。

-v　　見出し行を表示しません。

-u *user*
指定されたユーザに属するリクエストの情報を出力します。
ユーザ引数には、ログイン名，ユーザ ID のどちらかが指定可能です。

-g *group*
指定されたグループで実行されたリクエストの情報を出力します。

-x *acctcode*
指定されたアカウントコードで実行されたリクエストの情報を出力します。

-I *request-ID*
指定されたリクエスト ID を持つリクエストの情報を出力します。

-s *time*
指定された時刻あるいはそれ以降に投入されたリクエストの情報を出力します。時刻の指定形式は以下の通りです。

hour[:min[:sec[, MM/DD[/YY]]]]
（MM： 月, DD： 日, YY： 西暦の下二桁）

-e *time*

指定された時刻あるいはそれ以前に投入されたリクエストの情報を出力します。

-S *time*

指定された時刻あるいはそれ以降に実行を開始したリクエストの情報を出力します。

-E *time*

指定された時刻あるいはそれ以前に実行が終了したリクエストの情報を出力します。

-A *time*

指定された時刻あるいはそれ以降に実行が終了したリクエストの情報を出力します。

-b *time*

指定された時刻あるいはそれ以前に実行を開始したリクエストの情報を出力します。

-n *request-name*

指定されたリクエスト名と同じリクエスト名を持つリクエストの情報を出力します。

-Q *queue*

指定されたキューに登録されたリクエストの情報を出力します。

-l *templatename*

指定されたテンプレート名を持つリクエストの情報を出力します。
本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

-c *crname*

指定されたカスタムリソース情報があるリクエストの情報を出力します。

--hw-failure

HW 障害フラグを出力します。

1　リクエストのいずれかのジョブ実行中に JSV の LINKDOWN が発生した

20　リクエストのいずれかのジョブ終了時のノードヘルスチェックで障害を検知した

21　前述の両方が発生した場合

--long-request-name

出力される内容を規定の長さで切ることなく全て表示します。

-o *sortname*

グループ管理者権限以外で実行した場合は指定された項目でソートします。グループ管理者権限で実行した場合はグループ名毎に指定された項目でソートします。カスタムリソースでソートする場合はカスタムリソースの名前の前に”CR_”を付けて指定します。-o でカスタムリソースを指定した場合、-c オプションは無視されます。なお、カスタムリソースの消費量（宣言値）のみソートできます。*sortname* には以下の指定が可能です。

REQUEST_ID	REQUEST ID でソートします
REQUEST_NAME	REQUEST NAME でソートします
USER_NAME	USER NAME でソートします
GROUP_NAME	GROUP NAME でソートします
ACCOUNT_CODE	ACCOUNT CODE でソートします
QUEUE_NAME	QUEUE NAME でソートします
QUEUED_TIME	QUEUED TIME でソートします
START_TIME	START TIME でソートします
END_TIME	END TIME でソートします
CPU	CPU (SECS) でソートします
CPU_SYS	CPU (SECS) SYS でソートします
CPU_USER	CPU (SECS) USER でソートします
REAL	REAL (SECS) でソートします
REQUEST_PRTY	REQUEST PRTY でソートします
NICE	NICE でソートします
REQELAPS_TIME	REQELAPS TIME(S) でソートします
REQCPU_TIME	REQCPU TIME(S) でソートします
REQCPU_NUM	REQCPU NUM でソートします
REQMEM_SIZE	REQMEM SIZE(K) でソートします
REQGPU_NUM	REQGPU NUM でソートします
EXIT_STAT	STATUS もしくは EXIT STAT でソートします
KCORE_MIN	KCORE MIN でソートします
MEAN_SIZE	MEAN SIZE(K) でソートします
MAXMEM_SIZE	MAXMEM SIZE(K) でソートします
V_KCORE_MIN	V KCORE MIN でソートします
V_MEAN_SIZE	V MEAN SIZE(K) でソートします
V_MAXMEM_SIZE	V MAXMEM SIZE(K) でソートします

QUE_TYPE	QUE TYPE でソートします
EHOST_NUM	EHOST NUM でソートします
NUM_JOBS	JOBS でソートします
NUM_SUBREQ	SUBREQ でソートします
RERUN_COUNT	RERUN COUNT でソートします
PRERUN_COUNT	PRERUN COUNT でソートします
TEMPLATE_NAME	TEMPLATE NAME でソートします(*2)
CR_crname	カスタムリソース crname でソートします
REQVE_NUM	REQVE NUM でソートします (*1)
RSVVE_NUM	RSVVE NUM でソートします (*1)
VE_CPU	VE CPU(S)でソートします (*1)（*3）
VE_KCORE_MIN	VE KCORE MIN でソートします (*1)（*3）
VE_MEAN_SIZE	VE MEAN SIZE(K)でソートします (*1)（*3）
VE_MAXMEM_SIZE	VE MAXMEM SIZE(K)でソートします (*1)
REQCPUTIM_SUM	REQCPUTIME SUM(S)でソートします
REQCPUNUM_SUM	REQCPUNUM SUM でソートします
REQMEMSIZE_SUM	REQMEMSIZE SUM(M)でソートします
REQGPUNUM_SUM	REQGPUNUM SUM でソートします
HW_FAILURE	HW FAILURE でソートします
VE_CPU_PS	VE CPU(SECS)でソートします (*1)(*4)
VE_KCORE_MIN_PS	VE KCORE MIN でソートします (*1)(*4)
VE_MEAN_SIZE_PS	VE MEAN SIZE(K)でソートします (*1)(*4)
VE_TIME_SLICE	VE TIME SLICE でソートします (*1)
VE_MFLOPS	VE MFLOPS でソートします (*1)
VE_MFLOPS_REAL	VE MFLOPS(REAL)でソートします (*1)
VE_FPEC	VE FPEC(K)でソートします (*1)
VE_SYSTEM_CALL	VE SYSTEM CALL でソートします (*1)
VE_VH_DATA_TRANS	VE-VH DATA TRANS でソートします (*1)
VE_PROC_NUM	NUMBER OF VE PROCESS でソートします (*1)
VE_INSTRCT	VE INSTRCT(K)でソートします (*1)
VE_INST	VE-INST(K)でソートします (*1)
VEC_ELMT	VEC-ELMT(K)でソートします (*1)
VEC_EXE	VEC-EXE(SECS)でソートします (*1)
VE_INST_LOAD	VEC-INST LOAD(K)でソートします (*1)
VE_L1_CACHE_MISS_TIME	VE L1 CACHE MISS TIME(SECS) でソートします (*1)
VE_CPU_PORT_CONFLICT	VE CPU PORT CONFLICT(SECS) でソートします (*1)
VEC_ARITH_EXEC_TIME	VEC ARITH EXEC TIME(SECS) でソートします (*1)

VEC_LOAD_EXEC_TIME	VEC LOAD EXEC TIME(SECS) でソートします (*1)
VE_PORT_THROTTLING	VE POWER THROTTLING(SECS) でソートします (*1)
VE_THERMAL_THROTTLING	VE THERMAL THROTTLING(SECS) でソートします (*1)
VE_MAX_NTASK	VE MAX NTASK でソートします (*1)
VEC_OP_RATIO	VEC-OPERAND RATIO でソートします (*1)
GPU_NUM_PROCS	GPU プロセス数でソートします (*5)
GPU_ENERGY_CONSUMED	GPU 消費電力量でソートします (*5)
GPU_MEAN_POWER_USAGE	GPU 平均消費電力でソートします (*5)
GPU_MAX_POWER_USAGE	GPU 最大消費電力でソートします (*5)
GPU_MAX_MEM_USED	最大使用 GPU メモリ量 でソートします (*5)
GPU_MEAN_SM_CLOCK	GPU 平均周波数でソートします (*5)
GPU_MAX_SM_CLOCK	GPU 最大周波数でソートします (*5)
GPU_MEAN_SM_UTIL	GPU 平均使用率でソートします (*5)
GPU_MAX_SM_UTIL	GPU 最大使用率でソートします (*5)
GPU_MEAN_MEM_UTIL	GPU 平均メモリ使用率 でソートします (*5)
GPU_MAX_MEM_UTIL	GPU 最大メモリ使用率 でソートします (*5)

*1　これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

*2　これらの項目は、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

*3 これらの項目は、-V オプションで表示する VE CPU(SECS)、VE KCORE MIN、VE MEAN SIZE(K)の値でソートします。

*4 これらの項目は、--veps オプションで表示する VE CPU(SECS)、VE KCORE MIN、VE MEAN SIZE(K)の値でソートします。

*5 これらの項目は、実行ホストにおいて GPU アカウント出力が有効な場合のみ有効です。

-d *database*

出力の対象となるデータベースがあるディレクトリ名を指定します。
NQSV の古いバージョンのデータベースも指定可能です。
本オプションが指定されない場合、デフォルトのデータベース
(/var/opt/nec/nqsv/asv/配下の racct、racct_tally、racct_cr および racct_cr_tally)

が出力の対象となります。

関連項目

scacctjob(1-N), scacctrsv(1-N), racctjob(1-N), racctrsv(1-N).

1.21 scacctrsv/racctrsv(1)

名前

scacctrsv,racctrsv -- 予約アカウンティング情報の出力

形式

/opt/nec/nqsv/bin/scacctrsv [options]
/opt/nec/nqsv/bin/racctrsv acct-server [options]

機能説明

scacctrsv コマンドは、予約アカウンティング情報を出力します。

racctrsv コマンドは、アカウンティングサーバにアクセスして、
予約アカウンティング情報を出力します。第一引数にアカウンティング
サーバのホスト名を指定し、第二引数以降に scacctrsv コマンドと
同様なオプションを指定します。

既定値では下記の項目が出力されます。

RESERVATION ID	リソース予約区間 ID
RESERVATION TIME	
START	予定開始時刻
END	予定終了時刻
EHOST NUM	ノード数
CPU NUM	1 ホスト当たりの CPU 数
QUEUE	投入キュー名
STATUS	終了ステータス

オプションの指定によりさらに、下記の項目が出力されます。

BATCH_SERVER	バッチサーバホスト名
USER NAME	ユーザ名
GROUP NAME	グループ名
ACCOUNT CODE	アカウントコード
CREATE TIME	予約生成時刻
ACTUAL RESERVED TIME	
START	実開始時刻
END	実終了時刻
TEMPLATE NAME	テンプレート名 (*)

(*) 本項目は、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

また、下記のオプションが指定可能です。

-N　　scacctrsv のバージョン情報を出力します。
他のオプションを指定した場合でも、本オプションが優先され他の
オプションはすべて無視されます。

-P *priv*
コマンド実行時のアクセス権を指定します。priv には、次を指定します。

m 管理者権限
o　　操作員権限
g　　グループ管理者権限
s　　特別利用者権限
u　　一般利用者権限

本オプションが指定されない場合、コマンド実行時のアクセス権は、
環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。
環境変数も設定されていない場合は、一般ユーザ権限となります。
環境変数には、以下の指定が可能です。ただし、racctrsv は環境変数は
使用できません。

PRIV_MGR　　管理者権限
PRIV_OPE　　操作員権限
PRIV_GMGR　グループ管理者権限
PRIV_SPU　　特別利用者権限
PRIV_USR　　一般利用者権限

-U　　ユーザ名を表示します。

-Y　　グループ名を表示します。

-X　　アカウントコードを表示します。

-B　　バッチサーバホスト名を表示します。

-C　　予約生成時刻を表示します。

-F　　実開始時刻および実終了時刻を表示します。

-L　　テンプレート名を表示します。
本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

-R　　１リソース予約区間の情報を複数行で出力します。

-v　　見出し行を表示しません。

-h *bsv* 指定されたバッチサーバの予約アカウンティング情報を出力します。

-u *user*
指定されたユーザの予約アカウンティング情報を出力します。

-g *group*
指定されたグループで実行された予約アカウンティング情報を出力します。

-x *acctcode*
指定されたアカウントコードで実行された予約アカウンティング情報を出力します。

-I *RES-ID*
指定されたリソース予約区間 ID を持つ予約アカウンティング情報を出力します。

-s *time*
指定された時刻あるいはそれ以降に生成された予約アカウンティング情報を出力します。時刻の指定形式は以下の通りです。

hour[:min[:sec[, MM/DD[/YY]]]]
(MM： 月, DD： 日, YY： 西暦の下二桁)

-e *time*
指定された時刻あるいはそれ以前に生成された予約アカウンティング情報を出力します。

-S *time*
指定された時刻あるいはそれ以降に開始予定の予約アカウンティング情報を出力します。
-f オプションを指定した場合は実開始時刻で識別します。

-E *time*

指定された時刻あるいはそれ以前に終了予定の予約アカウンティング情報を出力します。
-f オプションを指定した場合は実終了時刻で識別します。

-A *time*

指定された時刻あるいはそれ以降に終了予定の予約アカウンティング情報を出力します。
-f オプションを指定した場合は実終了時刻で識別します。

-b *time*

指定された時刻あるいはそれ以前に始予定の予約アカウンティング情報を出力します。
-f オプションを指定した場合は実開始時刻で識別します。

-Q *queue*

指定されたキューの予約アカウンティング情報を出力します。

-l *templatename*

指定されたテンプレート名を持つ予約アカウンティング情報を出力します。
本オプションは、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

-o *sortname*

管理者権限もしくはオペレータ権限で実行した場合は指定された項目でソートします。
その他の権限で実行した場合はグループ名毎に指定された項目でソートします。
sortname には以下の指定が可能です。

RES_ID	RESERVATION ID でソートします
BSV	BATCH_SERVER HOST でソートします
USER_NAME	USER NAME でソートします
GROUP_NAME	GROUP NAME でソートします
ACCOUNT_CODE	ACCOUNT CODE でソートします
CREATE_TIME	CREATE TIME でソートします
RES_START_TIME	RESERVATION TIME の START TIME でソートします
RES_END_TIME	RESERVATION TIME の END TIME でソートします
START_TIME	ACTUAL RESERVED TIME の START TIME でソートします
END_TIME	ACTUAL RESERVED TIME の END TIME でソート

します

NODE_NUM	NODE NUM でソートします
CPU_NUM	CPU NUM でソートします
QUEUE	QUEUE でソートします
STATUS	STATUS でソートします
TEMPLATE_NAME	TEMPLATE NAME でソートします (*)

(*) 本項目は、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。

-d *database*

出力の対象となる予約アカウントデータベースのファイル名を指定します。本オプションが指定されない場合、デフォルトのデータベース(/var/opt/nec/nqsv/asv/rsvacct)が出力の対象となります。

関連項目

scacctjob(1-N), scacctreq(1-N), racctjob(1-N), racctreq(1-N).

1.22 sstat(1)

本コマンドは、システム管理者向けに NQSV/JobManipulator で行ったスケジューリング関連設定情報を参照するコマンドです。リクエストの詳細情報を確認する場合は、qstat(1)を使用してください。

名前

sstat -- JobManipulator の状態表示

形式

sstat [**-R**] [**-P** *privilege*] [**-V**] [**-f**] [**-h** *server_host*] [-s *scheduler_id*] [**-u** *userlist*] [**-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**

機能説明

1. **-R** オプションを指定、もしくは表示する情報を示すオプション(A,B,C,D,E,G,J,M,Q,S)を指定しない場合

バッチリクエストのスケジューリング情報を表示します。
batch_request_identifier に表示対象となるリクエスト ID を指定します。形式は以下のとおりです。

[*request_number*][.*batch_server*]

request_number
　リクエスト ID のシーケンス番号、パラメトリックリクエストのサブリクエストの場合は、その後に[n]の形式で指定します。n にサブリクエスト番号を指定します。
batch_server
　バッチサーバ名

request_number が省略された場合は、コマンド実行者が所有する全てのバッチリクエストとなります。
batch_server が省略された場合は、JobManipulator サーバが使用している NQSV バッチサーバ名となります。

batch_request_identifier が省略された場合は、コマンド実行者が所有する全てのバッチリクエストが対象となります。
以下のオプションが指定できます。

-f　　詳細情報を表示します。

-n　　ヘッダを出力しません。

-p *parametric_request_number*
　　parametric_request_number には、パラメトリックリクエストのシーケンス番号を指定します。
　　指定されたパラメトリックリクエストの JobManipulator 管理対象のサブリクエストの情報を表示します。

-q *queuelist*
　　queuelist で指定された実行キューに属するバッチリクエストが対象となります。
　　queuelist には、実行キュー名を","で区切って複数指定できます。

-u *userlist*
　　userlist で指定されたユーザが所有しているバッチリクエストが対象となります。
　　userlist にはログイン名を","で区切って複数指定できます。

--group[=*group_name*]

バッチリクエストのグループ名を表示します。
group_name にグループ名を指定した場合は、指定グループのバッチリクエストのみを表示します。

2.　**-A** オプションを指定した場合

バッチリクエストのアサインマップ情報を表示します。
以下のオプションが指定できます。

-b *begin_time*

表示するアサインマップ情報の開始時間を *begin_time* に指定します。
begin_time の形式は次のとおりです。

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

各々の 2 桁は次のように表されます。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時間がすでに過ぎている場合は、次の日となります。
本オプションが省略された場合は、現在時刻となります。
開始時間は、分単位に切り捨てられます。

-e *end_time*

表示するアサインマップ情報の終了時間を *end_time* に指定します。 *end_time* の形式は *begin_time* と同じです。
本オプションが省略された場合は、マップにアサインされているリクエストの終了予定時間の一番遅い時間となります。

-i interval

アサインマップ情報の表示間隔を *interval* に指定します。単位は分です。
本オプションが省略された場合は、1 分単位にアサインマップ情報が表示されます。

-n　　ヘッダを出力しません。

3. **-B** オプションを指定した場合

予約区間の情報を表示します。
以下のオプションが指定できます。

-f　　詳細情報を表示します。

-n　　ヘッダを出力しません。

--group[=*group_name*]
グループ指定で作成した予約区間の情報を表示します。
group_name を指定した場合は、指定グループの予約区間の情報のみを表示します。

--template
テンプレート指定の予約区間の情報のみ表示します。

4. **-C** オプションを指定した場合

JobManipulator が管理しているコンプレックスキューの情報を表示します。
complex_name に表示対象コンプレックスキューの名前を指定します。指定したコンプレックスキューの情報のみを表示します。
省略した場合は、全てのコンプレックスキューの情報を表示します。
以下のオプションが指定できます。

-n　　ヘッダを出力しません。

5. **-D** オプションを指定した場合

全ての省電力計画の情報を表示します。
以下のオプションが指定できます。

-f　　詳細情報を表示します。

-n　　ヘッダを出力しません。

6. **-E** オプションを指定した場合

JobManipulator が使用している実行ホストの情報を表示します。
execution_host に表示対象の実行ホストのホスト名を指定します。
指定した実行ホストのみの情報を表示します。
省略した場合は、下記の-a オプションにより表示対象は異なります。
詳細については、下記オプションをして参照してください。
以下のオプションが指定できます。

-a　　ATTACH された実行ホストの情報を表示します。
-a が省略された場合は、キューにバインドしている実行ホストの情報のみを表示します。

-g *node_group*
node_group にはノードグループ名を指定します。
指定したノードグループに含まれている実行ホストの情報を表示します。

-n　　ヘッダを出力しません。

--eco-status
実行ホストの省電力状態を表示します。-f が指定された場合は、省電力状態の詳細情報を表示します。

--hw-failure
実行ホストの HW 障害情報と HW 障害によりスケジューリングから除外したノードを表示します。

7. **-J** オプションを指定した場合
JobManipulator のジョブサーバ情報を表示します。
jobserver_identifier は対象となるジョブサーバ番号を指定します。
jobserver_identifier が省略され場合は、0～10239 までの全てのジョブサーバ情報を表示します。
以下のオプションが指定できます。

-n　　ヘッダを出力しません。

8. **-M** オプションを指定した場合

スケジューリングプライオリティの情報を表示します。

9.　**-Q** オプションを指定した場合

JobManipulator が管理しているキューの情報を表示します。
以下のオプションが指定できます。
一般利用者の場合、アクセス権のあるキューのみの表示となります。

-e　　実行キューのみの情報を表示します。

-f　　詳細情報を表示します。
-j を併せて指定した場合は、ジョブサーバアサインプライオリティ情報を表示します。**-j** を併せて指定した場合に**-a** も併せて指定した場合は、ATTACH された実行ホストのジョブサーバアサインプライオリティ情報を表示します。-a を指定しない場合は、バインドしているジョブサーバのみのジョブサーバアサインプライオリティ情報を表示します。
--limit を併せて指定した場合は、キュー単位でのユーザ毎とグループ毎の同時実行リクエスト数制限値、同時実行 CPU 台数制限値および、同時実行 VE ノード数制限値を表示します。

-i　　会話キューのみの情報を表示します。

-n　　ヘッダを出力しません。

queue　表示対象のキューのキュー名を指定します。

10. **-S** オプションを指定した場合

JobManipulator サーバの情報を表示します。
以下のオプションが指定できます。

-f　　詳細情報を表示します。
--limit を併せて指定した場合は、スケジューラ単位でのユーザとグループ個別の同時実行リクエスト数制限値、同時実行 CPU 台数制限値および、同時実行 VE ノード数制限値を表示します。

-n　　ヘッダを出力しません。

11. **--help** オプションを指定した場合

sstat の使い方を表示して終了します。

12. **--version** オプションを指定した場合

コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

共通で以下のオプションが指定できます。

-d

時間データの表示フォーマットを d+hh:mm:ss に変更します。

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

表示する情報を示すオプション(R,Q,S,E,B,C,オプションなし)に対応したアイテムにしたがって、出力フォーマットをカスタマイズします。

[アイテムリスト]

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*

JobManipulator が実行されるサーバホストを指定します。

本オプションが指定されない場合、デフォルトの JobManipulator サーバホスト(/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定したホスト名)を使用します。

デフォルトの JobManipulator サーバホストがない場合は、ローカルホストとなります。

-l

出力される内容を規定の長さで切ることなく全て表示します。

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

item で指定された項目をキーとして昇順にソートします。

本オプションは複数回指定することで、指定された順番にソート結果をさらにソートします。

-O オプション（降順ソート）と組み合わせて指定することも可能です。

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

item で指定された項目をキーとして降順にソートします。
本オプションは、複数回指定することで、指定された順番にソート結果をさらにソートします。
-o オプション（昇順ソート）と組み合わせて指定することも可能です。

-P privilege
コマンド実行時のアクセス権を指定します。*privilege* には、次の指定ができます。

m	管理者権限となります。
o	操作員権限となります。
g	グループ管理者権限となります。
s	特別利用者権限となります。
u	一般利用者権限となります。

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。
特別利用者権限では、一般利用者権限に加え他人のリクエスト情報等を参照可能です。グループ管理者権限では、管理グループ内のユーザのリクエスト情報等を参照可能です。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 **JM_PRIV** で指定されたアクセス権となります。
環境変数が設定されていない場合は、一般利用者権限となります。

-s *scheduler_id*
JobManipulator のスケジューラ ID を指定します。
本オプションが指定されない場合、スケジューラ ID は、環境変数 **JM_SCHED_ID** で指定されたスケジューラ ID となります。
環境変数が設定されていない場合は、/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定した JobManipulator スケジューラ ID を使用します。
いずれの設定もされていない場合は 0 となります。

-V
コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

標準出力

各リストで表示される内容は次のとおりです。

バッチリクエスト スケジューリング情報

- バッチリクエスト識別子（**RequestID**）

- バッチリクエスト名（**ReqName**）
- 所有者（**UserName**）
- グループ名（**GrpName**）(**--group** オプション指定)
- バッチリクエストが存在するバッチキュー名（**Queue**）
- スケジューリングプライオリティ（**Pri**）

  アサイン時 / エスカレーション時
- スケジューリング状態（**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

- 実行開始予定時間（**PlannedStartTime**）

バッチリクエスト スケジューリング情報（-f オプション指定）

- バッチリクエスト識別子（**Request ID**）
- バッチリクエスト名（**Request Name**）
- 所有者（**User Name**）
- 実行ユーザ ID (**User ID**)
- グループ名（**Group Name**）
- グループ ID (**Group ID**)
- 現在のスケジューリング状態（**Current State**）
- 一つ前のスケジューリングの状態（**Previous State**）
- 現在の状態に遷移した時間（**State Transition Time**）
- 現在の状態に遷移した理由（**State Transition Reason**）
- エラー発生により状態が遷移した場合、そのメッセージ（**Error Message**）

- サスペンド理由（Suspend Reason）

INTERRUPT	緊急/特別リクエストの割り込みによりサスペンドされたリクエスト
SIGSTOP	qsigコマンドによりサスペンドされたリクエスト
SMGR_SUSPEND	smgrコマンドによりサスペンドされたリクエスト
SMGR_RESUME	smgrコマンドによりリジューム要求されたサスペンドリクエスト
(none)	サスペンド状態以外のリクエスト

- バッチリクエストが存在するバッチキュー名（Queue）
- 予約区間 ID（Reservation ID）
- アサイン時のスケジューリングプライオリティ（Scheduling Priority (Assign)）

要素毎のスケジューリングプライオリティ値

ユーザシェア	(User Share)
ユーザランク	(User Rank)
リクエストプライオリティ	(Request Priority)
経過時間	(Elapse Time)
CPU 台数	(CPU Number)
VE 台数	(VE Number)
メモリサイズ	(Memory Size)
ジョブ数	(Job Number)
キューに投入してからの実行待ち時間	(Run Wait Time)
アサイン可能になってからの実行待ち時間	(Run Wait Time(ASS))
再開待ち時間	(Restart Wait Time)
緊急リクエストによって SUSPEND されたリクエストに対するベースアップ	(Baseup Interrupted)
再スケジューリングリクエストベースアップ	(Baseup Reschedule)
ユーザ定義ベースアップ	(Baseup User Definition)
使用実績のリクエストプライオリティ	(Pastusage Request Priority)
使用実績の経過時間	(Pastusage Elapse Time)
使用実績の CPU 台数	(Pastusage CPU Number)
使用実績の VE 台数	(Pastusage VE Number)
使用実績のメモリサイズ	(Pastusage Memory Size)

- エスカレーション時のスケジューリングプライオリティ（Scheduling Priority (Escalation)）

  要素毎のスケジューリングプライオリティ値

ユーザシェア	(User Share)
ユーザランク	(User Rank)
リクエストプライオリティ	(Request Priority)
経過時間	(Elapse Time)
CPU 台数	(CPU Number)
VE 台数	(VE Number)
メモリサイズ	(Memory Size)
ジョブ数	(Job Number)
キューに投入してからの実行待ち時間	(Run Wait Time)
アサイン可能になってからの実行待ち時間	(Run Wait Time(ASS))
再開待ち時間	(Restart Wait Time)
緊急リクエストによって SUSPEND されたリクエストに対するベースアップ	(Baseup Interrupted)
再スケジューリングリクエストベースアップ	(Baseup Reschedule)
ユーザ定義ベースアップ	(Baseup User Definition)
使用実績のリクエストプライオリティ	(Pastusage Request Priority)
使用実績の経過時間	(Pastusage Elapse Time)
使用実績の CPU 台数	(Pastusage CPU Number)
使用実績の VE 台数	(Pastusage VE Number)
使用実績のメモリサイズ	(Pastusage Memory Size)

- 実行開始予定時間（Planned Start Time）
- 実行終了予定時間（Planned End Time）
- デッドライン時刻（Deadline Time）
- デッドラインスケジューリング（Deadline Scheduling）

  ON　デッドラインリクエスト

  OFF　非デッドラインリクエスト

- Elapse マージン（Elapse Margin）
- クラウドバースティングプライオリティ（Cloud Bursting Priority）
- ジョブの所属するジョブサーバ番号（Job Server a Job belongs to (Job No.:JSV No.)）

- 資源使用量宣言値（**Resources Limits**）
  - 経過時間制限値（**Elapse Time**）
  - ジョブ毎の CPU 台数制限値（**CPU Number**）
  - ジョブ毎のメモリサイズ制限値（**Memory Size**）
- スケジューラメッセージ(**Scheduler Message**)
- エスカレーション制限理由(**Escalation Limit Reason**)

'Min Forward Time' is over	横方向エスカレーション開始時間差制限に該当
'Max Side Escalation Jobs' and 'No Escalation Period' is over.	横方向エスカレーションジョブ数制限および横方向エスカレーション開始時間制限値に該当
'Min Forward Time', 'Max Side Escalation Jobs' and 'No Escalation Period' is over.	上記の両方の条件に該当

JobManipulator サーバ情報　（-S オプション指定）

- JobManipulator サーバホスト（**JMServerHost**）
- スケジューラ単位での同時実行リクエスト数制限（**GRL**）

  ULIM　無制限
- ユーザ一律の同時実行リクエスト数制限（**URL**）

  ULIM　無制限
- ユーザ一律の同時アサインリクエスト数制限（**UAL**）

  ULIM　無制限
- JobManipulator が管理しているバッチリクエスト数（**TOT**）
- 各スケジューリング状態毎のバッチリクエスト数

ASSIGNED	(**ASG**)
EXCLUDE	(**EXC**)
EXITING	(**EXT**)
HELD	(**HLD**)
QUEUED	(**QUE**)
RUNNING	(**RUN**)
SUSPENDED	(**SUD**)

- JobManipulator サーバの状態(**Status**)

Active	スケジューリング中
No-Scheduling	スケジューリング停止

JobManipulator サーバ情報（-S,-f オプション指定）

- JobManipulator サーバホスト（**JobManipulator Server Host**）
- JobManipulator サーババージョン（**JobManipulator Version**）
- JobManipulator サーバの状態（**JobManipulator Status**）

  Active　スケジューリング中

  No-Scheduling　スケジューリング停止
- スケジューラ ID（**Scheduler ID**）
- スケジュール間隔（**Schedule Interval**）
- スケジュール可能時間（**Schedule Time**）
- Elapse 無制限スケジューリング（**Use Elapse Unlimited**）

  ON　無制限

  OFF　制限有り
- リクエストアサインポリシー（**Request Assign Policy**）

  Resource balance　リソース分散アサイン

  CPU concentration　CPU 台数リソース集中アサイン
- アサインポリシーの優先度（**Assign Policy Priority**）

  Network Topology　ネットワークトポロジーを考慮したアサイン

  Staging Job　予定実行開始時刻がクリアされたステージング中ジョブがあるノード以外のノードの優先アサイン
- スケジューラ単位での同時実行リクエスト数制限（**Global Run Limit**）

  UNLIMITED　無制限
- スケジューラ単位でのグループ一律の同時実行リクエスト数制限（**Group Run Limit**）

  UNLIMITED　無制限
- スケジューラ単位でのグループ個別の同時実行リクエスト数制限（**Group Run Limit of Each Group**）（**--limit オプション指定**）

  UNLIMITED　無制限
- スケジューラ単位でのグループ一律の同時実行 CPU 台数制限(**Group CPU Run Limit**)

  UNLIMITED　無制限
- スケジューラ単位でのグループ個別の同時実行 CPU 台数制限（**Group CPU Run Limit of Each Group**）（**--limit オプション指定**）

  UNLIMITED　無制限
- スケジューラ単位でのグループ一律の同時実行 VE ノード数制限(**Group VE Run Limit**)

  UNLIMITED　無制限

- スケジューラ単位でのグループ個別の同時実行 VE ノード数制限（**Group VE Run Limit of Each Group**）（**--limit オプション指定**）

  UNLIMITED　無制限

- スケジューラ単位でのユーザ一律の同時実行リクエスト数制限（**User Run Limit**）

  UNLIMITED　無制限

- スケジューラ単位でのユーザ個別の同時実行リクエスト数制限（**User Run Limit of Each User**）（**--limit オプション指定**）

  UNLIMITED　無制限

- スケジューラ単位でのユーザ一律の同時実行 CPU 台数制限(**User CPU Run Limit**)

  UNLIMITED　無制限

- スケジューラ単位でのユーザ個別の同時実行 CPU 台数制限（**User CPU Run Limit of Each User**）（**--limit オプション指定**）

  UNLIMITED　無制限

- スケジューラ単位でのユーザ一律の同時実行 VE ノード数制限(**User VE Run Limit**)

  UNLIMITED　無制限

- スケジューラ単位でのユーザ個別の同時実行 VE ノード数制限（**User VE Run Limit of Each User**）（**--limit オプション指定**）

  UNLIMITED　無制限

- ユーザ一律の同時アサインリクエスト数制限（**User Assign Limit**）

  UNLIMITED　無制限

- 半減期（**Half-Reduce Period**）
- 標準 CPU 台数（**Standard CPU Number**）
- 標準 VE 台数（**Standard VE Number**）
- 標準メモリサイズ（**Standard Memory Size**）
- 標準ジョブ数（**Standard Job Number**）
- エスカレーション（**Escalation**）

  Forward_Escalation　ノード変更しないで前方にエスカレーション

  All_Escalation　ノード変更する／しないに関係なくエスカレーション

  OFF　エスカレーション不可

- エスカレーション実行間隔（**Escalation Interval**）
- 指定した時間にアサインできないリクエスト場合の対応（**Treat unbookable request**）

  delete　削除

  after　指定された時間より後ろの最も近い時刻にアサイン

- 割り込まれたリクエスト処理（**Interruption method**）

  suspend　サスペンド

  rerun　リラン

- 優先リクエストの割込み位置（**Interrupt to where**）

  next_run　アサイン済実行待ちリクエストの先頭

  current　現在時刻

- 緊急リクエストによるアサイン禁止（**Interrupt assign block**）

  ON　実行待ちの緊急リクエストがアサインされている実行ホストへの新規投入された下位タイプリクエストのアサインを禁止する

  OFF　実行待ちの緊急リクエストがアサインされている実行ホストへの新規投入された下位タイプリクエストのアサインを許可する

- 予約区間自動削除（**Auto Delete Resource Reservation**）

  ON　自動削除機能を使用する

  OFF　自動削除機能を使用しない

- 実行中ジョブの強制リラン（**Forced Re-Scheduling**）

  ON　ストールした、バースティングされていない実行中ジョブをリランする

  OFF　ストールした、バースティングされていない実行中ジョブをリランしない

- バースティングされた実行中ジョブの強制リラン（**Forced Re-Scheduling(Cloud)**）

  ON　ストールした、バースティングされた実行中ジョブをリランする

  OFF　ストールした、バースティングされた実行中ジョブをリランしない

- 動的省電力運用機能（**Dynamic DC Control**）

  ON　動的電源制御を開始する

  OFF　動的電源制御を停止する

- 最大稼働ノード数（**Max Operation Hosts**）
- ピークカット緊急度（**Peak Cut Urgency**）

  right_now　即時にノードを停止する

  wait_run　実行中ジョブの終了を待ち合わせてノードを停止する

- 最低アイドル時間（**Min Idle Time**）
- 省電力停止見込み時間（**Estimated DC-OFF Time**）
- 省電力停止回数制限（**DC-OFF Limit**）
- 追い越し不可制御（**Use Overtake Priority**）
  - 実行キュー種別 = normal タイプ（**normal**）

    ON　追い越し不可制御を有効

    OFF　追い越し不可制御を無効

  - 実行キュー種別 = special タイプ（**special**）

    ON　追い越し不可制御を有効

    OFF　追い越し不可制御を無効

- 追い越し不可プライオリティ（**Overtake Priority**）

- 実行キュー種別 = normal タイプ（**normal**）
- 実行キュー種別 = special タイプ（**special**）

- 追い越し関連設定（**Overtake**）
  - 小規模リクエストの追い越し割り当て機能（**Allow Small Request**）

    ON　小規模リクエストの追い越し割り当て機能を有効

    OFF　小規模リクエストの追い越し割り当て機能を無効
- 障害発生後のマップ維持時間（**Keep Forward Schedule**）
- ステージインマージン(**Stage-in Margin**)
  - ステージイン時間の見積もり調整時間（**Additional Margin for Escalation**）
  - ステージイン開始時間閾値(**Stage-in Threshold**)
  - 初回ステージイン時間（**First Stage-in Time**）
- プロビジョニング環境により起動された、仮想マシン（VM）、ベアメタルサーバ、およびコンテナの起動失敗実行ホストのリトライ待ち時間(**Provisioning Start Retry Time**)
- スケジューリング方式(**Scheduling Method**)
  - VE ノード縮退時のスケジューリング方式の設定(**VE Degradation**)

    Continue 最新の VE ノード数に追随してスケジューリングする

    Exclude　VE ノードが縮退した VI をスケジューリング対象から除外する

    Auto　1 台以上の VE ノードが縮退すると当該 VI をスケジューリング対象から除外する。縮退したすべての VE ノードが復旧すると当該 VI が直ちにスケジューリング対象に戻る
  - リクエストのアサインモードの設定(**Assign Mode**)

    Continue スケジューリング処理を一定時間で打ち切らず、継続する。

    Reset　スケジューリング処理を一定時間で打ち切る
  - スケジューリング不可リクエストのキャッシュ(Non Scheduled Request Cache)

    ON　スケジューリング不可リクエストのキャッシュが有効

    OFF　スケジューリング不可リクエストのキャッシュが無効
- リクエスト即時スケジューリング(**Realtime Scheduling**)

  ON　リクエスト即時スケジューリングを有効

  OFF　リクエスト即時スケジューリングを無効
- リクエスト即時スケジューリングモード（**Realtime Scheduling Mode**）

  Default　リクエスト即時スケジューリングはイベント処理用時間帯では行わない

  Always　リクエスト即時スケジューリングを常に行う
- HCA デバイスグループトポロジスケジューリング(**Device Group Topology**)

  ON　HCA デバイスグループトポロジスケジューリングを有効

  OFF　HCA デバイスグループトポロジスケジューリングを無効
- 一度にクラウド資源にバースティングするリクエスト数の上限値(**Cloud Bursting Request Number**)

- クラウドバースティングプライオリティの重み付け(**Cloud Bursting Priority Weight**)

Resource	要求リソース量に対する重み付け
Wait Time	実行待ち時間に対する重み付け
Scheduling Priority	スケジューリングプライオリティに対する重み付け
Custom Resource	カスタムリソースに対する重み付け

- JobManipulator が管理しているバッチリクエスト数情報（**Request Statistical Information**）
  - JobManipulator が管理しているバッチリクエスト数（**Total Request**）
  - 各スケジューリング状態毎のバッチリクエスト数

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**)

- ログファイル情報（**Logfile Information**）
  - ログファイルパス名（**Logfile Path**）
  - ログファイルレベル（**Logfile level**）
  - 最大ログファイルサイズ（**Logfile MAX Size**）
  - 保存ログファイル数（**Logfile Save Count**）
- JobManipulator にバインドされているジョブサーバ（**Bind JobServer**）
  - ライセンスを使用しているジョブサーバ（**Use Licenses**）
  - ライセンスを使用していないジョブサーバ（**Unuse Licenses**）

スケジューリングプライオリティ情報　(-M オプション指定)

- スケジューリングプライオリティ計算タイミング

  アサイン時　　(**Assign**)

  エスカレーション時　(**Escalation**)

- 重み付け (**Scheduling Weight**)
  - ユーザシェア (**User Share**)
  - ユーザランク (**User Rank**)
  - リクエストプライオリティ (**Request Priority**)
  - 経過時間 (**Elapse Time**)
  - CPU 台数 (**CPU Number**)
  - VE 台数 (**VE Number**)
  - メモリサイズ (**Memory Size**)
  - ジョブ数 (**Job Number**)
  - キューに投入してからの実行待ち時間 (**Run Wait Time**)
  - アサイン可能になってからの実行待ち時間 (**Run Wait Time(ASS)**)
  - 再開待ち時間 (**Restart Wait Time**)
- ベースアップ (**Baseup**)
  - 緊急リクエストによって SUSPEND されたリクエスト (**Interrupted**)
  - 再スケジューリングリクエスト (**Reschedule**)
- 使用実績 (**PastUsage**)
  - リクエストプライオリティ (**Request Priority**)
  - 経過時間 (**Elapse Time**)
  - CPU 台数 (**CPU Number**)
  - VE 台数 (**VE Number**)
  - メモリサイズ (**Memory Size**)
- ユーザランク使用実績重み付け (**PastUsage(User Rank)**)
  - リクエストプライオリティ (**Request Priority**)
  - 経過時間 (**Elapse Time**)
  - CPU 台数 (**CPU Number**)
  - VE 台数 (**VE Number**)
  - メモリサイズ (**Memory Size**)

キュー情報　(-Q オプション指定)

- 実行キュー名 (**QueueName**)
- 実行キュー種別 (**Type**)　(会話キューには表示しない項目)
- 実行キュー内での同時実行リクエスト数制限 (**RL**)

  ULIM　無制限
- キュー内でのユーザ一律の同時実行リクエスト数制限 (**URL**)
- 実行キュー内でのユーザ一律の同時アサインリクエスト数制限 (**UAL**)

ULIM　無制限

- 実行キュー毎のバッチリクエスト数（**TOT**）
- 各スケジューリング状態毎のバッチリクエスト数

ASSIGNED　（**ASG**）
EXCLUDE　（**EXC**）
EXITING　（**EXT**）
HELD　（**HLD**）（会話キューには表示しない項目）
QUEUED　（**QUE**）
RUNNING　（**RUN**）
SUSPENDED　（**SUD**）

キュー情報　（-Q,-f オプション指定）

- 実行キュー名（**Queue Name**）
- 実行キュー種別（**Queue Type**）（会話キューには表示しない項目）
- マップ幅（**Schedule Time**）
- リクエストアサインポリシー（**Request Assign Policy**）

  Resource balance　リソース分散アサイン
  CPU concentration　CPU 台数リソース集中アサイン
  DEFAULT　スケジューラ単位の設定に従う
- 優先リクエストの割込み位置（**Interrupt to where**）（会話キューには表示しない項目）

  next_run　アサイン済実行待ちリクエストの先頭
  current　現在時刻
  DEFAULT　スケジューラ単位の設定に従う
- 実行キュー内での同時実行リクエスト数制限（**Run Limit**）

  UNLIMITED　無制限
- 実行キュー内でのグループ一律の同時実行リクエスト数制限（**Group Run Limit**）

  UNLIMITED　無制限
- 実行キュー内でのグループ個別の同時実行リクエスト数制限（**Group Run Limit of Each Group**）**(--limit オプション指定)**

  UNLIMITED　無制限
- 実行キュー内でのグループ一律の同時実行 CPU 台数制限（**Group CPU Run Limit**）

  UNLIMITED　無制限
- 実行キュー内でのグループ個別の同時実行 CPU 台数制限（**Group CPU Run Limit of Each Group**）**(--limit オプション指定)**

UNLIMITED　無制限

- 実行キュー内でのグループ一律の同時実行 VE ノード数制限（**Group VE Run Limit**）

  UNLIMITED　無制限

- 実行キュー内でのグループ個別の同時実行 VE ノード数制限（**Group VE Run Limit of Each Group**）**(--limit オプション指定)**

  UNLIMITED　無制限

- 実行キュー内でのユーザ一律の同時実行リクエスト数制限（**User Run Limit**）

  UNLIMITED　無制限

- 実行キュー内でのユーザ個別の同時実行リクエスト数制限（**User Run Limit of Each User**）**(--limit オプション指定)**

  UNLIMITED　無制限

- 実行キュー内でのユーザ一律の同時実行 CPU 台数制限（**User CPU Run Limit**）

  UNLIMITED　無制限

- 実行キュー内でのユーザ個別の同時実行 CPU 台数制限（**User CPU Run Limit of Each User**）**(--limit オプション指定)**

  UNLIMITED　無制限

- 実行キュー内でのユーザ一律の同時実行 VE ノード数制限（**User VE Run Limit**）

  UNLIMITED　無制限

- 実行キュー内でのユーザ個別の同時実行 VE ノード数制限（**User VE Run Limit of Each User**）**(--limit オプション指定)**

  UNLIMITED　無制限

- 実行キュー内でのユーザ一律の同時アサインリクエスト数制限（**User Assign Limit**）

  UNLIMITED　無制限

- Elapse マージン（**Elapse Margin**）
- クラウドバースティングのスケジューラマップ占有率の閾値（**Cloud Bursting Occupancy Ratio**）
- カレントのクラウドバースティングのスケジューラマップ占有率（**Current Cloud Bursting Occupancy Ratio**）
- カレントのクラウドバースティング可否の状態（**Current Cloud Bursting Status**）

  ENABLE　現在、クラウドバースティングが可能

  DISABLE　現在、クラウドバースティングが不可

- リクエストのクラウドバースティングプライオリティの表示可否情報（**Display Cloud Bursting Priority**）

  ON　リクエストのクラウドバースティングプライオリティを表示する

  OFF　リクエストのクラウドバースティングプライオリティを表示しない

- エスカレーション対象リクエストの選択条件（**Escalation Limit**）

- ○ 横方向エスカレーションジョブ数制限値（**Max Side Escalation Jobs**）

  DISABLE　無効

- ○ 横方向エスカレーション開始時間制限値（**No Escalation Period**）

  DISABLE　無効

- ○ 横方向エスカレーション開始時間差制限値（**Min Forward Time**）

  DISABLE　無効

- 再スケジューリングの待ち時間（**Retry Time**）
  - ○ ステージイン処理失敗時の再スケジューリング待ち時間（**Staging**）
  - ○ PRE-RUNNING 処理失敗時の再スケジューリング待ち時間（**Pre-running**）
- ステージアウト待ち時間(**Wait_Stageout**)
- デッドライン制御情報（**Deadline Control**）（会話キューには表示しない項目）
  - ○ デッドラインスケジューリング（**Deadline Scheduling**）

    ON　デッドラインスケジューリング利用可能

    OFF　デッドラインスケジューリング利用不可
  - ○ 減免率下限（**Minimum Reduce Rate**）
  - ○ オンタイム減免率（**Ontime Reduce Rate**）
  - ○ 減免率上限（**Maximum Reduce Rate**）
  - ○ 減免率減少終了時間（**End Time of Rate Decrease**）
  - ○ 減免率減少開始時間（**Start Time of Rate Decrease**）
  - ○ 減免率増加終了時間（**End Time of Rate Increase**）
  - ○ 減免率増加開始時間（**Start Time of Rate Increase**）
- 最小稼働ノード数（**Min Operation Hosts**）
- 予約区間マージン(**Reservation Margin**)
  - ○ PRE-MARGIN (**Pre-margin**)
  - ○ POST-MARGIN (**Post-margin**)
- 最小ネットワークトポロジーノードグループ選択に関する情報(**Network Topology Control**)

  (最小ネットワークトポロジーグループ選択が ON の場合)
  - ○ 最小ネットワークトポロジーノードグループ選択を行うか否かの設定(**Network Topology Minimum Scheduling**)

    ON　最小ネットワークトポロジーノードグループ選択を行う
  - ○ 1 ネットワークトポロジーノードグループあたりの実行ホスト数(**Hosts per group**)

    JobManipulator によって自動計算します。値の後方に"(Default)"と表示します。
- スケジュール種別(**Schedule Type**)

  Backfill　バックフィルスケジューリングを有効

  FIFO　FIFO スケジューリングを有効
- JSV アサインプライオリティ情報（**JSV Assign Priority**)(**-j** オプション指定）（会話キューには表示しない項目）

- Dynamic JSV Priority 情報（**JSV Dynamic Assign Priority**)(**-j** オプション指定）（会話キューには表示しない項目）
- Dynamic JSV Priority の有効/無効情報（**Dynamic JSV Priority Control**）（会話キューには表示しない項目）

  ON　　有効

  OFF　　無効
- Dynamic JSV Priority 計算のための設定値（Dynamic JSV Priority Item）（会話キューには表示しない項目）
- JobManipulator が管理しているバッチリクエスト数情報（**Request Statistical Information**）
  - JobManipulator が管理しているバッチリクエスト数（**Total Request**）
  - 各スケジューリング状態毎のバッチリクエスト数

EXCLUDE	(**Exclude Request**)
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**)

実行ホスト情報　（-E オプション指定）

- 実行ホスト名（**ExecutionHost**）
- 使用可能 CPU 台数制限（**CPUNRatio**）
- 使用メモリ量制限（**MemRatio**）

実行ホスト情報　（-E　--eco-status オプション指定）

- 実行ホスト名（**ExecutionHost**）
- 省電力状態（**EcoStatus**）

EXCLUDED	省電力制御対象外
PEAKCUT	ピークカット対象ノードとして選択
—	上記のいずれにも該当しない

- 省電力状態遷移時刻(**StateTransitionTime**)
- 省電力停止回数 （当日）(**OFF(D)**)
- 省電力停止回数 （累積）(**ACCUM**)

実行ホスト情報 （-E --eco-status -f オプション指定）

- 実行ホスト名（**Execution Host**）
- 省電力状態（**Eco Status**）
- 省電力状態遷移時刻(**State Transition Time**)
- 省電力制御対象外となった理由(**Exclude Reason**)

ACTIVE_LINKDOWN	ノードが起動しているが、JSV が LINKDOWN 状態
HW_FAILURE	HW 障害
START_FAIL	起動失敗
Cloud Bursting Node	クラウドバースティング機能で起動しているノード

- 省電力停止回数(当日) (**DC-OFF Times (Day)**)
- 省電力停止回数(累積) (**DC-OFF Times(ACCUM)**)

実行ホスト情報 （-E --hw-failure オプション指定）

- 実行ホスト名（**ExecutionHost**）
- 障害遭遇状況（**Status**）

ABNORMAL STOP	ノード管理エージェントにより通知された障害
DEGRADED	VE ノードは縮退しているがスケジューリングから除外されていない状態
EXCLUDED	プロビジョニング環境において起動された仮想マシン(VM)、ベアメタルサーバ、またはコンテナの停止失敗によりスケジューリング対象から除外された状態 もしくは VE ノード縮退によりスケジューリングから除外された状態

実行ホスト情報 （-E -f オプション指定）

- 実行ホスト名（**Execution Host**）
- 実行ホストの使用可能 CPU 台数制限（**CPU Number Ratio**）

- RSG でリソースを分割した場合の各 RSG の使用可能 CPU 台数制限（**CPU Number Ratio of RSG**）
- 実行ホストの使用メモリ量制限（**Memory Size Ratio**）
- RSG でリソースを分割した場合の各 RSG の使用メモリ量制限（**Memory Size Ratio of RSG**）
- リンク状態（**LINK Batch Server**）

  UP　　リンクアップ

  DOWN　　リンクダウン
- バインド状態（**BIND Queue**）

  BIND　　バインド

  UNBIND　　アンバインド
- 実行ホストの省電力情報（**Eco Status**）
  - 省電力状態（**Status**）
  - 省電力状態遷移時刻(**State Transition Time**)
  - 省電力制御対象外となった理由(**Exclude Reason**)

    HW_FAILURE　　HW 障害

    START_FAIL　　起動失敗

    ACTIVE_LINKDOWN　　ノードが起動しているが、JSV が LINKDOWN 状態

    Cloud Bursting Node　　クラウドバースティング機能で起動しているノード

  - 省電力停止回数(当日) (**DC-OFF Times (Day)**)
  - 省電力停止回数(累積) (**DC-OFF Times(ACCUM)**)
- 実行ホストの障害遭遇情報（**Hardware Failure**）
  - 障害遭遇状況（**Status**）

    ABNORMAL STOP　　ノード管理エージェントにより通知された障害

    DEGRADED　　VE ノードは縮退しているがスケジューリングから除外されていない状態

    EXCLUDED　　プロビジョニング環境において起動された仮想マシン(VM)、ベアメタルサーバ、またはコンテナの停止失敗によりスケジューリング対象から除外された状態
    もしくは VE ノード縮退によりスケジューリングから除外された状態

  - スケジューリング対象外となった理由（Exclude Reason）

    BAREMETAL_ABNORMAL_LINKDOWN　　プロビジョニング環境において起動されたベアメタルサーバで JSV が異常 LINKDOWN した

BAREMETAL_DELAY_STAGEIN	プロビジョニング環境 においてステージインが遅延している
BAREMETAL_DELAY_STAGEOUT	プロビジョニング環境においてステージアウトが遅延している
BAREMETAL_UNKNOWN_TEMPLATE	プロビジョニング環境においてテンプレート情報が不明状態で起動している
STOPFAIL_PROVISIONING	プロビジョニング環境においてマシンの停止に失敗した

- VE ノードが縮退しているかどうかの情報（VE Degradation）

  YES　　縮退している

  (非表示)　縮退していない

- 実行ホストのスケジューリング状態（**Scheduling**）
  - スケジューリング対象かどうか（**Status**）

    Available　　スケジューリング対象

    Unavailable　　スケジューリング対象外

  - スケジューリング対象外の理由（**Reason**）

    NQS API Error　API のエラーにより資源情報の取得に失敗した

    No License　　ライセンスが無い
- プロビジョニングのリトライ待ち（**Provisioning Waiting Retry**）
  - テンプレート名 = リトライ待ち終了時間

アサインマップ情報　（-A オプション指定）

- ジョブサーバ番号（**JSV**）
- ジョブ ID（**Job-ID**）
- CPU 台数制限値（**CPU**）
- メモリサイズ制限値（**Memory**）

予約区間情報　（-B オプション指定）

- 予約区間 ID（**RES ID**）
- 予約区間開始時刻（**Start Time**）

- 予約区間終了時刻（**End Time**）
- 予約区間が使用する実行ホスト数（**NodeNum**）
- 予約区間の実行ホスト当たりの CPU 台数（**CPUNum**）
- 実行キュー（**Queue**）(実行キュー指定による予約区間設定時)
- グループ名（**GrpName**)(**--group** オプション指定)

予約区間情報　(-B --template オプション指定）(テンプレート指定の予約区間の場合)

- 予約区間 ID (**RES ID**)
- 予約区間開始時刻（**Start Time**）
- 予約区間終了時刻（**End Time**）
- 予約作成時に指定されたテンプレート名(**Template**)
- 予約作成時に指定されたマシンの台数(**MacNum**)
- 実行キュー（**Queue**）
- グループ名（**GrpName**)(**--group** オプション指定)

予約区間情報　(-B,-f オプション指定)

- 予約区間 ID (**Resource Reservation ID**)
- 予約区間名(**Resource Reservation Name**)
- グループ名(**Group Name**)
- 実行キュー（**Queue Name**）
  実行キュー指定による予約区間設定時
- 予約区間開始時刻（**Reserve Start Time**）
- 予約区間終了時刻（**Reserve End Time**）
- 予約区間が使用する実行ホスト数（**Execution Host Number**）
- 予約区間の実行ホスト当たりの CPU 台数（**Reserve CPU Number by Host**）
- 予約区間が使用する実行ホスト名と実行ホストの予約状態(**Reserved Hosts (HOST_NAME : STATUS)**)
- 予約区間を使用するリクエスト一覧（**Requests uses this reservation area**）

予約区間情報　(-B --template,-f オプション指定) (テンプレート指定の予約区間の場合)

- 予約区間 ID (**Resource Reservation ID**)
- 予約区間名(**Resource Reservation Name**)
- グループ名(**Group Name**)
- 実行キュー（**Queue Name**）
- 予約区間開始時刻（**Reserve Start Time**）
- 予約区間終了時刻（**Reserve End Time**）

- 予約作成時に指定されたテンプレート名（**Reserve Template**）
- 予約作成時に指定されたマシンの台数（**Reserve Machine Number**）
- 予約区間が使用する実行ホスト名と実行ホストの予約状態(**Reserved Machines (HOST_NAME : STATUS)**)
- 予約区間を使用するリクエスト一覧（**Requests uses this reservation area**）

コンプレックスキュー情報　（-C オプション指定）

- コンプレックスキュー名（**QueueName**）
- 所属する実行キュー（**QueueName**）
- 所属する実行キューの種別（**Type**）
- 同時実行リクエスト数制限（**RL**）

  ULIM　無制限
- ユーザ一律の同時実行リクエスト数制限（**URL**）

  ULIM　無制限
- ユーザ一律の同時アサインリクエスト数制限（**UAL**）

  ULIM　無制限
- コンプレックスキュー、所属する実行キュー毎のバッチリクエスト数（**TOT**）
- 各スケジューリング状態毎のバッチリクエスト数

  ASSIGNED　（**ASG**）
  EXCLUDE　（**EXC**）
  EXITING　（**EXT**）
  HELD　（**HLD**）
  QUEUED　（**QUE**）
  RUNNING　（**RUN**）
  SUSPENDED　（**SUD**）

計画省電力区間情報（-D オプション指定）

- 計画省電力 ID（**EcoID**）
- 計画省電力区間の開始時刻（**EcoStartTime**）
- 計画省電力区間の終了時刻（**EcoEndTime**）
- 計画省電力区間の対象ホスト名（**ExecutionHost**）

計画省電力計画情報（-D,-f オプション指定）

- 計画省電力 ID（**Eco Schedule ID**）
- 計画省電力区間の開始時刻（**Scheduled Start Time**）

- 計画省電力区間の終了時刻（**Scheduled End Time**）
- 計画省電力区間の対象ホスト数（**Number of Scheduled Hosts**）
- 計画省電力区間の対象ホスト名(**Scheduled Hosts**)

ジョブサーバ情報　（-J オプション指定）

- ジョブサーバ番号（**JSVNO**）
- キュー名(**Queue**)
- JSV アサインプライオリティ（**Priority**）
- Dynamic JSV Priority (**Dynamic Priority**)　（**--dynamic** オプション指定）

環境変数

JM_PRIV

デフォルトのコマンド実行時のアクセス権

JM_PRIV_MGR 管理者権限

JM_PRIV_OPE 操作員権限

JM_PRIV_GMGR　グループ管理者権限

JM_PRIV_SPU 特別利用者権限

JM_PRIV_USR 一般利用者権限

JM_SCHED_ID

デフォルトの JobManipulator スケジューラ ID

ファイル

/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf

1.23 wdel (1)

名前

wdel ― ワークフローの削除

形式

wdel　[-P *privilege*] [-V] *wfl-id* ...

機能説明

wdel は、*wfl-id* で指定したワークフローを削除します。*wfl-id* に空白文字で区切って複数のワークフローID を指定することができます。
このとき、指定したワークフローおよび、ワークフローとして投入、実行しているすべてのリクエストを削除します。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

wdel 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

wstart(1)

1.24 wstart (1)

名前

wstart ― ワークフローの実行

形式

wstart [-V] [--cancel-after] [*wfl-script*]

機能説明

wstart は、*wfl-script* で指定されたシェルスクリプトファイルをワークフローとして実行します。*wfl-script* の指定がない場合は、標準入力からワークフロースクリプトの入力を受け付けます。

wstart は、バッチサーバからワークフローID を取得して標準出力に表示します。

wstart は、ワークフロースクリプトを子プロセスとして実行します。その後、wstart は起動したワークフロースクリプトの実行プロセスの終了を待ち合わせます。
ワークフロースクリプト内では、qsub(1)、qwait(1)、qwait2(1)コマンドを使用して、NQSV リクエストのワークフローを記述できます。
【ワークフロースクリプトの例】

```
1.  qsub -q batch1 -N REQ-A ./job_A.sh
2.  qsub -q batch2 -N REQ-B ./job_B.sh
3.  qsub -q batch2 -N REQ-C --after REQ-A,REQ-B ./job_C.sh
4.  qwait2 REQ-C
```

ワークフロースクリプトの実行による標準入力、標準出力、標準エラー出力は、wstart を実行した端末で入出力を行います。
また、実行中のワークフローは、wdel(1)、または、wstart を実行した端末上で ctrl-C を押下することにより削除できます。ワークフロー内で投入した全リクエストの削除と wstart 自身の終了を行います。
wdel(1) によりワークフローが削除されたことにより、wstart が終了した場合は、標準エラー出力にメッセージを表示します。

wstart によって実行される NQSV リクエストのワークフローは、wstart が終了し、かつ、ワークフロースクリプト内で投入された全リクエストが終了するまでの間 NQSV システム内に存在し、wstat(1)コマンドにより参照することができます。

以下のオプションが指定できます。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

--cancel-after

ワークフロースクリプト内の全ての qsub(1)コマンドに --cancel-after オプションを自動的に指定します。

関連項目

qsub(1), qwait(1), qwait2(1), wdel(1), wstat(1)

1.25 wstat (1)

名前

wstat ― ワークフローの参照

形式

wstat　[-P *privilege*] [-V] [*wfl-id* ...]

機能説明

wstat は、ワークフローの情報を表示します。

wfl-id が省略された場合は、wstat を起動したユーザが所有するすべてのワークフローの一覧を表示します。
表示する項目は以下のとおりです。

WFL-ID	ワークフローID
Request	ワークフロー内で投入済のリクエスト数。 終了したリクエストも含みます。パラメトリックリクエストの場合は、サブリクエストの数です。
Owner	ワークフローの所有者名
Stat	ワークフローの状態

wfl-id が指定された場合は、指定したワークフロー内の投入済みのリクエストの情報を表示します。*wfl-id* に空白文字で区切って複数のワークフローID を指定することもできます。
表示する項目は以下のとおりです。

WFL-ID	ワークフローID
RequestID	リクエスト ID
ReqName	リクエスト名
Stat	リクエストの状態
Exit	リクエストの終了コード（16 進表示） リクエストが終了していない場合は、 '-' を表示します。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次を指定します。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

環境変数

NQSV_PRIV または NQSII_PRIV

wstat 実行時に使用するアクセス権の既定値

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

wstart(1)

第2章　　システム管理コマンド

システム管理コマンドは、以下の通りです。

コマンド	説明
budgetedit(1M)	予算の表示、編集
nmapmgr(1M)	NQSV ネットワークの構成管理
nqs_ntfr(1M)	実行ホストでの障害をBSVに通知
qmgr(1M)	NQSV の構成管理および運用管理
subedit(1M)	課金レートの表示、編集
sceditacct(1M)	アカウントDBの編集
scsumacct(1M)	日報、月報情報表示
scturnacct(1M)	アカウントデータの退避
smgr(1M)	JobManipulatorの構成管理および運用管理
sushare(1M)	JobManipulator ユーザシェア管理

2.1　budgetedit(1M)

名前

budgetedit - 予算の表示、編集

形式

/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

機能説明

budgetedit コマンドは予算管理項目のうち、予算の設定情報、および予算の実績の表示、登録、更新、削除などを行います。

オプションを指定しない場合には、登録されているすべての予算設定、予算実績が表示されます。

```
==================================================
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
```

オプション

オペレーション、適用範囲、ターゲット、予算設定を下記のように指定します。

budgetedit add -u usr1:192345.87/1229445.98/0.00

add　　予算設定を新たに追加します。ターゲットがすでに存在する場合にはエラー メッセージを表示します。

upd　予算設定を更新します。ターゲットが存在しない場合にはエラーメッセージを表示します。予算を増減する場合は *initial* のみを指定してください。残高は自動的に更新されます。メンテナンスなどで予算データを直接修正する場合は、*initial*/*remain*/*estimate* をすべて指定してください。

金額は以下のいずれかの方法で指定します。

- 絶対値指定
  予算額を絶対値で変更します。符号なしで指定してください。符号なしと符号ありの値を混ぜると指定エラーになります。

```
# budgetedit add -u usr1:9999/8888/7777
```

- 相対値指定
  予算金額に対して加減する金額を相対値として指定します。指定する金額には符号が必要です。マイナスとプラスの符号を混ぜて指定できません。変更しない値は 0 で指定してください。

```
# budgetedit add -u usr1:+1000/+1000/+1000
```

del　予算設定を削除します。ターゲットが存在しない場合にはエラーメッセージを表示します。

-N　budgetedit のバージョン情報を表示します。他のオプションを指定した場合でも、本オプションが優先され他のオプションはすべて無視されます。

-u　ユーザの予算設定を操作します。

-g　グループの予算設定を操作します。

-a　アカウントコードの予算設定を操作します。

name　予算設定する対象を指定します。-u の場合はユーザ名、-g の場合はグループ名、-a の場合はアカウントコードとなります。アカウントコードはアカウントコードファイル(/etc/opt/nec/nqsv/acctcode)に記述されているアカウントコード名を指定します。アカウントコード名の最大長は 15 バイトまでです。

initial

予算の初期値を設定します。

remain

予算の残量を設定します。通常、予算残高はアカウンティングサーバによって更新されます。メンテナンスなどで予算残高の修正が必要な場合にのみ指定してください。

estimate

予定課金額を設定します。通常、予定課金額はアカウンティングサーバによって更新されます。メンテナンスなどで予定課金額の修正が必要な場合にのみ指定してください。

関連項目

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

2.2　nmapmgr (1M)

名前

nmapmgr —NQSV マシン ID データベースの管理

形式

nmapmgr

機能説明

nmapmgr コマンドは、NQSV が使用するマシン ID データベースを管理します。 このデータベースは次の 4 つの基本的な要素を含みます。

alias　ネットワーク内のマシンの principal-name の別名。 alias はローカル NQS ホストにのみ理解されます。alias は 255 バイト以内の長さの文字列です。 複数の IP アドレスを持つマシンでアドレスごとに違う名前をつけたい場合に使用します。

mid　NQSV ネットワークを構成しているマシン間で一意になるように付けられたマシン ID 番号。 NQSV はこの mid を特定のマシンを識別するために使用します。この値の最小値は 1 です。この値の最大値は(2^31)-1 です。

principal-name

NQSV が管理するマシン ID データベース中でユニークなホスト名。その名前でネットワークにアクセスします。

type　サポートされている NQS のタイプ。現在以下のタイプが用意されています。

nqsv　NQSV（デフォルト）

nmapmgr コマンドを実行すると、プロンプト NMAPMGR>: を表示します。この状態で、以下に述べるようなサブコマンドを入力できます。 シェルに戻る時には exit サブコマンドを使用します。

なお、マシン ID データベースを変更する操作は、スーパーユーザしか行えません。

コマンド

すべてのコマンドは 1 行で入力してください。

Add Mid *mid principal-name*

新しいマシンを *mid* で指定されたマシン ID と *principal-name* で指定されたホスト名で NQSV ネットワーク構成へ追加します。

Add Name *alias to-mid*

to-mid で指定されたマシン ID のホスト名の別名として *alias* を追加します。

Change Name *mid principal-name*

mid で指定されたマシン ID を持つホストのホスト名を *principal-name* に変更します。

Create

マシン ID データベースを初期化します。

Delete Mid *mid*

マシン ID データベースから指定された *mid* のマシンを削除します。

Delete Name *alias*

マシン ID データベースから *alias* で指定された別名を削除します。

Exit

nmapmgr コマンドを終了します。

Get Mid *name*

name で指定されたホスト名、またはホスト名の別名を持つマシンのマシン ID を表示します。

Get Name *mid*

mid で指定されたマシン ID を持つマシンのホスト名を表示します。

Help

サブコマンド一覧を表示します。

Set Type *mid type*

mid で指定されたマシンに type で指定された NQS のタイプを設定します。

Show State [*name*]

name で指定されたホスト名、またはホスト名の別名を持つマシンのマッピング状態を表示します。 *name* を省略した場合はすべてのマシンのマッピング状態を示します。

Quit

nmapmgr コマンドを終了します。

ファイル

/etc/opt/nec/nqsv/nmap

/etc/opt/nec/nqsv/nmap/machines

注意

このコマンドは Sterling Software 社によって開発されたものをベースに機能拡張したものです。

2.3　nqs_ntfr(1M)

名前

nqs_ntfr ―実行ホストでの障害を BSV に通知
　　冗長化機能による監視対象ホストに関する通知

形式

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*

機能説明

nqs_ntfr は、実行ホストについての障害通知 と、冗長化機能による監視対象ホストに関する通知 の２つの機能があります。

1.　実行ホストについての障害通知

nqs_ntfr コマンドは、*hostname* もしくは *IP_address* で指定した実行ホストで障害が発生したことを、BSV に通知します。
障害通知を受けて BSV が JSV に対してどのように操作を行うかを、*operation* で指定します。

以下のオプションが指定できます。

-o *operation*
　障害通知時の JSV の操作を指定します。
　operation には、下記のいずれかが指定可能です。
　但し、JSV が LINKUP の状態の時のみ JSV を操作します。

nothing	JSV は操作しません。
down	JSV を LINKDOWN します。
unbind	JSV をキューから UNBIND します。

　-o オプションの指定が無い場合、down の動作になります。

-m *message*
　障害内容を *message* で指定します。
　message の最大文字列長は 255 バイトです。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示して終了します。

2.　冗長化機能による監視対象ホストに関する通知

nqs_ntfr コマンドは、*hostname* もしくは *IP_address* で指定した冗長化機能による監視対象ホスト関する情報を、ノード管理エージェントに通知します。

以下のオプションが指定できます。
但し、-r、-R、-s の各オプションはそれぞれ他のオプションとは排他です。

-r

障害を起こした BSV や JM 等が動作している監視対象ホストの復旧処理開始をノード管理エージェントに指示し、代替ホストの運用を停止させます。

-R

障害を起こした BSV や JM 等が動作している監視対象ホストの復旧処理完了をノード管理エージェントに通知します。
起動や停止に失敗した代替ホストの復旧完了の通知も行います。この場合は、復旧する代替ホストを引数に *hostname* もしくは *IP_address* で指定します。

-s

BSV や JM が動作している監視対象ホストの障害発生をノード管理エージェントに通知し、代替ホストへの切り替えを開始します。

関連項目

qrls(1), qhold(1), qstat(1)

2.4　qmgr (1M)

名前

qmgr ― NQSV の構成管理および運用管理

形式

qmgr [-P *privilege*] [-V] [-h *batch_server_host*]

機能説明

qmgr は、NQSV システムの管理者と操作員が、batch_server_host 上の NQSV を制御するために使用するプログラムです。以下のオプションが指定できます。

-h *batch_server_host*

NQSV システムの管理操作を行う対象のバッチサーバのホスト名を指定します。 本オプションが指定されない場合、/etc/opt/nec/nqsv/api_client.conf ファイルに指定されたバッチサーバホストを使用します。/etc/opt/nec/nqsv/api_client.conf ファイルにも指定がない場合は、ローカルホスト上のバッチサーバが対象となります。

-P *privilege*

コマンド実行時のアクセス権を指定します。 *privilege* には次の指定ができます。

m	管理者権限
o	操作員権限
g	グループ管理者権限
s	特別利用者権限
u	一般利用者権限

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 NQSV_PRIV または NQSII_PRIV で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している NQSV API のバージョンを表示します。

qmgr コマンドを起動すると、qmgr はプロンプト Mgr:を表示し、入力待ちとなります。このプロンプト上で下記のサブコマンドを実行することによって、NQSV に対する各種の管理操作が行えます。
qmgr コマンドを終了するには exit サブコマンドを実行するか、プロンプト上で、ctrl-D を押下します。
また、Mgr:プロンプト行での文字入力中に、以下の編集コマンドが使用できます。

Ctrl-f	順方向に一文字進みます
Ctrl-b	逆方向に一文字戻ります

Ctrl-h	前の文字を削除します
Ctrl-p	一つ前の入力コマンドを表示します
Ctrl-n	一つ後の入力コマンドを表示します
<TAB>	入力中の文字列に一致するサブコマンドを表示もしくは補完します

qmgr は、サブコマンドによる出力（help サブコマンド、show サブコマンド）に対して画面のページングを行います。ページングに使用するプログラムは、PAGER 環境変数によって指定することができます（既定値は、/bin/more です）。

サブコマンド

以下に、qmgr で実行できるサブコマンドについて記述します。
qmgr の各サブコマンドには、実行に必要なアクセス権が決まっています。 管理者権限はすべてのサブコマンドを使用することが許されていますが、操作員権限はある一部の制限されたコマンドのみ使用することができます。 一般利用者権限および特別利用者権限では、exit, help, list および、show サブコマンドのみが使用できます。
予算管理機能に関するサブコマンドの詳細は［アカウンティング・予算管理編］を参照してください。

abort execution_queue = *queue* [*seconds*]

queue に指定されたバッチキュー内で現在実行しているすべてのリクエストを以下のようにアボートさせます(対象となるリクエストの状態は RUNNING, SUSPENDING, SUSPENDED, RESUMING, CHKPNTING)。
まず、現在そのキューで実行中のリクエストの各プロセスに SIGTERM シグナルを送ります。 次に *seconds* に指定された秒数が経過した後、指定されたキュー内で実行している各リクエストの残存プロセスに SIGKILL シグナルを送ります。 *seconds* の指定がない場合の待ち合わせ時間の既定値は、60 秒になります。
このコマンドによってアボートされるすべてのリクエストはキュー上から削除され、そのリクエストの結果ファイルを返します。

【アクセス権】対象リクエストの所有者もしくは管理者権限が必要です。

abort interactive_queue = *queue* [*seconds*]

queue に指定された会話キュー内で現在実行しているすべてのリクエストを以下のようにアボートさせます(対象となるリクエストの状態は RUNNING, SUSPENDING, SUSPENDED, RESUMING)。
まず、現在そのキューで実行中のリクエストの各プロセスに SIGTERM シグナルを送ります。 次に *seconds* に指定された秒数が経過した後、指定されたキュー内で実行している各リクエストの残存プロセスに SIGKILL シグナルを送ります。 *seconds* の指定がない場合の待ち合わせ時間の既定値は、60 秒になります。
このコマンドによってアボートされるすべてのリクエストはキュー上から削除され、実

行が終了します。

【アクセス権】対象リクエストの所有者もしくは管理者権限が必要です。

add execution_queue groups = *group queue*
add execution_queue groups = (*group* [, *group* ...]) *queue*

queue で指定されたバッチキューのグループリストに *group* で指定されたグループを追加します。グループリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

add execution_queue users = *user queue*
add execution_queue users = (*user* [, *user* ...]) *queue*

queue で指定されたバッチキューのユーザリストに *user* で指定されたユーザを追加します。ユーザリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

add interactive_queue groups = *group queue*
add interactive_queue groups = (*group* [, *group* ...]) *queue*

queue で指定された会話キューのグループリストに *group* で指定されたグループを追加します。グループリストのモードは、**set interactive_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

add interactive_queue users = *user queue*
add interactive_queue users = (*user* [, *user* ...]) *queue*

queue で指定された会話キューのユーザリストに *user* で指定されたユーザを追加しま

す。ユーザリストのモードは、**set interactive_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

add routing_queue destination = *destination queue* [*position*]
add routing_queue destination = (*destination* [, *destination* ...]) *queue* [*position*]

queue で指定された転送キューの転送先キューとして、*destination*(キュー名[@キューが存在するバッチサーバホスト名]) で指定されたキューが追加されます。転送キューに転送先キューの設定が複数存在する場合は、設定された順番に転送先キューへの転送が試みられます。
position には、*queue* で指定した転送キューにすでに設定されている転送先キューと *destination* で指定した転送先キューの設定の順番を指定します。*position* には以下の指定が可能です。

after *queue_name*	新しく指定した転送先キューが *queue_name* で指定したキューの後になるよう設定します。
before *queue_name*	新しく指定した転送先キューが *queue_name* で指定したキューの前になるよう設定します。
first	新しく指定した転送先キューが転送先の先頭になるよう設定します。
last	新しく指定した転送先キューが転送先の最後になるよう設定します。(*position* を指定しない場合の既定値)

【アクセス権】操作員権限が必要です。

add routing_queue groups = *group queue*
add routing_queue groups = (*group* [, *group* ...]) *queue*

queue で指定された転送キューのグループリストに *group* で指定されたグループを追加します。グループリストのモードは、**set routing_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

add routing_queue users = *user queue*
add routing_queue users = (*user* [, *user* ...]) *queue*

queue で指定された転送キューのユーザリストに *user* で指定されたユーザを追加します。ユーザリストのモードは、**set routing_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

attach baremetal_host host = *host_name*　**job_server_id** = *jsv_id*　**cpu** = *cpunum*
　　memory = *memsz*　**gpu** = *gpunum*
attach baremetal_host file = *file_path*

ベアメタルサーバをバッチサーバに登録します。
登録の方法には、ベアメタルサーバの情報をサブコマンドで指定する方法と、ファイルで指定する方法があります。

サブコマンドで指定する場合、登録するベアメタルサーバのホスト名、その実行ホスト上で起動するジョブサーバのジョブサーバ番号、CPU 数、メモリサイズ、GPU 数を、それぞれ、*host_name* 、*jsv_id* 、*cpunum* 、*memsz*、*gpunum* に指定します。

ファイルを指定する場合、*file_ path* に、ベアメタルサーバの情報を記載したファイルパスを指定します。ファイルには、以下のように、ベアメタルサーバのホスト名、ジョブサーバ番号、CPU 数、メモリ量、GPU 数を空白文字で区切って記載します。

```
#HOSTNAME JSVNO CPU  MEM   GPU
Bhost1      101  4   10GB   0
Bhost2      102  4   20GB   1
```

【アクセス権】管理者権限が必要です。

attach execution_host host = *host_name*　**job_server_id** = *jsv_id*
attach execution_host file = *file_name*

NQSV システムの実行ホストをバッチサーバに登録します。
登録の方法には、登録する実行ホストのホスト名と、その実行ホスト上で起動するジョブサーバのジョブサーバ番号をそれぞれ、*host_name* 、*jsv_id* に指定する方法と、登録対象のホスト名とジョブサーバ番号のリストを記載したファイルを *file_name* に指定する方法があります。
ファイル指定の場合、*file_name* に指定するファイルには、以下のようにホスト名とジョブサーバ番号を記載しておきます(ファイル中、#文字で始まる行はコメントとして扱

います)。

```
#HOSTNAME     JSVNO
host_001          1
host_002          2
host_003          3
```

【アクセス権】管理者権限が必要です。

bind execution_queue job_server *queue*　**job_server_id** = *n*

queue で指定されたバッチキューと *n* で指定されたジョブサーバ ID に一致するジョブサーバを接続します。

【アクセス権】管理者権限が必要です。

bind execution_queue node_group *queue*　**node_group** = *ngrp_name*

queue で指定されたバッチキューと *ngrp_name* で指定されたノードグループを接続します。

【アクセス権】管理者権限が必要です。

bind execution_queue scheduler *queue*　**scheduler_id** = *n*

queue で指定されたバッチキューと *n* で指定されたスケジューラ ID に一致するスケジューラを接続します。

【アクセス権】管理者権限が必要です。

bind interactive_queue job_server *queue*　**job_server_id** = *n*

queue で指定された会話キューと *n* で指定されたジョブサーバ ID に一致するジョブサーバを接続します。

【アクセス権】管理者権限が必要です。

bind interactive_queue node_group *queue*　**node_group** = *ngrp_name*

queue で指定された会話キューと *ngrp_name* で指定されたノードグループを接続します。

【アクセス権】管理者権限が必要です。

bind interactive_queue scheduler *queue*　**scheduler_id** = *n*

queue で指定された会話キューと *n* で指定されたスケジューラ ID に一致するスケジュ

ーラを接続します。

【アクセス権】管理者権限が必要です。

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*"]

template_name で指定した名前のクラウドバースティング用テンプレート情報を作成します。

テンプレート情報として、以下を指定します。

image	起動するクラウドインスタンスのイメージ名。
cpu	割り当てる CPU 数。 １以上の整数を指定します。
memsz	割り当てるメモリサイズ。 1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。
gpu	割り当てる GPU 数。 0 以上の整数を指定します。 既定値は 0 です。
ve	割り当てる VE 数。 0 以上の整数を指定します。 既定値は 0 です。
boot_timeout	クラウドインスタンスの起動見込時間。 秒単位で、1～2147483647 の整数を指定します。 既定値は 300 秒です。
stop_timeout	クラウドインスタンスの停止見込時間。 秒単位で、1～2147483647 の整数を指定します。 既定値は 300 秒です。
custom	起動環境として独自に定義した情報。
comment	テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

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*"]

template_name で指定した名前のコンテナ用テンプレート情報を作成します。

テンプレート情報として、以下を指定します。

image　　起動するコンテナのイメージ名。

cpu　　割り当てる CPU 数。
１以上の整数を指定します。

memsz　　割り当てるメモリサイズ。
1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。

gpu　　割り当てる GPU 数。
0 以上の整数を指定します。

ve　　割り当てる VE 数。
0 以上の整数を指定します。

hca　　割り当てる HCA ポート数。
hcanum は、<io>,<mpi>,<all>の形式で指定します。
<io>,<mpi>,<all>には 0 以上の整数を指定します。
例）共用の HCA ポート数に 1 を設定する場合
hca = (0,0,1)

boot_timeout　　起動見込時間。
秒単位で、1～2147483647 の整数を指定します。
既定値は 900 秒です。

stop_timeout　　停止見込時間。
秒単位で、1～2147483647 の整数を指定します。
既定値は 900 秒です。

custom　　起動環境として独自に定義した情報。

comment　　テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

create custom_resource = *cr_name*　**consumer** = { **job** | **request** }
[**check_mode** = {**off** | **moment** | **integrate**}]
[**terminate_job** = {**off** | **on**}]
[unit = *unit*]
[**type** = { **bsv** | **host** }　**available** = *n*]

cr_name で指定した名前のカスタムリソース情報を作成します。

consumer には、カスタムリソースを消費する単位 job, request のいずれかを指定します。

check_mode には、このカスタムリソースを使用した資源監視及び使用量の出力について off、moment、integrate のいずれかを指定します。省略時は off として扱います。

terminate_job には、このカスタムリソースについて資源超過時のジョブ終了機能を off、

on のいずれかで指定します。省略時は off として扱います。
unit には、このカスタムリソースの単位を指定します。省略時は単位を設定しません。
type には、カスタムリソースの利用量制御の対象種別 bsv, host のいずれかを指定します。
available には、**type** で指定した対象種別で同時に利用可能なカスタムリソースの利用量の最大値を、0～2147483647 の整数で指定します。

【アクセス権】管理者権限が必要です。

create execution_queue = *queue* **priority =** *n*

queue で指定された名前のバッチキューをキュープライオリティ *n* (0～63)で作成します。 63 は最も高い優先度を定義し、0 は最も低い優先度を定義します。

【アクセス権】管理者権限が必要です。

create interactive_queue = *queue* **priority =** *n*

queue で指定された名前の会話キューをキュープライオリティ *n* (0～63)で作成します。63 は最も高い優先度を定義し、0 は最も低い優先度を定義します。

【アクセス権】管理者権限が必要です。

create network_queue = *queue* **staging_machine =** *staging_machine* **priority =** *n* [**run_limit =** *run_limit*]

queue に指定した名前のネットワークキューをプライオリティ *n* で作成します。
staging_machine には、ステージングファイルの転送先または転送元のホスト名を指定します。(ホスト名の代わりに、マシン ID [<*machine-id*>] を指定することも可能です。) たとえばホスト名が"machine1"、マシン ID が 100 の場合、"machine1" か "[100]" のどちらかを指定します。

ネットワークキューは結果ファイル転送用のキューで、転送先ごとに作成しなければなりません。 もし転送先ホストに対応するネットワークキューがなければ、デフォルト・ネットワークキュー (DefaultNetQue) が用いられます。 1 つのホストについて複数のネットワークキューを作成できますが、使われるのはキュープライオリティの最も高いキューだけで、その他のキューは無視されます。 1 つのネットワークキューに複数のホストを指定することはできません。また、キュー作成後にホストの変更はできません。

run_limit は、個々のネットワークキューで同時に実行できるリクエスト数を指定します。
既定値は 1 です。

【アクセス権】管理者権限が必要です。

create node_group = *ngrp_name* [**type** = *ngrp_type*] [**comment** = "*comment*"]
[**switch_layer** = *layer*] [**template** = *template_name*] [**priority** = *priority*]
[**network** = *network_name*]
ngrp_name に指定した名前のノードグループを作成します。*ngrp_type* にはノードグループのタイプを指定します。
指定可能なタイプは以下のとおりです。

common	一般的なノードグループ（既定値）
nw_topo	ネットワークトポロジーを考慮したスケジューリングを行うための同一スイッチ配下の実行ホスト群のグループを示すためのノードグループ
cloud	クラウドバースティングを行うためのノードグループ

comment に、生成するノードグループに対するコメント文字列を指定することができます。

layer にスイッチレイヤーのレベルを指定します。1～4 が指定できます。*ngrp_type* に nw_topo を指定した場合のみ設定が可能です。既定値は 1 です。

template_name に予め定義されているクラウドバースティング用テンプレートを指定します。テンプレートは起動するインスタンスの諸元を示します。*ngrp_type* に cloud を指定した場合のみ設定が可能です。

priority にクラウドバースティングノードグループの優先度を指定します。0～63 が指定できます。*ngrp_type* に cloud を指定した場合のみ設定が可能です。既定値は 1 です。

*network_name*に起動するクラウドインスタンスが置かれているクラウド環境のネットワークの名前を指定します。クラウド環境のネットワークを区別する必要がある場合に指定します。最大は 255 文字まで指定できます。*ngrp_type* に cloud を指定した場合のみ設定が可能です。

【アクセス権】管理者権限が必要です。

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*"]
template_name で指定した名前の OpenStack テンプレート情報を作成します。

テンプレート情報として、以下を指定します。

image	OpenStack が起動する OS ディスクイメージ名。
flavor	OpenStack のフレーバ名。
cpu	割り当てる CPU 数。 １以上の整数を指定します。
memsz	割り当てるメモリサイズ。 1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。
gpu	割り当てる GPU 数。 ０以上の整数を指定します。
boot_timeout	起動見込時間。 秒単位で、1～2147483647 の整数を指定します。 既定値は 900 秒です。
stop_timeout	停止見込時間。 秒単位で、1～2147483647 の整数を指定します。 既定値は 900 秒です。
custom	起動環境として独自に定義した情報。
comment	テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

create routing_queue = *queue*　**priority** = *n* [**run_limit** = *run_limit*]
[**destination** = <*destination*>]
[**destination** = (<*destination*>[,<*destination*>...])]

queue で指定した名前の転送キューをキュープライオリティ *n* (0～63)で作成します。63 は最も高い優先度を定義し、0 は最も低い優先度を定義します。*destination* には、この転送キューに投入されたリクエストを転送する、1 つもしくはそれ以上の宛先キューのリストを指定します。
run_limit には、個々の転送キューで同時に実行できるリクエスト数を指定します。既定値は 1 です。

【アクセス権】管理者権限が必要です。

delete batch_server group_submit_limit groups = *group*
delete batch_server group_submit_limit groups = (*group*[, *group* …])
group で指定したグループのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

delete batch_server user_submit_limit users = *user*
delete batch_server user_submit_limit users = (*user*[, *user* …])
user で指定したユーザのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

delete cloud_template = *template_name*
template_name で指定した名前のクラウドバースティング用テンプレート情報を削除します。
但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、削除できません。

【アクセス権】管理者権限が必要です。

delete container_template = *template_name*
template_name で指定した名前のコンテナ用テンプレート情報を削除します。
但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、削除できません。

【アクセス権】管理者権限が必要です。

delete custom_resource = *cr_name*
cr_name で指定した名前のカスタムリソース情報を削除します。

【アクセス権】管理者権限が必要です。

delete openstack_template = *template_name*
template_name で指定した名前の OpenStack テンプレート情報を削除します。
但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、削除できません。

【アクセス権】管理者権限が必要です。

delete execution_queue groups = *group queue*
delete execution_queue groups = (*group* [, *group* ...]) *queue*
queue で指定されたバッチキューのグループリストから *group* で指定されたグループを削除します。グループリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを削除する場合、

group として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete execution_queue group_submit_limit groups = *group*　*queue*
delete execution_queue group_submit_limit groups = (*group*[,*group* …])　*queue*
キュー*queue* について、*group* で指定したグループのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

delete execution_queue hold_privilege *queue*
queue で指定されたバッチキューに設定されている、ホールド制限を解除します。

【アクセス権】操作員権限が必要です。

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*
キュー*queue* について、*group* または *user* で指定した、グループまたはユーザのリクエストのジョブ数の上限値／下限値の設定を削除します。

【アクセス権】操作員権限が必要です。

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*
キュー*queue* について、*group* または *user* で指定した、グループまたはユーザのリクエスト単位の経過時間制限値の設定を削除します。

【アクセス権】操作員権限が必要です。

delete execution_queue queue = *queue*
queue で指定されたバッチキューを削除します。バッチキューを削除するためには、そのキュー上にリクエストがなく、かつそのキューが受付不可状態（**disable execution_queue** 参照)であることが必要です。

【アクセス権】管理者権限が必要です。

delete execution_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定されたバッチキューにおける、リクエスト投入経路に対する投入拒否設定から、*from* で指定された投入経路に対する拒否を解除します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit	qsub からのリクエスト投入
qmove	qmove からのリクエスト投入
local_routing	ローカルからのリクエスト転送
remote_routing	リモートからのリクエスト転送

【アクセス権】操作員権限が必要です。

delete execution_queue suspend_privilege　*queue*

queue で指定されたバッチキューに設定されているサスペンド制限を解除します。

【アクセス権】操作員権限が必要です。

delete execution_queue userexit location **=** *location*　**queue** = *queue*

queue で指定されたバッチキューに設定されているユーザ EXIT のスクリプトから、*location* で指定されたタイミングに実行するように設定されているスクリプトを削除します。*location* に指定できるのは以下のタイミングです。

pre-running	実行開始前
post-running	実行終了直後

【アクセス権】操作員権限が必要です。

delete execution_queue users = *user queue*

delete execution_queue users = (*user* [, *user* ...]) *queue*

queue で指定されたバッチキューのユーザリストから *user* で指定されたユーザを削除します。ユーザリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを削除する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete execution_queue user_submit_limit users = *user*　*queue*

delete execution_queue user_submit_limit users = (*user*[,*user* ...])　*queue*

キュー*queue* について、*user* で指定したユーザのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です

delete interactive_queue groups = *group queue*

delete interactive_queue groups = (*group* [, *group* ...]) *queue*

queue で指定された会話キューのグループリストから *group* で指定されたグループを削除します。グループリストのモードは、**set interactive_queue access** | **noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを削除する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete interactive_queue group_submit_limit groups = *group*　*queue*

delete interactive_queue group_submit_limit groups = (*group*[,*group* ...])　*queue*

キュー*queue* について、*group* で指定したグループのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

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*

キュー*queue* について、*group* または *user* で指定した、グループまたはユーザのリクエストのジョブ数の上限値／下限値の設定を削除します。

【アクセス権】操作員権限が必要です。

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*

キュー*queue* について、*group* または *user* で指定した、グループまたはユーザのリクエスト単位の経過時間制限値の設定を削除します。

【アクセス権】操作員権限が必要です。

delete interactive_queue queue = *queue*

queue で指定された会話キューを削除します。会話キューを削除するためには、そのキュー上にリクエストがなく、かつそのキューが受付不可状態（**disable interactive_queue** 参照)であることが必要です。

【アクセス権】管理者権限が必要です。

delete interactive_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定された会話キューにおける、リクエスト投入経路に対する投入拒否設定から、*from* で指定された投入経路に対する拒否を解除します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit　　qlogin または qrsh からのリクエスト投入

【アクセス権】操作員権限が必要です。

delete interactive_queue restrict_shell *queue*

queue で指定した会話キューに設定されている強制実行シェルの設定を削除します。

【アクセス権】操作員権限が必要です。

delete interactive_queue suspend_privilege　*queue*

queue で指定された会話キューに設定されているサスペンド制限を解除します。

【アクセス権】操作員権限が必要です。

delete interactive_queue userexit location **=** *location*　**queue** = *queue*

queue で指定された会話キューに設定されているユーザ EXIT のスクリプトから、*location* で指定されたタイミングに実行するように設定されているスクリプトを削除します。*location* に指定できるのは以下のタイミングです。

pre-running　実行開始前
post-running　実行終了直後

【アクセス権】操作員権限が必要です。

delete interactive_queue users = *user queue*
delete interactive_queue users = (*user* [, *user* ...]) *queue*

queue で指定された会話キューのユーザリストから *user* で指定されたユーザを削除します。ユーザリストのモードは、**set interactive_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを削除する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete interactive_queue user_submit_limit users = *user　queue*
delete interactive_queue user_submit_limit users = (*user*[,*user* …])　*queue*

キュー*queue* について、*user* で指定したユーザのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

delete network_queue queue = *queue*

queue で指定されたネットワークキューを削除します。ネットワークキューを削除するためには、そのネットワークキュー上にリクエストがなく、かつそのキューが受付不可状態（**disable network_queue** 参照)であることが必要です。
システムに既定値として設定されている DefaultNetQue は、このサブコマンドで削除できません。

【アクセス権】管理者権限が必要です。

delete node_group = *ngrp_name*

ngrp_name に指定した名前のノードグループを削除します。ノードグループの削除は、ノードグループがキューとバインドされていない状態で実行してください。

【アクセス権】管理者権限が必要です。

delete routing_queue destination = *destination queue*
delete routing_queue destination = (*destination* [, *destination* ...]) *queue*

queue で指定した転送キューの宛先キューリストから *destination* (キュー名[@キューが存在するバッチサーバホスト名]) で指定した宛先キューを削除します。 削除の実行時に宛先キューへ転送中のリクエストが存在した場合、そのリクエストの転送は最後まで実行されます。
転送キューのすべての宛先キューが削除された場合、その転送キューは停止状態と同様の状態になります。

【アクセス権】操作員権限が必要です。

delete routing_queue groups = *group queue*
delete routing_queue groups = (*group* [, *group* ...]) *queue*

queue で指定された転送キューのグループリストから *group* で指定されたグループを削除します。グループリストのモードは、**set routing_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを削除する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete routing_queue group_submit_limit groups = *group　queue*
delete routing_queue group_submit_limit groups = (*group*[,*group* ...])　*queue*

キュー*queue* について、*group* で指定したグループのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

delete routing_queue queue = *queue*

queue で指定された転送キューを削除します。転送キューを削除するためには、その転送キュー上にリクエストがなく、かつそのキューが受付不可状態（**disable routing_queue** 参照)であることが必要です。

【アクセス権】管理者権限が必要です。

delete routing_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定された転送キューにおける、リクエスト投入経路に対する投入拒否設定から、*from* で指定された投入経路に対する拒否を解除します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit	qsub からのリクエスト投入
qmove	qmove からのリクエスト投入
local_routing	ローカルからのリクエスト転送
remote_routing	リモートからのリクエスト転送

【アクセス権】操作員権限が必要です。

delete routing_queue users = *user queue*

delete routing_queue users = (*user* [, *user* ...]) *queue*

queue で指定された転送キューのユーザリストから *user* で指定されたユーザを削除します。ユーザリストのモードは、**set routing_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。

たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを削除する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

delete routing_queue user_submit_limit users = *user*　*queue*

delete routing_queue user_submit_limit users = (*user*[,*user* ...])　*queue*

キュー*queue* について、*user* で指定したユーザのリクエストの投入数の最大数の設定を削除します。

【アクセス権】操作員権限が必要です。

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

バッチサーバに登録されているベアメタルサーバの登録を削除します。削除を行うベアメタルサーバのジョブサーバを使用するジョブが存在しない状態で、ジョブサーバがリンクダウンしている必要があります。

削除対象のベアメタルサーバの指定方法は、host =で *host_name* にホスト名を指定する方法と、job_server_id = に対して、*jsv_id* にジョブサーバ番号を指定する方法、および、all 指定でバッチサーバに登録されている全ベアメタルサーバを一括して削除する方法があります。

ホスト名指定の場合、() の中にカンマで区切って複数のホスト名を指定することが可能です。また、ジョブサーバ番号指定の場合は、() の中にカンマで区切って複数のジョブサーバ番号を指定する方法に加えて、*jsv_id1-jsv_id2* のようにジョブサーバ番号の範囲指定が可能です。

【アクセス権】管理者権限が必要です。

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

NQSV システムの実行ホストとしてバッチサーバに登録されている実行ホストの登録を削除します。登録の削除を行う実行ホストは、そのホストのジョブサーバを使用するジョブが存在しない状態で、ジョブサーバがリンクダウンしている必要があります。
削除対象の実行ホストの指定方法は、host =で *host_name* にホスト名を指定する方法と、job_server_id = に対して、*jsv_id* にジョブサーバ番号を指定する方法、および、all 指定でバッチサーバに登録されている全実行ホストを一括して削除する方法があります。
ホスト名指定の場合、() の中にカンマで区切って複数のホスト名を指定することが可能です。また、ジョブサーバ番号指定の場合は、() の中にカンマで区切って複数のジョブサーバ番号を指定する方法に加えて、*jsv_id1-jsv_id2* のようにジョブサーバ番号の範囲指定が可能です。

【アクセス権】管理者権限が必要です。

disable execution_queue = *queue*

queue で指定されたバッチキューをリクエスト受付不可状態にします。 受付不可状態のバッチキューに新たなリクエストは投入できません。

【アクセス権】操作員権限が必要です。

disable interactive_queue = *queue*

queue で指定された会話キューをリクエスト受付不可状態にします。 受付不可状態の会話キューに新たな会話リクエストは投入できません。

【アクセス権】操作員権限が必要です。

disable network_queue = *queue*

queue で指定されたネットワークキューをリクエスト受付不可状態にします。 受付不可状態のネットワークキューに新たなリクエストは投入できません。

システムに既定値として設定されている DefaultNetQue は、このサブコマンドで受付不可状態にすることはできません。

【アクセス権】操作員権限が必要です。

disable routing_queue = *queue*

queue で指定された転送キューをリクエスト受付不可状態にします。 受付不可状態の

転送キューに新たなリクエストは投入できません。

【アクセス権】操作員権限が必要です。

edit custom_resource add type = **bsv**　**available** = *n*　*cr_name*
edit custom_resource add type = **host**　［**target** = *host_name*］
　　available = *n*　*cr_name*

cr_name で指定した名前のカスタムリソース情報について、**type**、および **target** で指定した利用量制御情報を追加します。

type にはカスタムリソースの利用量制御の対象種別を指定します。

bsv　利用量制御の対象を BSV とし、BSV 全体で同時に利用可能なカスタムリソースの量を制限し、スケジューリングします。

host　利用量制御の対象を実行ホストとし、1 実行ホスト当たりで同時に利用可能なカスタムリソースの量を制限し、スケジューリングします。
但し、カスタムリソース *cr_name* の消費単位が、request に設定されている場合、host の指定はできません。
また、対象種別が host の場合、**target** で個別にホスト名 *host_name* を合わせて指定することも可能です。

available には、**type** および **target** で指定した対象での同時に利用可能なカスタムリソースの利用量の最大値を、0～2147483647 の整数で指定します。

既に設定されている場合は、上書きされます。

【アクセス権】管理者権限が必要です。

edit custom_resource delete type = **bsv**　*cr_name*
edit custom_resource delete type = **host**　［**target** = *host_name*］　*cr_name*

cr_name で指定した名前のカスタムリソース情報について、**type**、および **target** で指定した利用量制御情報を削除します。

type にはカスタムリソースの利用量制御の対象種別を指定します。

bsv　BSV 全体で同時に利用可能なカスタムリソース量の設定を削除します。

host　実行ホスト当たりで同時に利用可能なカスタムリソース量の設定を削除します。
但し、**target** 指定が無い場合、対象種別 host の情報を一括削除します。
また、**target** が指定されている場合、指定された実行ホスト *host_name* の設定のみ削除します。

【アクセス権】管理者権限が必要です。

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_ngrp*, *src_ngrp*2 ...) *ngrp_name*

ngrp_name で指定したノードグループにジョブサーバを追加します。追加するジョブサーバの指定方法には以下の方法があります。

・ジョブサーバ番号 *n* を指定する
・*n*,*m*...のようにカンマで区切って複数のジョブサーバ番号を指定する
・*n*-*m* のようにハイフンでジョブサーバ番号の範囲を指定する
・*src_ngrp* に、既に生成済みのノードグループを指定して、そのノードグループに属する全ジョブサーバを新しいノードグループに追加する
・*src_ngrp*, *src_ngrp*2 のようにカンマで区切って指定するノードグループを複数指定する

cloud タイプのノードグループに追加するジョブサーバは他のノードグループにまたがって属することはできません。

【アクセス権】操作員権限が必要です。

edit node_group add template = *template_name*　*ngrp_name*
edit node_group add template = (*template_name1*, *template_name*2 ...) *ngrp_name*

ngrp_name で指定したクラウドバースティングノードグループにクラウドバースティング用テンプレートを追加します。複数のテンプレートをまとめて追加する場合は、*template_name1*, *template_name*2 のようにカンマで区切って指定します。
1 つのノードグループに追加できるテンプレートの数は最大 20 です。

【アクセス権】操作員権限が必要です。

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_ngrp*, *src_ngrp*2 ...) *ngrp_name*

ngrp_name で指定したノードグループからジョブサーバを削除します。削除するジョブサーバの指定方法には以下の方法があります。

・ジョブサーバ番号 *n* を指定する
・*n*,*m*...のようにカンマで区切って複数のジョブサーバ番号を指定する
・*n*-*m* のようにハイフンでジョブサーバ番号の範囲を指定する
・*src_ngrp* に、既に生成済みのノードグループを指定して、そのノードグループに属する全ジョブサーバを *ngrp_name* で指定したノードグループから削

除する
・ *src_ngrp*, *src_ngrp2* のようにカンマで区切って指定するノードグループを複数指定する

【アクセス権】操作員権限が必要です。

edit node_group delete template = *template_name　ngrp_name*
edit node_group delete template = (*template_name1*, *template_name2* ...) *ngrp_name*
ngrp_name で指定したクラウドバースティングノードグループからクラウドバースティング用テンプレートを削除します。複数のテンプレートをまとめて削除する場合は *template_name1*, *template_name2* のようにカンマで区切って指定します。

【アクセス権】操作員権限が必要です。

enable execution_queue = *queue*
queue で指定されたバッチキューをリクエスト受付可能状態にします。 バッチキューを受付可能状態にすることによって、そのバッチキューにリクエストを投入できるようになります。

【アクセス権】操作員権限が必要です。

enable interactive_queue = *queue*
queue で指定された会話キューをリクエスト受付可能状態にします。 会話キューを受付可能状態にすることによって、その会話キューに会話リクエストを投入できるようになります。

【アクセス権】操作員権限が必要です。

enable network_queue = *queue*
queue で指定されたネットワークキューをリクエスト受付可能状態にします。 ネットワークキューを受付可能状態にすることによって、そのネットワークキューにリクエストを投入できるようになります。

【アクセス権】操作員権限が必要です。

enable routing_queue = *queue*
queue で指定された転送キューをリクエスト受付可能状態にします。 転送キューを受付可能状態にすることによって、その転送キューにリクエストを投入できるようになります。

【アクセス権】操作員権限が必要です。

exit

qmgr コマンドを終了します。

help [*command*]

help 情報を表示します。引数なしで help コマンドを使用すると、使用できるコマンドについての情報を表示します。 引数を付けて help コマンドを使用すると、指定されたコマンドについての情報を表示します。 コマンドの指定はそれがユニークである限りは部分的に指定してもかまいません。 より詳しい形で *command* 引数を指定すれば、受け取る情報もより詳しくなります。

hold job_server　*job_server_id*

job_server_id で指定したジョブサーバ上で現在実行しているすべてのリクエストをホールドします(対象となるリクエストの状態は RUNNING)。 ただし、リクエストがホールド禁止に設定されている場合はホールドされません。

【アクセス権】操作対象リクエストの所有者もしくは操作員権限が必要です。

list all [**file** = *file_name*]

現在運用されている NQSV バッチサーバ、実行ホスト（ジョブサーバ)、ノードグループ、キュー、カスタムリソース、テンプレートに関する情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。ただし、bind 関連のサブコマンドについては出力しません。できあがったファイルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

list batch_server [**file** = *file_name*]

現在運用されている NQSV バッチサーバに関する情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。 できあがったファイルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

list bind [**file** = *file_name*]

現在運用されている NQSV キューとジョブサーバの bind 情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。 できあがったファイ

ルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

list custom_resource [**file** = *file_name*]

現在運用されているカスタムリソース情報の定義を、**qmgr** サブコマンドの形に変換して、*file_name* で指定したファイルに保存します。

list node [**file** = *file_name*]

現在運用されている NQSV の実行ホスト（ジョブサーバ）とノードグループに関する情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。できあがったファイルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

list queue [**file** = *file_name*]

現在運用されている NQSV キューに関する情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。ただし、bind 関連のサブコマンドについては出力しません。できあがったファイルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

list template [**file** = *file_name*]

現在運用されている NQSV のテンプレートに関する情報を **qmgr** のサブコマンドの形に変換して、*file_name* で指定したファイルに保存します。できあがったファイルを **qmgr** に読み込ませることで、NQSV 環境の復旧が容易に行えます。

file_name で指定したファイルと同名のファイルが存在する場合は、強制的に上書きします。*file_name* を省略した場合、標準出力に出力します。

load nqsd_conf

nqsd.conf ファイルの設定を読み込みます。

【アクセス権】管理者権限が必要です。

lock cloud_template = t*emplate_name*

template_name で指定した名前のクラウドバースティング用テンプレートの使用をロックします。

【アクセス権】管理者権限が必要です。

lock container_template = t*emplate_name*

template_name で指定した名前のコンテナ用テンプレートの使用をロックします。

【アクセス権】管理者権限が必要です。

lock node_group = *ngrp_name*

ngrp_name で指定した名前のクラウドバースティングノードグループの使用をロックします。

【アクセス権】管理者権限が必要です。

lock openstack_template = t*emplate_name*

template_name で指定した名前の OpenStack テンプレートの使用をロックします。

【アクセス権】管理者権限が必要です。

migrate job = *job_id*　**job_server_id** = *job_server_id*

ジョブマイグレーション機能により、*job_id* で指定したジョブを *job_server_id* で指定したジョブサーバへ移動します。 HELD 状態のリクエストのみ移動が可能です。

【アクセス権】操作員権限が必要です。

purge execution_queue = *queue*

queue で指定されたバッチキュー内のすべてのキューイング状態のリクエスト(対象となるリクエストの状態は QUEUED,WAITING,HELD,STAGING)を削除します。

【アクセス権】対象リクエストの所有者もしくは管理者権限が必要です。

purge interactive_queue = *queue*

queue で指定された会話キュー内のすべてのキューイング状態のリクエスト(対象となるリクエストの状態は QUEUED,WAITING,STAGING)を削除します。

【アクセス権】対象リクエストの所有者もしくは管理者権限が必要です。

purge routing_queue = *queue*

queue で指定された転送キュー内のすべてのキューイング状態のリクエスト(対象となるリクエストの状態は QUEUED,WAITING,HELD)を削除します。

【アクセス権】対象リクエストの所有者もしくは管理者権限が必要です。

reset baremetal_host host = *host_name*

障害等によって LINKDOWN したベアメタルサーバについて、紐付けられたテンプレート情報をクリアします。
対象ホスト上にストール中のジョブが無い状態で実行してください。

【アクセス権】管理者権限が必要です。

set batch_server acct_server [**hostname =** *host_name*] [**port =** *port_number*]

予算超過チェック時に使用するサーバの設定を行います。
host_name には、ASV のアカウンティングサーバのホスト名を指定します。
port_number には、サーバが予算超過チェック要求を待つ TCP ポート番号を指定します。

【アクセス権】操作員権限が必要です。

set batch_server allow_absolute_exepath { **on** | **off** }

外部ステージングにおけるステージングファイルの絶対パス指定の可否を設定します。

on　絶対パス指定を許可します
off　絶対パス指定を拒否します（既定値）

【アクセス権】管理者権限が必要です。

set batch_server auto_delete_failed_request { **on** | **off** }

投入に失敗した連携リクエストを自動的に削除するか否かを設定します。

on　投入に失敗した連携リクエストを自動的に削除します。(既定値)
off　投入に失敗した連携リクエストを自動的に削除しません。そのリクエストは HELD 状態で残ります。

【アクセス権】管理者権限が必要です。

set batch_server nqs_budget_chk = on　[**target =** *target*]
set batch_server nqs_budget_chk = off

ASV による予算管理を利用する場合、予算超過のチェックを実施するか否かを設定します。

on　　予算超過のチェックを実行します。
さらに、target = *target* の指定により、予算超過のチェックを行う対象の種類を指定できます。*target* には、以下の指定が可能です。

request
リクエストに対してのみ予算超過のチェックを行います。

reservation
予約区間に対してのみ予算超過のチェックを行います。

request,reservation
リクエストと予約区間の両方に対して予算超過のチェックを行います。
target の指定を行わない場合、この設定になります。

off　　予算超過のチェックを行いません。(既定値)

【アクセス権】管理者権限が必要です。

set batch_server get_resource_interval = *interval*

バッチサーバがジョブサーバからジョブの資源使用量を採取する間隔を *interval* で指定した値に設定します。単位は秒です。 0 に設定した場合、ジョブの資源使用量採取停止となります。

【アクセス権】操作員権限が必要です。

set batch_server group_submit_limit = *limit*

同一グループのリクエストの投入数の最大数を設定します。
バッチサーバ上に存在する同一グループのリクエスト数が*limit* で指定された数に達した場合、そのグループのリクエストを投入することができません。
limit には 0 から 1000000までの値を指定できます。リクエストの投入数の制限を行わない場合は、*limit* に0、または、"unlimited"を指定します。
既定値はunlimited(0)です。
グループ名を個別指定しての制限が設定されていないグループのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set batch_server group_submit_limit = *limit*　**groups** = *group*
set batch_server group_submit_limit = *limit*　**groups** = (*group*[, *group* ...])

group で指定したグループのリクエストの投入数の最大数を設定します。
group には、グループ名を指定します。
グループ *group* のリクエスト数が *limit* で指定された数に達した場合、そのグループ

のリクエストを投入することができません。
リクエストの投入数を無制限にしたい場合は、*limit* に 0、または、"unlimited"を指定します。

【アクセス権】操作員権限が必要です。

set batch_server heartbeat_interval = *interval*

ジョブサーバがバッチサーバに送信するハートビートの送信間隔を *interval* で指定した値に設定します。単位は秒です。 0 に設定した場合、ハートビートは送信停止となります。

【アクセス権】操作員権限が必要です。

set batch_server jacct_dir = *dir_name*

ASV による予算管理を利用する場合、バッチサーバホスト上で、ジョブアカウントファイルを出力するためのディレクトリを指定します。
既定値は、/var/opt/nec/nqsv/acm/jacct です。

【アクセス権】管理者権限が必要です。

set batch_server load_interval = *interval*

バッチサーバがジョブサーバから実行ホストの負荷情報やハードウェア資源搭載量情報を採取する間隔を *interval* で指定した値に設定します。単位は秒です。0 に設定した場合、実行ホストの負荷情報やハードウェア資源搭載量情報の採取が停止となります。

【アクセス権】操作員権限が必要です。

set batch_server log_file [**file** = *file_name*] [**size** = (*file_size*)] [**level** = *log_level*] [**save** = *save_num*]

NQSV ログファイルの設定をします。
file_name には、ログファイル名を指定します。
file_size には、ログファイルの上限サイズを指定します。指定できる値は 1024～2^31-1 バイトです。 サイズの指定には、単位の指定ができます。単位の指定がない場合はバイトとなります。0 は、無制限となります。 ログファイルがサイズの上限値に達するとそのファイルをバックアップファイルとして保存し、ログファイルのサイズが 0 にリセットされます。
log_level には、ログファイルの出力レベルを指定します。
save_num には、バックアップファイルの保存数を指定します。

ログファイルの出力メッセージには以下のレベルがあり、ログファイルには、

log_level に設定した数値以下のレベルのメッセージが全て出力されます。

Level 0　　DEBUG メッセージは出力されません。

Level 1　　バッチサーバの起動時/終了時の処理状況
スケジューラの接続状況
ジョブサーバの接続状況
バッチリクエストの処理状況
マスタージョブの終了状態
ネットワークリクエストの処理状況

Level 2　　ジョブの処理状況
NQSV/API クライアントの接続状況
全ジョブの終了状態
ユーザ EXIT スクリプトの標準/エラー出力
ステージングスクリプトの標準/エラー出力

Level 3　　ジョブ制御用パケット(task パケット)の処理状況

Level 4　　通信パケットの処理概要

Level 5　　通信パケットの内容表示

【アクセス権】操作員権限が必要です。

set batch_server network_queue retry_interval = *interval*

ネットワークキューが次に転送を試みるまでの待ち時間を設定します。指定できる値は秒単位です。

【アクセス権】操作員権限が必要です。

set batch_server network_queue retry_span = *span*

ネットワークキューがリトライ動作を繰り返す期間を設定します。*span* に指定できる値は 1～259200 秒(3 日)で、秒単位で指定します。初期設定値は、259200 秒です。

【アクセス権】操作員権限が必要です。

set batch_server network_queue run_limit = *limit*

limit で指定された値をシステム全体のネットワークキュー同時実行数として設定します。バッチサーバ上の全ネットワークキューの転送数の合計値が本コマンドで設定した

値になると、それ以上同時に転送は実行されません。

【アクセス権】操作員権限が必要です。

set batch_server req_account {on | off}

リクエストアカウント情報出力及びジョブアカウント情報出力を開始・停止します。

on　　出力開始

off　　出力停止

【アクセス権】管理者権限が必要です。

set batch_server req_account_file = *file_name*

file_name で指定したファイルをバッチサーバホスト上のリクエストアカウント情報の出力ファイルとします。*file_name* には、バッチサーバホスト上での絶対パス名を指定します。

【アクセス権】管理者権限が必要です。

set batch_server reservation_acct = {on | off}

ASV による予算管理を利用する場合、JobManipulator による事前予約（Resource Reservation）に対して課金を行うか、事前予約に対しては課金せず、予約内で投入リクエストごとに課金するかを設定します。

on　　事前予約（Resource Reservation）に対する課金

off　　予約内投入リクエストごとの課金

【アクセス権】管理者権限が必要です。

set batch_server routing_queue retry_interval = *interval*

転送キューが次に転送を試みるまでの待ち時間を設定します。指定できる値は秒単位です。

【アクセス権】操作員権限が必要です。

set batch_server routing_queue retry_span = *span*

転送キューがリトライ動作を繰り返す期間を設定します。*span* に指定できる値は 1～259200 秒(3 日)で、秒単位で指定します。初期設定値は、259200 秒です。

【アクセス権】操作員権限が必要です。

set batch_server routing_queue run_limit = *limit*

limit で指定された値をシステム全体の転送キュー同時実行数として設定します。バッチサーバ上の全転送キューの転送数の合計値が本コマンドで設定した値になると、それ以上同時に転送は実行されません。

【アクセス権】操作員権限が必要です。

set batch_server rsvacct_file = *path_name*

ASV による予算管理を利用する場合、JobManipulator による事前予約に対するアカウントファイルのパス名を *path_name* に設定します。

【アクセス権】管理者権限が必要です。

set batch_server specify_group_request { **on** | **off** }

リクエストのグループ指定実行機能を使用するか否かを設定します。

on　　リクエストのグループ指定実行機能を使用します。リクエスト投入時に補助グループが指定でき、そのグループでジョブを実行します。

off　　リクエストのグループ指定実行機能を使用しません。

【アクセス権】操作員権限が必要です。

set batch_server submit_limit = *limit*

バッチサーバに投入できるリクエストの数を設定します。投入できるリクエスト数とは、バッチサーバ上に同時に存在しているリクエストの総数です。

limit で指定された数を超えてリクエストを投入することはできません。

limit には 1 から 1000000 までの値を指定することができます。 初期設定値は 1000 です。

【アクセス権】操作員権限が必要です。

set batch_server subrequest_entry_limit = *limit*

バッチサーバに投入されたパラメトリックリクエストに対して、パラメトリックリクエスト毎にバッチサーバ内で同時に生成するサブリクエストの数（同時生成サブリクエスト数）を *limit* で指定した数に制限します。初期設定値は 100 です。

【アクセス権】操作員権限が必要です。

set batch_server user_submit_limit = *limit*

同一ユーザが投入できるリクエストの最大数を設定します。投入できるリクエスト数とは、バッチサーバ上に同時に存在しているリクエストの総数です。

バッチサーバ上に存在する同一ユーザのリクエスト数が *limit* で指定された数に達した場合、そのユーザはそれ以上リクエストを投入することができません。
limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。
既定値は unlimited(0)です。
ユーザ名を個別指定しての制限が設定されていないユーザのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set batch_server user_submit_limit = *limit*　**users** = *user*
set batch_server user_submit_limit = *limit*　**users** = (*user*[, *user* …])
user で指定したユーザのリクエストの投入数の最大数を設定します。
user には、ユーザ名を指定します。
ユーザ *user* のリクエスト数が *limit* で指定された数に達した場合、そのユーザのリクエストを投入することができません。
リクエストの投入数を無制限にしたい場合は、*limit* に0、または、"unlimited"を指定します。

【アクセス権】操作員権限が必要です。

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*
template_name で指定した名前のクラウドバースティング用テンプレート情報の内容を変更します。
但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、変更できません。

テンプレート情報として、以下を指定します。

image	起動するクラウドインスタンスのイメージ名。
cpu	割り当てる CPU 数。 １以上の整数を指定します。
memsz	割り当てるメモリサイズ。

1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。

gpu　割り当てる GPU 数。
0 以上の整数を指定します。

ve　割り当てる VE 数。
0 以上の整数を指定します。

boot_timeout　クラウドインスタンスの起動見込時間。
秒単位で、1～2147483647 の整数で指定します。
既定値は 300 秒です。

stop_timeout　クラウドインスタンスの停止見込時間。
秒単位で、1～2147483647 の整数で指定します。
既定値は 300 秒です。

custom　起動環境として独自に定義した情報。

comment　テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

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*

template_name で指定した名前のコンテナ用テンプレート情報の内容を変更します。
但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、変更できません。

テンプレート情報として、以下を指定します。

image　起動するコンテナのイメージ名。

cpu　割り当てる CPU 数。
１以上の整数を指定します。

memsz　割り当てるメモリサイズ。
1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。

gpu　割り当てる GPU 数。
0 以上の整数を指定します。

ve	割り当てる VE 数。 0 以上の整数を指定します。
hca	割り当てる HCA ポート数。 *hcanum* は、<io>,<mpi>,<all>の形式で指定します。 <io>,<mpi>,<all>には 0 以上の整数を指定します。
boot_timeout	起動見込時間。 秒単位で、1～2147483647 の整数で指定します。 既定値は 900 秒です。
stop_timeout	停止見込時間。 秒単位で、1～2147483647 の整数で指定します。 既定値は 900 秒です。
custom	起動環境として独自に定義した情報。
comment	テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

set custom_resource consumer = { **job** | **request** }　*cr_name*

cr_name で指定したカスタムリソース情報について、カスタムリソースを消費する単位を job、 request のいずれかに設定変更します。
但し、job から request に変更する場合、既に host タイプの利用量制御情報が登録されている時は変更できません。その場合は先に host タイプの利用量制御情報を削除してください。

【アクセス権】管理者権限が必要です。

set custom_resource check_mode = { **off** | **moment** | **integrate** }　*cr_name*

cr_name で指定したカスタムリソース情報について、このカスタムリソースを使用した資源監視及び使用量の出力について、off、moment、integrate のいずれかに設定変更します。

off	資源監視及び使用量の出力を行いません。
moment	資源監視及び使用量の出力を行います。使用量は「瞬間値」として扱います。
integrate	資源監視及び使用量の出力を行います。使用量は「積算値」として扱います。

※「瞬間値」「積算値」はアカウンティング編を参照してください。
但し、off に変更する場合、既に terminate_job に on が設定されている時は変更できません。その場合は、先に terminate_job に off を設定してください。

【アクセス権】管理者権限が必要です。

set custom_resource terminate_job = { **off** | **on** }　*cr_name*

cr_name で指定したカスタムリソース情報について、このカスタムリソースを使用して、資源超過時のジョブ終了機能について、off、on のいずれかに設定変更します。

off　　資源超過時にジョブを終了させません。

on　　資源超過時にジョブを終了させます。

但し、on に変更する場合、既に check_mode に off が設定されている時は変更できません。その場合は、先に check_mode に off 以外を設定してください。

【アクセス権】管理者権限が必要です。

set custom_resource unit = *unit*　*cr_name*

cr_name で指定したカスタムリソース情報について、このカスタムリソースの単位を設定変更します。単位は 5 文字までの文字列で、5 文字を超えた部分は切り捨てられます。*unit* に unspecified を指定した場合は、既に設定している単位を削除します（無名数として扱います)。

【アクセス権】管理者権限が必要です。

set execution_queue access *queue*

queue で指定されたバッチキューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを許可します。 モードを変更した場合、root 以外のそれまでのユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set execution_queue auto_bind_jobserver { **on** | **off** } *queue*

queue で指定したバッチキューに対して、ジョブサーバ起動時に、それまでのキューとの接続関係を保持するか、自動的に接続関係の切り離しを実施するかを設定します。**on** の場合は、ジョブサーバ起動時に、ジョブサーバのキューとの接続関係を保持します。**off** の場合は、ジョブサーバ起動時に接続関係を切り離します。

【アクセス権】操作員権限が必要です。

set execution_queue cpunum_per_gpunum = *cpunum queue*

queue で指定された GPU-CPU Affinity 機能が有効なバッチキューに投入されたリクエストに対して GPU あたりの CPU コア数の既定値を設定します。

cpunum には 1～2147483647 の整数が指定できます。既定値は 1 です。

cpunum に指定する値は、以下の式で算出する値です。

1 ソケットの CPU コア数 ÷ 当該ソケットに物理的に近い GPU 台数

【アクセス権】操作員権限が必要です。

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*

queue で指定されたキューに対して、*cr_name* で指定した名前のカスタムリソース利用量制限情報の設定をします。

standtard で、リクエスト投入時に指定するカスタムリソースの利用量の既定値を設定します。*std* には 1～2147483647 の整数、または unused (0) が指定できます。

range で、リクエスト投入時に指定するカスタムリソースの利用量の指定範囲（下限値,上限値）を設定します。*min*、*max* には、1～2147483647 の整数が指定できます。

permit_unused で、リクエスト投入時のカスタムリソースの利用量制御対象外指定（unused (0)）を許可するか否かを設定します。

yes	カスタムリソースの利用量制御対象外とする指定（unused (0)）を許可します。(既定値)
no	カスタムリソースの利用量制御対象外とする指定は不可です。

【アクセス権】操作員権限が必要です。

set execution_queue defined_ve_number = *venum*　*queue*

queue で指定したキューに対して、VEノード総数指定によるリクエスト投入を行う場合の、ジョブ数の自動換算に使用する既定搭載VEノード数を設定します。
venum には 1 から 256 までの値を指定できます。既定値は 1 です。

【アクセス権】操作員権限が必要です。

set execution_queue delete_failed_urgent_request = { **on** | **off** }　*queue*

queue で設定されたキューに投入された緊急リクエストが実行不可である場合にその緊急リクエストを削除する機能を設定します。
onの場合は、実行できなかった緊急リクエストを削除します。
offの場合は、実行できなかった緊急リクエストを削除しません。この時、割り込まれた通常リクエストはリランされます。
既定値は off です。

【アクセス権】操作員権限が必要です。

set execution_queue exclusive_submit { **on** | **off** } *queue*

queue で指定したバッチキューに対する排他実行リクエストの投入可否を設定します。

on の場合は、排他実行リクエストの投入を許可します。off の場合は排他実行リクエストの投入を拒否します。

【アクセス権】操作員権限が必要です。

set execution_queue file_stageout = { **on** | **off** }　*<queue>*

*queue*で指定したバッチキューに投入されたリクエストにおいて、ステージアウトを実施するかどうかを設定します。

onの場合は、ステージアウトを実施します。offの場合はステージアウトを抑止します。

【アクセス権】操作員権限が必要です。

set execution_queue gpu_affinity = { **on** | **off** }　*<queue>*

*queue*で指定したバッチキューに投入されたリクエストにおいて、GPU-CPU Affinity機能を使用するかどうかを設定します。

onの場合は、GPU-CPU Affinity機能を使用します。offの場合はGPU-CPU Affinity機能を使用しません。GPU-CPU Affinity機能を使用する場合は、ソケットスケジューリング機能を有効にする必要があります。既定値はoffです。

【アクセス権】操作員権限が必要です。

set execution_queue group_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一グループのリクエストの投入数の最大数を設定します。

キュー*queue* 内に同時に存在する同一グループのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。

既定値は unlimited(0)です。

グループ名を個別指定しての制限が設定されていないグループのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set execution_queue group_submit_limit = *limit*　**groups** = *group*　*queue*

set execution_queue group_submit_limit = *limit*　**groups** = (*group*[, *group* ...])　*queue*

queue で指定したキューに対して、*group*で指定したグループのリクエストの投入数の最大数を設定します。
group には、グループ名を指定します。
キュー*queue* 内に同時に存在するグループ*group* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。
limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set execution_queue groups = *group queue*
set execution_queue groups = (*group* [, *group* ...]) *queue*

queue で指定されたバッチキューのグループリストに *group* で指定されたグループを設定します。それまでのグループリストの設定は削除されます。グループリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

set execution_queue hca_number_range = (*min*, *max*) mode=*hca-mode queue*

queue で指定されたバッチキューに *hca-mode* で指定された種別の HCA のポート数制限として（下限値, 上限値）を設定します。*min*, *max* には、0 以上 32 以下の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。*min* と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。
hca-mode には下記のいずれかを指定できます。

io	SCATEFS ダイレクト IO 用の HCA です。
mpi	MPI 用の HCA です。
all	SCATEFS ダイレクト IO 用と MPI 用の両方の HCA です。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定する HCA のポート数がこの設定範囲に制限されます。
既定値は（0, unlimited）です。HCA 種別ごとの制限値は独立しており、お互いの設定には干渉しません。

【アクセス権】操作員権限が必要です。

set execution_queue hold_privilege = *privilege queue*

queue で指定されたバッチキューに対して、ホールド要求が可能な権限 *privilege* を設定します。この設定により、*queue* で指定されたバッチキューに投入されたリクエストをホールドできるのは、*privilege* で設定した権限以上の権限を持つユーザに制限されます。*privilege* には以下の権限が指定できます。

user	一般利用者権限
specialuser	特別利用者権限
groupmanager	グループ管理者権限
operator	操作員権限
manager	管理者権限
scheduler	スケジューラ権限

【アクセス権】操作員権限が必要です。

set execution_queue hook_function { **on** | **off** } *queue*

queue に指定されたバッチキューで、NQSV のフックスクリプト機能を使用するかどうかの設定を行います。

on	フックスクリプト機能を使用します。 /opt/nec/nqsv/sbin/hook_prog 下に配置されているフックスクリプトを実行します。
off	フックスクリプト機能を使用しません。（既定値）

【アクセス権】操作員権限が必要です。

set execution_queue intmpi_process_manager = hydra *queue*

queue で指定したバッチキューに対して、IntelMPI のリクエストが使用するプロセスマネージャを、**hydra** に設定します。

既定値は **hydra** です。

【アクセス権】操作員権限が必要です。

set execution_queue jobs_range = (*jobs_min* , *jobs_max*）　*queue*

queue で指定されたキューに対して、リクエストごとのジョブ数の上限値／下限値を設定します。

この設定により、*queue* で指定されたキューに投入されるリクエストのジョブ数は *jobs_min* と *jobs_max* で指定した数の範囲に限定されます。*jobs_min*、*jobs_max* に指定できる数は、1～10240 の整数値です。

queue で指定されたキューにリクエストを投入する際、リクエストに指定されたジョブ数が、範囲外の時は投入エラーとなります。

【アクセス権】操作員権限が必要です。

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*

queue で指定されたキューに対して、*group* または *user* で指定した、グループまたはユーザのリクエストごとのジョブ数の上限値／下限値を設定します。
group には、グループ名を指定します。*user* には、ユーザ名を指定します。
ジョブ数は*jobs_min* と*jobs_max* で指定した数の範囲に限定されます。*jobs_min*、*jobs_max* に指定できる数は、1～10240の整数値です。
queue で指定されたキューにリクエストを投入する際、リクエストのグループ・ユーザが、*group* または *user* である場合、リクエストに設定されたジョブ数が、ここで設定された範囲外の時、投入エラーとなります。

【アクセス権】操作員権限が必要です。

set execution_queue kernel_param nice = *value queue*

queue で指定されたバッチキューに、nice 値 *value* を設定します。この設定により、*queue*で指定されたバッチキューに投入されたリクエストが実行する際に使用する nice 値として *value* が設定されます。

【アクセス権】操作員権限が必要です。

set execution_queue kernel_param rsg_number = *value queue*

queue で指定されたバッチキューに、リソースシェアリンググループ番号（RSG 番号）*value* を設定します。この設定により、*queue* で指定されたバッチキューに投入されたリクエストが実行する際に使用するリソースシェアリンググループ番号（RSG 番号）として *value* が設定されます。

【アクセス権】操作員権限が必要です。

set execution_queue lhost cpu_number_range = (*min, max*）　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 台数制限に（下限値, 上限値）を設定します。*min, max* には、1 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。
この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の CPU 台数が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

既定値は（1, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost cpu_time_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 時間制限に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の CPU 時間が、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost gpu_number_range = (*min, max*）　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の GPU 台数制限に（下限値, 上限値）を設定します。*min, max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の GPU 台数が、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost memory_size_range = (*min, max*, [*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）のメモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）のメモリサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost stderr_size_range = (*min*, *max* [,*warn*]) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の標準エラー出力の結果ファイルサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の標準エラー出力の結果ファイルサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost stdout_size_range = (*min*, *max* [,*warn*]) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の標準出力の結果ファイルサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の標準出力の結果ファイルサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost ve_number_range = (*min, max*）　*queue*

queue で指定されたキューの論理ホスト毎の VE ノード数制限に（下限値, 上限値）を設定します。*min*, *max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎の VE ノード数が、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost vecpu_time_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE の CPU 時間制限に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の VE の CPU 時間が、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost vememory_size_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の VE のメモリサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue lhost virtual_memory_size_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の仮想メモリサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue necmpi_process_manager = { **hydra** | **mpd** } *queue*

queue で指定したバッチキューに対して、NEC MPI のリクエストが使用するプロセスマネージャを、hydra または mpd に設定します。
既定値は mpd です。

【アクセス権】操作員権限が必要です。

set execution_queue noaccess *queue*

queue で指定されたバッチキューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを禁止します。 モードを変更した場合、root 以外のそれまでのユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set execution_queue numa_control = { **on** | **off** }　*queue*

queue に指定されたキューに対して、NUMA アーキテクチャの実行ホストに対するソケットスケジューリング機能の ON/OFF を設定します。
この設定を **on** にセットしたキューに対しては、NUMA アーキテクチャのスカラーマシン（Linux）の実行ホスト上で起動したジョブサーバをバインドする必要があります。また、すべて同じソケット構成の実行ホストとする必要があります。

on	ソケットスケジューリング機能を利用します
off	ソケットスケジューリング機能を利用しません　（既定値）

【アクセス権】操作員権限が必要です。

set execution_queue numa_option core_bind_policy = *policy*　*queue*

queue に指定されたキューに対して、*policy* で指定されたコアバインドポリシーを設定します。*policy* には以下を指定することができます。

concentration	ソケット集中ポリシー　（既定値） 空きコアがより少ないソケットを優先して割り当てます。
balance	ソケット分散ポリシー 空きコアがより多いソケットを優先して割り当てます。

【アクセス権】操作員権限が必要です。

set execution_queue numa_option memory_allocation_policy = *policy*　*queue*

queue に指定されたキューに対して、*policy* で指定されたメモリアロケーションポリシーを設定します。*policy* には以下を指定することができます。

membind	membind ポリシー ジョブの実行コアが属するソケットのメモリのみを使用します。メモリが不足した場合はスワップを使用します。
localalloc	localalloc ポリシー　（既定値） ジョブの実行コアが属するソケットのメモリを優先的に使用します。 モリが不足した場合は他ジョブのソケット上のメモリを使用します。
interleave	interleave ポリシー ジョブに割り当てたソケットのメモリを交互に使用します。メモリが不足した場合は他ジョブのソケット上のメモリを使用します。

【アクセス権】操作員権限が必要です。

set execution_queue numa_unit_check = {**on** | **off**}　*queue*

queue に指定されたキューに対して、リクエスト投入時に指定したジョブあたりの CPU 数とメモリ使用量の制限値が、実行ホストのソケットあたりの搭載 CPU(コア)数とメモリ容量の比率に一致するかのチェックの ON/OFF を設定します。
この設定を ON に設定したキューでは、投入したリクエストが、この比率に一致しない場合は、投入エラーとなります。

【アクセス権】操作員権限が必要です。

set execution_queue partial_process_swapping = {**on** | **off**}　*queue*

queue で指定されたキューに投入されたリクエストをサスペンドするときに、VEOS の Partial Process Swapping 機能の利用可否を設定します。
on の場合は、緊急リクエスト実行時に Partial Process Swapping を利用して通常リクエストをサスペンドし、緊急リクエストを実行します。
off の場合は、緊急リクエスト実行時に Partial Process Swapping を利用せずに通常リクエストをサスペンドし、緊急リクエストを実行します。
既定値は on です。

【アクセス権】操作員権限が必要です。

set execution_queue per_job cpu_number_limit = *limit queue*

queue で指定されたバッチキューのジョブ単位の CPU 台数制限に *limit* を設定します。この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するジョブ単位の CPU 台数が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で CPU 台数が制限できるのは、実行ホストがジョブ単位の CPU 台数制限機能をサポートしている場合のみです。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_job cpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位の CPU 時間制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するジョブ単位の CPU 時間制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で CPU 時間が制限できるのは、実行ホストがジョブごとの CPU 時間制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_job gpu_number_limit = *limit*　*queue*

queue で指定されたキューのジョブ単位の GPU 台数制限に *limit* を設定します。この設定により、*queue* で指定されたキューに投入するリクエストに指定するジョブ単位の GPU 台数が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で GPU 台数が制限できるのは、実行ホストが GPU をサポートしている場合のみです。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_job memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位のメモリサイズ制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するジョブ単位のメモリサイズ制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上でメモリサイズ制限ができるのは、実行ホストがジョブごとのメモリサイズ制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_job virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位の仮想メモリサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するジョブ単位の仮想メモリサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがジョブごとの仮想メモリサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

仮想メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc core_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のコアファイルサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位のコアファイルサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でコアファイルサイズ制限ができるのは、実行ホストがプロセスごとのコアファイルサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

コアファイルサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc cpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の CPU 時間制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位の CPU 時間制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で CPU 時間制限ができるのは、実行ホストがプロセスごとの CPU 時間制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

CPU 時間制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc data_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のデータサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位のデータサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でデータサイズ制限ができるのは、実行ホストがプロセスごとのデータサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

データサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc file_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のファイルサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位のファイルサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でファイルサイズ制限ができるのは、実行ホストがプロセスごとのファイルサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

ファイルサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc open_file_number_limit = *limit queue*

queue で指定されたバッチキューのプロセス単位の同時オープンファイル数制限値に *limit* を設定します。

この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位の同時オープンファイル数制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で同時オープンファイル数制限ができるのは、実行ホストがプロセスごとの同時オープンファイル数制限機能をサポートしている場合のみです。

同時オープンファイル数制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc stack_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のスタックサイズ制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位のスタックサイズ制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上でスタックサイズ制限ができるのは、実行ホストがプロセスごとのスタックサイズ制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
スタックサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc vecpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の VE CPU 時間制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセスの VE CPU 時間制限値が *limit* に制限されます。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
VE CPU 時間制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc vememory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の VE メモリサイズ制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位の VE メモリサイズ制限値が *limit* に制限されます。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
VE メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_prc virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の仮想メモリサイズ制限値に *limit* を設定します。
この設定により、*queue* で指定されたバッチキューに投入するリクエストに指定するプロセス単位の仮想メモリサイズ制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホ

ストがプロセスごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
仮想メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue per_req elapse_time_limit = (*limit*)　*queue*

queue で指定されたキューのリクエスト単位の経過時間制限値として*limit* を設定します。
queue で指定されたキューにリクエストを投入する際、リクエストに設定された制限値とここで設定された *limit* を比較し、*limit* を超える場合には投入エラーとなります。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

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*

キュー*queue* に対して、*group* または*user* で指定した、グループまたはユーザのリクエスト単位の経過時間制限値として *limit* を設定します。
*group*には、グループ名を指定します。*user* には、ユーザ名を指定します。
queue で指定されたキューにリクエストを投入する際に、リクエストのグループ・ユーザが、*group* または *user* である場合、リクエストに設定された制限値とここで設定された*limit* を比較し、*limit* を超える時は投入エラーとなります。

【アクセス権】操作員権限が必要です。

set execution_queue priority = *priority*　*queue*

queue で指定されたバッチキューのキュープライオリティを *priority* に設定します。

【アクセス権】操作員権限が必要です。

set execution_queue qattach = *mode*　*queue*

queue で指定されたバッチキューに投入されたリクエストに対して、**qattach** によるア

タッチを許可するか否かを *mode* で設定します。アタッチを許可する場合は、*mode* に"enable"を指定し、許可しない場合は、"disable"を指定します。
既定値は、"enable"です。

【アクセス権】管理者権限が必要です。

set execution_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定されたバッチキューへのリクエスト投入に対して、*from* で指定した投入経路からのリクエスト投入を拒否します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit	qsub からのリクエスト投入
qmove	qmove からのリクエスト投入
local_routing	ローカルからのリクエスト転送
remote_routing	リモートからのリクエスト転送

【アクセス権】操作員権限が必要です。

set execution_queue request_priority_range *privilege* = (*min* , *max*) *queue*

queue で指定されたバッチキューへのリクエスト投入時に指定可能なリクエストプライオリティの上限/下限値をユーザ権限毎(*privilege*)に設定します。*min* , *max* に指定できるリクエストプライオリティの上限/下限値は、-1024～1023 の範囲の整数値です。
privilege には、各ユーザ権限に対して以下の文字列が指定できます。

manager	管理者権限
operator	操作員権限
specialuser	特別利用者権限
user	一般利用者権限

【アクセス権】管理者権限が必要です。

set execution_queue rerun_default = *mode* [**limit** = *n*] *queue*

queue で指定されたバッチキューに投入されたリクエストでリラン可否の指定が明示的に行われなかった場合に、リクエストに適用されるリラン可否の既定値を *mode* で指定します。リランを許可する場合は、*mode* に yes を指定し、許可しない場合は、no を指定します。
mode が yes の場合は **limit** でリラン回数の上限を設定することができます。*n* には 1 から 2147483647 の整数を指定できます。**limit** を省略すると、リラン回数は無制限です。
mode が no の場合に limit を指定しても、何の効力もありません。

既定値は yes で、リラン回数は無制限です。

【アクセス権】操作員権限が必要です。

set execution_queue standard hca_number = *limit* mode=*hca-mode queue*

queue で指定されたバッチキューの *hca-mode* で指定された種別の HCA のポート数制限値の既定値として *limit* を設定します。*limit* には 0 以上 32 以下の整数または unlimited が指定できます。

queue で指定されたバッチキューにリクエストを投入する際に、HCA のポート数制限値が指定されていない場合、ここで設定された *limit* が HCA のポート数制限値として使用されます。

hca-mode には下記のいずれかを指定できます。

io	SCATEFS ダイレクト IO 用の HCA です。
mpi	MPI 用の HCA です。
all	SCATEFS ダイレクト IO 用と MPI 用の両方の HCA です。

既定値は 0 です。HCA 種別ごとの制限値は独立しており、お互いの設定には干渉しません。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost cpu_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の CPU 台数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の CPU 台数の制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost cpu_time_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の CPU 時間制限が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の CPU 時間制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】管理者権限が必要です。

set execution_queue standard lhost gpu_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の GPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の GPU 台数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の GPU 台数の制限値として使用されます。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost memory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）のメモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）のメモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost stderr_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の標準エラー出力の結果ファイルサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の標準エラー出力の結果ファイルサイズが指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の標準エラー出力の結果ファイルサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost stdout_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の標準出力の結果ファイルサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の標準出力の結果ファイルサイズが指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の標準出力の結果ファイルサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost ve_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎の VE ノード数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎の VE ノード数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎の VE ノード数の制限値として使用されます。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost vecpu_time_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の VE の CPU 時間制限が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の VE の CPU 時間制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost vememory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard lhost virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_job cpu_number_limit = *limit queue*

queue で指定されたバッチキューのジョブ単位の CPU 台数制限の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、ジョブ単位の CPU 台数が指定されていない場合は、ここで設定された *limit* がジョブ単位の CPU 台数の制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で CPU 台数が制限できるのは、実行ホストがジョブ単位の CPU 台数制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_job cpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、ジョブ単位の CPU 時間制限が指定されていない場合、ここで設定された *limit* がジョブ単位の CPU 時間制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で CPU 時間が制限できるのは、実行ホストがジョブごとの CPU 時間制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_job gpu_number_limit = *limit queue*

queue で指定されたキューのジョブ単位の GPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、ジョブ単位の GPU 台数が指定されていない場合は、ここで設定された *limit* がジョブ単位の GPU 台数の制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で GPU 台数が制限できるのは、実行ホストがジョブ単位の GPU 台数制限機能をサポートしている場合のみです。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_job memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位のメモリサイズ制限値の既定値として

limit を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、ジョブ単位のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* がジョブ単位のメモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でメモリサイズ制限ができるのは、実行ホストがジョブごとのメモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_job virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのジョブ単位の仮想メモリサイズ制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、ジョブ単位の仮想メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がジョブ単位の仮想メモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがジョブごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc core_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のコアファイルサイズ制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位のコアファイルサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のコアファイルサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でコアファイルサイズ制限ができるのは、実行ホストがプロセスごとのコアファイルサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc cpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の CPU 時間制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位の CPU 時間制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の CPU 時間制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で CPU 時間制限ができるのは、実行ホストがプロセスごとの CPU 時間制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc data_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のデータサイズ制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位のデータサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のデータサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でデータサイズ制限ができるのは、実行ホストがプロセスごとのデータサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc file_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のファイルサイズ制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位のファイルサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のファイルサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でファイルサイズ制限ができるのは、実行ホストがプロセスごとのファイルサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc open_file_number_limit = *limit queue*

queue で指定されたバッチキューのプロセス単位の同時オープンファイル数制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位の同時オープンファイル数制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の同時オープンファイル数制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で同時オープンファイル数制限ができるのは、実行ホストがプロセスごとの同時オープンファイル数制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc stack_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位のスタックサイズ制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位のスタックサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のスタックサイズ制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上でスタックサイズ制限ができるのは、実行ホストがプロセスごとのスタックサイズ制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc vecpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の VE CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位の VE CPU 時間制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の VE CPU 時間制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc vememory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の VE メモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位の VE メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の VE メモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_prc virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューのプロセス単位の仮想メモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、プロセス単位の仮想メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の

仮想メモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがプロセスごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard per_req elapse_time_limit = (*limit*) *queue*

queue で指定されたバッチキューのリクエスト単位の経過時間制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、リクエスト単位の経過時間制限値が指定されていない場合、ここで設定された *limit* がリクエスト単位の経過時間制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard venode vecpu_time_limit = (*limit*) *queue*

queue で指定されたバッチキューの VE ノード単位の VE の CPU 時間制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、VE ノード単位の VE の CPU 時間制限値が指定されていない場合、ここで設定された *limit* が VE ノード単位の VE の CPU 時間制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue standard venode vememory_size_limit = (*limit*) *queue*

queue で指定されたバッチキューの VE ノード単位の VE のメモリサイズ制限値の既定値として *limit* を設定します。
queue で指定されたバッチキューにリクエストを投入する際に、VE ノード単位の VE のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* が VE ノード単位の VE のメモリサイズ制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue submit_cpu_unit = { **cpu** | **socket** | **any** }　*queue*

queue で指定されたキューに対して、リクエスト投入時の、ジョブあたりの CPU 数の指定の方法を設定します。

cpu	cpunum_job のみ指定可能
socket	socknum_job のみ指定可能
any	cpunum_job、socknum_job のどちらでも指定可能（既定値）

【アクセス権】操作員権限が必要です。

set execution_queue submit_limit = *limit queue*

queue で指定されたバッチキューに投入できるリクエストの最大数として *limit* に指定した数を設定します。投入できるリクエスト数とは、1 つのキュー内に同時に存在しているリクエストの総数です。 *limit* で指定した数を超えたリクエストの投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。0 は無制限を表します。

既定値は 0（無制限）です。

【アクセス権】操作員権限が必要です。

set execution_queue submit_venode_range = (*min, max*) *queue*

queue で指定されたキューに VE ノード総数指定（qsub --venode）で投入できる VE ノード総数の範囲（上限値と下限値）を設定します。*min, max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。

min よりも小さい、または、*max* よりも大きい値で VE ノード総数を指定したリクエストの投入はエラーとなります。

min と *max* が同じ値であった場合、VE ノード総数としてその値しか指定できなくなります。*min* と *max* がともに 0 の場合、VE ノードを使用しないリクエストのみ投入できます。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set execution_queue subrequest_limit = *limit queue*

queue で指定されたバッチキューにパラメトリックリクエストを投入する際に指定可能なサブリクエスト数の最大値として *limit* に指定した数を設定します。*limit* で指定した数を超える数のサブリクエストを生成するパラメトリックリクエストの投入はエラーとなります。

【アクセス権】操作員権限が必要です。

set execution_queue supplementary_groups_check { **on** | **off** } *queue*

queue で指定されたバッチキューへリクエスト投入する際に行う、グループに対するアクセス権チェックにおいて、補助グループ ID を使用するか否かを設定します。**on** の場

合はチェックに補助グループ ID を使用し、**off** の場合は使用しません。

【アクセス権】操作員権限が必要です。

set execution_queue suspend_privilege = *privilege queue*

queue で指定されたバッチキューに対して、サスペンド要求が可能な権限 *privilege* を設定します。この設定により、*queue* で指定されたバッチキューに投入されたリクエストをサスペンドできるのは、*privilege* で設定した権限以上の権限を持つユーザに制限されます。*privilege* には以下の権限が指定できます。

user	一般利用者権限
specialuser	特別利用者権限
groupmanager	グループ管理者権限
operator	操作員権限
manager	管理者権限
scheduler	スケジューラ権限

【アクセス権】操作員権限が必要です。

set execution_queue userexit location = *location*　**script** = (*script*) **queue** = *queue*

queue で指定されたバッチキューに投入されたリクエストに対して、*location* で指定されたタイミングで実行する実行するユーザ EXIT のプログラムとして *script* を設定します。*location* に指定できるタイミングは以下のとおりです。

pre-running	ジョブ実行開始前
post-running	ジョブ実行終了直後

また、*script* には、実行するスクリプトのファイル名を指定します。ファイル名にパス要素 ('/') を含めることはできません。指定されたファイルはバッチサーバホスト上の /opt/nec/nqsv/sbin/uex_prog 直下に存在する必要があります。
script には、","(カンマ)または、":"(コロン)で区切って、ファイルを 4 つまで指定できます。ファイルを","(カンマ)で区切った場合、前後のファイルを直列に、":"(コロン)で区切って指定した場合は、前後のファイルを並列に実行します。
ファイル名を指定する"("、")"及び区切り文字の","（カンマ）、":"（コロン）の前後の空白文字は自動的に削除されます。空白文字を含むファイル名は指定できません。

【アクセス権】操作員権限が必要です。

set execution_queue userexit_timeout = *time queue*

queue で指定したバッチキューのユーザ Exit のタイムアウト時間を *time* で指定した時

間（単位：秒）に設定します。0 を指定した場合、機能は無効です。既定値は無効（0）です。

【アクセス権】操作員権限が必要です。

set execution_queue userpp_timeout = *time queue*

queue で指定したバッチキューのユーザ PP スクリプトのタイムアウト時間を *time* で指定した時間（単位：秒）に設定します。0 を指定した場合、機能は無効です。既定値は 300 秒です。

【アクセス権】操作員権限が必要です。

set execution_queue user_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一ユーザのリクエストの投入数の最大数を設定します。

キュー*queue* 内に同時に存在する同一ユーザのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。

既定値は unlimited(0)です。

ユーザ名を個別しての制限が設定されていないユーザのリクエストには、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set execution_queue user_submit_limit = *limit*　**users** = *user*　*queue*

set execution_queue user_submit_limit = *limit*　**users** = (*user* [,*user*...])　*queue*

queue で指定したキューに対して、*user* で指定したユーザのリクエストの投入数の最大数を設定します。

user には、ユーザ名を指定します。

キュー*queue* 内に同時に存在するユーザ*user* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set execution_queue users = *user queue*

set execution_queue users = (*user* [, *user* ...]) *queue*

queue で指定されたバッチキューのユーザリストに *user* で指定されたユーザを設定し

ます。それまでのユーザリストの設定は削除されます。ユーザリストのモードは、**set execution_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、"user1"か"[1001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

set execution_queue venode vecpu_time_range = (min, max [,warn]) queue

queue で指定されたバッチキューの VE ノード毎の VE の CPU 時間制限に（下限値,上限値）を設定します。min は max 以下の値を指定してください。この設定により、queue で指定されたキューに投入するリクエストに指定する VE ノード毎の VE の CPU 時間が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。 *warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は(0,unlimited)です。

【アクセス権】操作員権限が必要です。

set execution_queue venode vememory_size_range = (min, max [,warn]) queue

queue で指定されたバッチキューの VE ノード毎の VE のメモリサイズ制限値に（下限値, 上限値）を設定します。min は max 以下の値を指定してください。この設定により、queue で指定されたキューに投入するリクエストに指定する VE ノード毎の VE のメモリサイズが、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。 *warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は(0,unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue access *queue*

queue で指定された会話キューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを許可します。 モードを変更した場合、root 以

外のそれまでのユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set interactive_queue auto_bind_jobserver { **on** | **off** } *queue*

queue で指定した会話キューに対して、ジョブサーバ起動時に、それまでのキューとの接続関係を保持するか、自動的に接続関係の切り離しを実施するかを設定します。**on** の場合は、ジョブサーバ起動時に、ジョブサーバのキューとの接続関係を保持します。**off** の場合は、ジョブサーバ起動時に接続関係を切り離します。

【アクセス権】操作員権限が必要です。

set interactive_queue cpunum_per_gpunum = *cpunum queue*

*queue*で指定された GPU-CPU Affinity 機能が有効な会話キューに投入されたリクエストに対して、GPU あたりの CPU コア数の既定値を設定します。

cpunum には 1～2147483647 の整数が指定できます。既定値は 1 です。

cpunum に指定する値は、以下の式で算出する値です。

1 ソケットの CPU コア数 ÷ 当該ソケットに物理的に近い GPU 台数

【アクセス権】操作員権限が必要です。

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** } *queue*

queue で指定されたキューに対して、*cr_name* で指定した名前のカスタムリソース利用量制限情報の設定をします。

standtard で、リクエスト投入時に指定するカスタムリソースの利用量の既定値を設定します。*std* には 1～2147483647 の整数、または unused (0) が指定できます。

range で、リクエスト投入時に指定するカスタムリソースの利用量の指定範囲（下限値,上限値）を設定します。*min*、*max* には、1～2147483647 の整数が指定できます。

permit_unused で、リクエスト投入時のカスタムリソースの利用量制御対象外指定（unused (0)）を許可するか否かを設定します。

yes	カスタムリソースの利用量制御対象外とする指定（unused (0)）を許可します。（既定値）
no	カスタムリソースの利用量制御対象外とする指定は不可です。

【アクセス権】操作員権限が必要です。

set interactive_queue defined_ve_number = *venum*　*queue*

queue で指定したキューに対して、VEノード総数指定によるリクエスト投入を行う場合の、ジョブ数の自動換算に使用する既定搭載VEノード数を設定します。

venum には 1 から 256 までの値を指定できます。既定値は 1 です。

【アクセス権】操作員権限が必要です。

set interactive_queue exclusive_submit {on | off} *queue*

queue で指定した会話キューに対する排他実行リクエストの投入可否を設定します。

on の場合は、排他実行リクエストの投入を許可します。off の場合は排他実行リクエストの投入を拒否します。

【アクセス権】操作員権限が必要です。

set interactive_queue gpu_affinity = { on | off } *queue*

*queue*で指定した会話キューに投入されたリクエストにおいて、GPU-CPU Affinity機能を使用するかどうかを設定します。

onの場合は、GPU-CPU Affinity機能を使用します。offの場合はGPU-CPU Affinity機能を使用しません。GPU-CPU Affinity機能を使用する場合は、ソケットスケジューリング機能を有効にする必要があります。既定値はoffです。

【アクセス権】操作員権限が必要です。

set interactive_queue group_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一グループのリクエストの投入数の最大数を設定します。

キュー*queue* 内に同時に存在する同一グループのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。

既定値は unlimited(0)です。

グループ名を個別指定しての制限が設定されていないグループのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set interactive_queue group_submit_limit = *limit*　**groups** = *group*　*queue*

set interactive_queue group_submit_limit = *limit*　**groups** = (*group*[, *group* ...])　*queue*

queue で指定したキューに対して、*group* で指定したグループのリクエストの投入数の最大数を設定します。
group には、グループ名を指定します。
キュー*queue* 内に同時に存在するグループ*group* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。
limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set interactive_queue groups = *group queue*
set interactive_queue groups = (*group* [, *group* ...]) *queue*

queue で指定された会話キューのグループリストに *group* で指定されたグループを設定します。それまでのグループリストの設定は削除されます。グループリストのモードは、**set interactive_queue access** | **noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

set interactive_queue hca_number_range = (*min*, *max*) mode=*hca-mode queue*

queue で指定された会話キューに *hca-mode* で指定された種別の HCA のポート数制限として（下限値, 上限値）を設定します。*min*, *max* には、0 以上 32 以下の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。*min* と *max* が同じ値の場合は、リクエスト投入時その値のみ指定可能です。
hca-mode には下記のいずれかを指定できます。

io　SCATEFS ダイレクト IO 用の HCA です。
mpi　MPI 用の HCA です。
all　SCATEFS ダイレクト IO 用と MPI 用の両方の HCA です。

この設定により、*queue* で指定された会話キューに投入するリクエストに指定する HCA のポート数がこの設定範囲に制限されます。
既定値は（0, unlimited）です。HCA 種別ごとの制限値は独立しており、お互いの設定には干渉しません。

【アクセス権】操作員権限が必要です。

set interactive_queue hook_function { **on** | **off** } *queue*

queue に指定された会話キューで、NQSV のフックスクリプト機能を使用するかどうか

の設定を行います。

on　　フックスクリプト機能を使用します。
/opt/nec/nqsv/sbin/hook_prog 下に配置されているフックスクリプトを実行します。

off　　フックスクリプト機能を使用しません。（既定値）

【アクセス権】操作員権限が必要です。

set interactive_queue idle_timer = *timer queue*

queue で指定した会話キューに投入された会話リクエストに設定するアイドルタイマーの時間を *timer* に分単位で指定します。*timer* に 0 を指定した場合、アイドルタイマーの設定を行いません。
qlogin コマンド実行時に-l オプションでアイドルタイマーを指定した場合は、qlogin での指定が優先されます。

【アクセス権】操作員権限が必要です。

set interactive_queue intmpi_process_manager = hydra *queue*

queue で指定した会話キューに対して、IntelMPI のリクエストが使用するプロセスマネージャを、**hydra** に設定します。
既定値は **hydra** です。

【アクセス権】操作員権限が必要です。

set interactive_queue jobs_range = (*jobs_min* , *jobs_max*)　*queue*

queue で指定されたキューに対して、リクエストごとのジョブ数の上限値／下限値を設定します。
この設定により、*queue* で指定されたキューに投入されるリクエストのジョブ数は *jobs_min* と *jobs_max* で指定した数の範囲に限定されます。*jobs_min*、*jobs_max* に指定できる数は、1～10240 の整数値です。
queue で指定されたキューにリクエストを投入する際、リクエストに指定されたジョブ数が、範囲外の時は投入エラーとなります。

【アクセス権】操作員権限が必要です。

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

queue で指定されたキューに対して、*group* または *user* で指定した、グループまたはユーザのリクエストごとのジョブ数の上限値／下限値を設定します。
group には、グループ名を指定します。*user* には、ユーザ名を指定します。
ジョブ数は*jobs_min* と*jobs_max* で指定した数の範囲に限定されます。*jobs_min*、*jobs_max* に指定できる数は、1～10240 の整数値です。
queue で指定されたキューにリクエストを投入する際、リクエストのグループ・ユーザが、*group* または *user* である場合、リクエストに設定されたジョブ数が、ここで設定された範囲外の時、投入エラーとなります。

【アクセス権】操作員権限が必要です。

set interactive_queue kernel_param nice = *value queue*

queue で指定された会話キューに、nice 値 *value* を設定します。この設定により、*queue* で指定された会話キューに投入された会話リクエストが実行する際に使用する nice 値として *value* が設定されます。

【アクセス権】操作員権限が必要です。

set interactive_queue kernel_param rsg_number = *value queue*

queue で指定された会話キューに、リソースシェアリンググループ番号（RSG 番号）*value* を設定します。この設定により、*queue* で指定された会話キューに投入された会話リクエストが実行する際に使用するリソースシェアリンググループ番号（RSG 番号）として *value* が設定されます。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost cpu_number_range = (*min, max*)　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 台数制限に（下限値, 上限値）を設定します。*min, max* には、1 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。
この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の CPU 台数が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。
既定値は（1, unlimited）です。

【アクセス権】管理者権限が必要です。

set interactive_queue lhost cpu_time_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 時間制限に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の CPU 時間が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost gpu_number_range = (*min, max*)　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の GPU 台数制限に（下限値, 上限値）を設定します。*min, max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。
この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の GPU 台数が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost memory_size_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）のメモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）のメモリサイズが、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost ve_number_range = (*min, max*)　*queue*

queue で指定されたキューの論理ホスト毎の VE ノード数制限に（下限値, 上限値）を

設定します。*min*, *max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。
この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎の VE ノード数が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost vecpu_time_range = (*min, max* [,*warn*])　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE の CPU 時間制限に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の VE の CPU 時間が、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost vememory_size_range = (*min, max* [,*warn*]) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。
この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の VE のメモリサイズが、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue lhost virtual_memory_size_range = (*min, max* [,*warn*]) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値に（下限値, 上限値）を設定します。*min* は *max* 以下の値を指定してください。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する論理ホスト毎（ジョブ単位）の仮想メモリサイズが、設定範囲に制限されます。
min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。同時に警告値 *warn* を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。
なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。
既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue noaccess *queue*

queue で指定された会話キューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを禁止します。モードを変更した場合、root 以外のそれまでのユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set interactive_queue numa_control = { **on** | **off** }　*queue*

queue に指定されたキューに対して、NUMA アーキテクチャの実行ホストに対するソケットスケジューリング機能の ON/OFF を設定します。
この設定を **on** にセットしたキューに対しては、NUMA アーキテクチャのスカラーマシン（Linux）の実行ホスト上で起動したジョブサーバをバインドする必要があります。また、すべて同じソケット構成の実行ホストとする必要があります。

on	ソケットスケジューリング機能を利用します
off	ソケットスケジューリング機能を利用しません　（既定値）

【アクセス権】操作員権限が必要です。

set interactive_queue numa_option core_bind_policy = *policy*　*queue*

queue に指定されたキューに対して、*policy* で指定されたコアバインドポリシーを設定します。*policy* には以下を指定することができます。

concentration	ソケット集中ポリシー　（既定値） 空きコアがより少ないソケットを優先して割り当てます。
balance	ソケット分散ポリシー 空きコアがより多いソケットを優先して割り当てます。

【アクセス権】操作員権限が必要です。

set interactive_queue numa_option memory_allocation_policy = *policy*　*queue*

queue に指定されたキューに対して、*policy* で指定されたメモリアロケーションポリシーを設定します。*policy* には以下を指定することができます。

membind	membind ポリシー ジョブの実行コアが属するソケットのメモリのみを使用します。メモリが不足した場合はスワップを使用します。
localalloc	localalloc ポリシー　（既定値） ジョブの実行コアが属するソケットのメモリを優先的に使用します。 メモリが不足した場合は他ジョブのソケット上のメモリを使用します。
interleave	interleave ポリシー ジョブに割り当てたソケットのメモリを交互に使用します。メモリが不足した場合は他ジョブのソケット上のメモリを使用します。

【アクセス権】操作員権限が必要です。

set interactive_queue numa_unit_check = { **on** | **off** }　*queue*

queue に指定されたキューに対して、リクエスト投入時に指定したジョブあたりの CPU 数とメモリ使用量の制限値が、実行ホストのソケットあたりの搭載 CPU(コア)数とメモリ容量の比率に一致するかのチェックの ON/OFF を設定します。
この設定を ON に設定したキューでは、投入したリクエストが、この比率に一致しない場合は、投入エラーとなります。

【アクセス権】操作員権限が必要です。

set interactive_queue per_job cpu_number_limit = *limit queue*

queue で指定された会話キューのジョブ単位の CPU 台数制限に *limit* を設定します。
この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するジョブ単位の CPU 台数が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で CPU 台数が制限できるのは、実行ホストがジョブ単位の CPU 台数制限機能をサポートしている場合のみです。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_job cpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位の CPU 時間制限値に *limit* を設定します。
この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するジョブ単位の CPU 時間制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で CPU 時間が制限できるのは、実行ホストが

ジョブごとの CPU 時間制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_job gpu_number_limit = *limit*　*queue*

queue で指定されたキューのジョブ単位の GPU 台数制限に *limit* を設定します。この設定により、*queue* で指定されたキューに投入するリクエストに指定するジョブ単位の GPU 台数が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上で GPU 台数が制限できるのは、実行ホストが GPU をサポートしている場合のみです。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_job memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位のメモリサイズ制限値に *limit* を設定します。
この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するジョブ単位のメモリサイズ制限値が *limit* に制限されます。
ただし、ジョブの実行時に実行ホスト上でメモリサイズ制限ができるのは、実行ホストがジョブごとのメモリサイズ制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_job virtual_memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位の仮想メモリサイズ制限値に *limit* を設定します。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがジョブごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
仮想メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc core_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のコアファイルサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位のコアファイルサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でコアファイルサイズ制限ができるのは、実行ホストがプロセスごとのコアファイルサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

コアファイルサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc cpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の CPU 時間制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位の CPU 時間制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で CPU 時間制限ができるのは、実行ホストがプロセスごとの CPU 時間制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

CPU 時間制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc data_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のデータサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位のデータサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でデータサイズ制限ができるのは、実行ホストがプロセスごとのデータサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

データサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc file_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のファイルサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位のファイルサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でファイルサイズ制限ができるのは、実行ホストがプロセスごとのファイルサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

ファイルサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc open_file_number_limit = *limit queue*

queue で指定された会話キューのプロセス単位の同時オープンファイル数制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位の同時オープンファイル数制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で同時オープンファイル数制限ができるのは、実行ホストがプロセスごとの同時オープンファイル数制限機能をサポートしている場合のみです。

同時オープンファイル数制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc stack_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のスタックサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位のスタックサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上でスタックサイズ制限ができるのは、実行ホストがプロセスごとのスタックサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

スタックサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc vecpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の VE CPU 時間制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位の VE CPU 時間制限値が *limit* に制限されます。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

VE CPU 時間制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc vememory_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の VE メモリサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位の VE メモリサイズ制限値が *limit* に制限されます。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

VE メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_prc virtual_memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の仮想メモリサイズ制限値に *limit* を設定します。

この設定により、*queue* で指定された会話キューに投入する会話リクエストに指定するプロセス単位の仮想メモリサイズ制限値が *limit* に制限されます。

ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがプロセスごとの仮想メモリサイズ制限機能をサポートしている場合のみです。

制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。

仮想メモリサイズ制限値の既定値は無制限です。

【アクセス権】操作員権限が必要です。

set interactive_queue per_req elapse_time_limit = (*limit*)　*queue*

queue で指定されたキューのリクエスト単位の経過時間制限値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際、リクエストに設定された制限値とここで設定された *limit* を比較し、*limit* を超える場合には投入エラーとなりま

す。
制限値として指定する *limit* の構文については、後述の**資源制限**の項を参照してください。
既定値は無制限です。

【アクセス権】操作員権限が必要です。

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
キュー*queue* に対して、*group* または*user* で指定した、グループまたはユーザのリクエスト単位の経過時間制限値として *limit* を設定します。
group には、グループ名を指定します。*user* には、ユーザ名を指定します。
queue で指定されたキューにリクエストを投入する際に、リクエストのグループ・ユーザが、*group* または *user* である場合、リクエストに設定された制限値とここで設定された *limit* を比較し、*limit* を超える時は投入エラーとなります。

【アクセス権】操作員権限が必要です。

set interactive_queue priority = *priority queue*
queue で指定された会話キューのキュープライオリティを *priority* に設定します。

【アクセス権】操作員権限が必要です。

set interactive_queue qattach = *mode queue*
queue で指定された会話キューに投入された会話リクエストに対して **qattach(1)** によるアタッチを許可するか否かを *mode* で設定します。アタッチを許可する場合は、*mode* に"enable"を指定し、許可しない場合は、"disable"を指定します。
既定値は、"enable"です。

【アクセス権】管理者権限が必要です。

set interactive_queue real_time_scheduling = *mode queue*
queue で指定した会話キューに投入された会話リクエストに対して、スケジューラが即時実行開始可能な実行ホストの割り当てを行えなかった場合に、会話リクエストを実行ホストが割り当てられるまで待ち合わせるか、投入をキャンセルしてリターンとするかのモードを *mode* で設定します。*mode* には以下の指定が可能です。

submit_cancel　投入をキャンセルしてリターン
wait　割り当ての待ち合わせを行う
manual　qlogin 実行時の-W オプションの指定に従う

会話キュー作成時点での既定値は、submit_cancel となっています。

【アクセス権】操作員権限が必要です。

set interactive_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定された会話キューへのリクエスト投入に対して、*from* で指定した投入経路からのリクエスト投入を拒否します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit　qlogin または qrsh からのリクエスト投入

【アクセス権】操作員権限が必要です。

set interactive_queue restrict_shell = *path_name queue*

queue で指定した会話キューに投入された会話リクエストが実行時に使用するシェルとして、*path_name* で指定した実行ホスト上のプログラムを使用するように設定します。この設定を行った会話キューに投入した会話リクエストでは、投入時の**-S** オプションでシェルが指定されていても、ここで設定されたシェルが強制的に実行されます。

【アクセス権】操作員権限が必要です。

set interactive_queue standard hca_number = *limit* mode=*hca-mode queue*

queue で指定された会話キューの *hca-mode* で指定された種別の HCA のポート数制限値の既定値として *limit* を設定します。*limit* には 0 以上 32 以下の整数または unlimited が指定できます。

queue で指定された会話キューにリクエストを投入する際に、HCA のポート数制限が指定されていない場合、ここで設定された *limit* が HCA のポート数制限値として使用されます。

hca-mode には下記のいずれかを指定できます。

io　SCATEFS ダイレクト IO 用の HCA です。
mpi　MPI 用の HCA です。
all　SCATEFS ダイレクト IO 用と MPI 用の両方の HCA です。

既定値は 0 です。

HCA 種別ごとの制限値は独立しており、お互いの設定には干渉しません。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost cpu_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の CPU 台数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の CPU 台数の制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost cpu_time_limit = (*limit*)　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の CPU 時間制限が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の CPU 時間制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost gpu_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の GPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の GPU 台数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の GPU 台数の制限値として使用されます。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost memory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）のメモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）のメモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost ve_number_limit = *limit queue*

queue で指定されたキューの論理ホスト毎の VE ノード数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎の VE ノード数が指定されていない場合は、ここで設定された *limit* が論理ホスト毎の VE ノード数の制限値として使用されます。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost vecpu_time_limit = (*limit*)　*queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定されたバッチキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の VE の CPU 時間制限が指定されていない場合は、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の VE の CPU 時間制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost vememory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の VE のメモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard lhost virtual_memory_size_limit = (*limit*) *queue*

queue で指定されたキューの論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値が指定されていない場合、ここで設定された *limit* が論理ホスト毎（ジョブ単位）の仮想メモリサイズ制限値として使用されます。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_job cpu_number_limit = *limit queue*

queue で指定された会話キューのジョブ単位の CPU 台数制限の既定値として *limit* を設定します。

queue で指定された会話キューに会話リクエストを投入する際に、ジョブ単位の CPU 台数が指定されていない場合は、ここで設定された *limit* がジョブ単位の CPU 台数の制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で CPU 台数が制限できるのは、実行ホストがジョブ単位の CPU 台数制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_job cpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位の CPU 時間制限値の既定値として *limit* を設定します。

queue で指定された会話キューに会話リクエストを投入する際に、ジョブ単位の CPU 時間制限が指定されていない場合、ここで設定された *limit* がジョブ単位の CPU 時間制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で CPU 時間が制限できるのは、実行ホストがジョブごとの CPU 時間制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_job gpu_number_limit = *limit　queue*

queue で指定されたキューのジョブ単位の GPU 台数制限の既定値として *limit* を設定します。

queue で指定されたキューにリクエストを投入する際に、ジョブ単位の GPU 台数が指定されていない場合は、ここで設定された *limit* がジョブ単位の GPU 台数の制限値として使用されます。

ただし、ジョブの実行時に実行ホスト上で GPU 台数が制限できるのは、実行ホストがジョブ単位の GPU 台数制限機能をサポートしている場合のみです。

既定値は 0 です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_job memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位のメモリサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、ジョブ単位のメモリサイズ制限値が指定されていない場合、ここで設定された *limit* がジョブ単位のメモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でメモリサイズ制限ができるのは、実行ホストがジョブごとのメモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_job virtual_memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのジョブ単位の仮想メモリサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、ジョブ単位の仮想メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がジョブ単位の仮想メモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがジョブごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc core_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のコアファイルサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位のコアファイルサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のコアファイルサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でコアファイルサイズ制限ができるのは、実行ホストがプロセスごとのコアファイルサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc cpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の CPU 時間制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位の CPU 時間制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の CPU 時

間制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で CPU 時間制限ができるのは、実行ホストがプロセスごとの CPU 時間制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc data_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のデータサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位のデータサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のデータサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でデータサイズ制限ができるのは、実行ホストがプロセスごとのデータサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc file_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のファイルサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位のファイルサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のファイルサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でファイルサイズ制限ができるのは、実行ホストがプロセスごとのファイルサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc open_file_number_limit = *limit queue*

queue で指定された会話キューのプロセス単位の同時オープンファイル数制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位の同時オープンファイル数制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の同時オープンファイル数制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で同時オープンファイル数制限ができるのは、実行ホストがプロセスごとの同時オープンファイル数制限機能をサポートしている場合のみです。

既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc stack_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位のスタックサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位のスタックサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位のスタックサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上でスタックサイズ制限ができるのは、実行ホストがプロセスごとのスタックサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc vecpu_time_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位 VE CPU 時間制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位の VE CPU 時間制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の VE CPU 時間制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc vememory_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の VE メモリサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位の VE メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の VE メモリサイズ制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_prc virtual_memory_size_limit = (*limit*) *queue*

queue で指定された会話キューのプロセス単位の仮想メモリサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、プロセス単位の仮想

メモリサイズ制限値が指定されていない場合、ここで設定された *limit* がプロセス単位の仮想メモリサイズ制限値として使用されます。
ただし、ジョブの実行時に実行ホスト上で仮想メモリサイズ制限ができるのは、実行ホストがプロセスごとの仮想メモリサイズ制限機能をサポートしている場合のみです。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard per_req elapse_time_limit = (*limit*) *queue*

queue で指定された会話キューのリクエスト単位の経過時間制限値の既定値として *limit* を設定します。
queue で指定された会話キューに会話リクエストを投入する際に、リクエスト単位の経過時間制限値が指定されていない場合、ここで設定された *limit* がリクエスト単位の経過時間制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard venode vecpu_time_limit = (*limit*) *queue*

queue で指定された会話キューの VE ノード単位の VE の CPU 時間制限値の既定値として *limit* を設定します。
queue で指定された会話キューにリクエストを投入する際に、VE ノード毎の VE の CPU 時間制限が指定されていない場合は、ここで設定された *limit* が VE ノード毎の VE の CPU 時間制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue standard venode vememory_size_limit = (*limit*) *queue*

queue で指定された会話キューの VE ノード単位の VE のメモリサイズ制限値の既定値として *limit* を設定します。
queue で指定された会話キューにリクエストを投入する際に、VE ノード毎の VE のメモリサイズ制限が指定されていない場合は、ここで設定された limit が VE ノード毎の VE のメモリサイズ制限値として使用されます。
既定値は無制限(unlimited)です。

【アクセス権】操作員権限が必要です。

set interactive_queue submit_cpu_unit = { **cpu** | **socket** | **any** }　*queue*

queue で指定されたキューに対して、リクエスト投入時の、ジョブあたりの CPU 数の

指定の方法を設定します。

cpu	cpunum_job のみ指定可能
socket	socknum_job のみ指定可能
any	cpunum_job、socknum_job のどちらでも指定可能（既定値）

【アクセス権】操作員権限が必要です。

set interactive_queue submit_limit = *limit queue*

queue で指定した会話キューに投入できるリクエストの最大数として *limit* に指定した数を設定します。投入できるリクエスト数とは、1 つのキュー内に同時に存在しているリクエストの総数です。 *limit* で指定した数を超えたリクエストの投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。0 は無制限を表します。既定値は 0（無制限）です。

【アクセス権】操作員権限が必要です。

set interactive_queue submit_venode_range = (*min, max*) *queue*

queue で指定されたキューに VE ノード総数指定（qsub --venode）で投入できる VE ノード総数の範囲（上限値と下限値）を設定します。*min, max* には、0 以上の整数または unlimited が指定できます。*min* は *max* 以下の値を指定してください。

min よりも小さい、または、*max* よりも大きい値で VE ノード総数を指定したリクエストの投入はエラーとなります。

min と *max* が同じ値であった場合、VE ノード総数としてその値しか指定できなくなります。*min* と *max* がともに 0 の場合、VE ノードを使用しないリクエストのみ投入できます。

既定値は（0, unlimited）です。

【アクセス権】操作員権限が必要です。

set interactive_queue supplementary_groups_check { **on** | **off** } *queue*

queue で指定された会話キューへリクエスト投入する際に行う、グループに対するアクセス権チェックにおいて、補助グループ ID を使用するか否かを設定します。**on** の場合はチェックに補助グループ ID を使用し、**off** の場合は使用しません。

【アクセス権】操作員権限が必要です。

set interactive_queue suspend_privilege = *privilege queue*

queue で指定された会話キューに対して、サスペンド要求が可能な権限 *privilege* を設定します。この設定により、*queue* で指定された会話キューに投入されたリクエストを

サスペンドできるのは、*privilege* で設定した権限以上の権限を持つユーザに制限されます。*privilege* には以下の権限が指定できます。

user	一般利用者権限
specialuser	特別利用者権限
groupmanager	グループ管理者権限
operator	操作員権限
manager	管理者権限
scheduler	スケジューラ権限

【アクセス権】操作員権限が必要です。

set interactive_queue userexit location = *location*　**script** = (*script*) **queue** = *queue*

queue で指定された会話キューに投入されたリクエストに対して、*location* で指定されたタイミングで実行する実行するユーザ EXIT のプログラムとして *script* を設定します。*location* に指定できるタイミングは以下のとおりです。

pre-running	ジョブ実行開始前
post-running	ジョブ実行終了直後

また、*script* には、実行するスクリプトのファイル名を指定します。ファイル名にパス要素（'/'）を含めることはできません。指定されたファイルはバッチサーバホスト上の /opt/nec/nqsv/sbin/uex_prog 直下に存在する必要があります。
script には、","(カンマ)または、":"(コロン)で区切って、ファイルを 4 つまで指定できます。ファイルを","(カンマ)で区切った場合、前後のファイルを直列に、":"(コロン)で区切って指定した場合は、前後のファイルを並列に実行します。
ファイル名を指定する"("、")"及び区切り文字の","（カンマ）、":"（コロン）の前後の空白文字は自動的に削除されます。空白文字を含むファイル名は指定できません。

【アクセス権】操作員権限が必要です。

set interactive_queue userexit_timeout = *time queue*

queue で指定した会話キューのユーザ Exit のタイムアウト時間を *time* で指定した時間（単位：秒）に設定します。0 を指定した場合、機能は無効です。既定値は無効（0）です。

【アクセス権】操作員権限が必要です。

set interactive_queue userpp_timeout = *time queue*

queue で指定した会話キューのユーザ PP スクリプトのタイムアウト時間を *time* で指定した時間（単位：秒）に設定します。0 を指定した場合、機能は無効です。既定値は 300 秒です。

【アクセス権】操作員権限が必要です。

set interactive_queue user_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一ユーザのリクエストの投入数の最大数を設定します。
キュー*queue* 内に同時に存在する同一ユーザのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。
limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。
既定値は unlimited(0)です。
ユーザ名を個別指定しての制限が設定されていないユーザのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set interactive_queue user_submit_limit = *limit*　**users** = *user*　*queue*
set interactive_queue user_submit_limit = *limit*　**users** = (*user* [,*user*...])　*queue*

queue で指定したキューに対して、*user* で指定したユーザのリクエストの投入数の最大数を設定します。
user には、ユーザ名を指定します。
キュー*queue* 内に同時に存在するユーザ*user* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。
limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set interactive_queue users = *user queue*
set interactive_queue users = (*user* [, *user* ...]) *queue*

queue で指定された会話キューのユーザリストに *user* で指定されたユーザを設定します。それまでのユーザリストの設定は削除されます。ユーザリストのモードは、**set interactive_queue access | noaccess** サブコマンドによって設定しておく必要があります。 *user* には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。
たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、″user1″か″[1001]″のどちらかを指定します。

【アクセス権】管理者権限が必要です。

set interactive_queue venode vecpu_time_range = *(min, max* [*,warn*] *) queue*

queue で指定された会話キューの VE ノード毎の VE の CPU 時間制限に（下限値,上限値）を設定します。

この設定により、*queue* で指定されたキューに投入するリクエストに指定する VE ノード毎の VE の CPU 時間が、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

同時に警告値 warn を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は *（0, unlimited）* です。

【アクセス権】操作員権限が必要です。

set interactive_queue venode vememory_size_range = (*min, max* [*,warn*]) *queue*

queue で指定されたキューの VE ノード毎の VE のメモリサイズ制限値に（下限値, 上限値）を設定します。

min は *max* 以下の値を指定してください。この設定により、*queue* で指定されたキューに投入するリクエストに指定する VE ノード毎の VE のメモリサイズが、設定範囲に制限されます。

min と *max* が同じ値の場合は、リクエスト投入時にその値のみ指定可能です。

同時に警告値 warn を設定することもできます。*warn* を指定しなかった場合には、*max* の値が自動的に設定されます。

なお、制限値として指定する値の単位等の構文については、後述の**資源制限**の項を参照してください。

既定値は *（0, unlimited）* です。

【アクセス権】操作員権限が必要です。

set job_server hca_failure_check = {off | down | unbind} job_server_id = *jsvid*

jsvid で指定したジョブサーバに対する HCA 障害時の動作を設定します。

off を指定すると、HCA 障害を検知しても何もしません。unbind を指定すると障害を検知すると、キューから JSV をアンバインドします。実行中のジョブが存在する場合、そのジョブは実行を継続します。down を指定すると、HCA 障害を検知した際に JSV を終了させます。実行中のジョブが存在する場合、そのジョブはリランされます。

【アクセス権】操作員権限が必要です。

set job_server health_check_action = {nothing | unbind | down} job_server_id = *jsvid*

jsvid で指定したジョブサーバに対する、ノードヘルスチェックで障害を検知した時の動作を設定します。

nothing を指定すると、ノードヘルスチェックで障害を検知しても何もしません。

unbind を指定するとノードヘルスチェックで障害を検知すると、キューから JSV をアンバインドします。実行中のジョブが存在する場合、そのジョブは実行を継続します。

down を指定すると、ノードヘルスチェックで障害を検知した際に JSV を終了させます。実行中のジョブが存在する場合、そのジョブはリランされます。

【アクセス権】操作員権限が必要です。

set job_server migration_file_transfer_parameter interface_hostname = *hostname* **socketbuffer_size =** (*socket_size*) **iobuffer_size =** (*io_size*) **job_server_id =** *job_server_id*

job_server_id で指定したジョブサーバにジョブマイグレーション時のファイル転送用パラメータを設定します。

hostname には、ファイル転送時の TCP/IP 接続用ネットワークインタフェースのホスト名を指定します。

socket_size には、ソケットバッファサイズを 0 以上の整数値で指定します。0 を指定した場合は OS 既定のソケットバッファサイズを使用します。 *socket_size* の初期設定値は 0 です。

io_size には、ファイル I/O サイズを指定します。*io_size* には 1 バイトから 8 メガバイトまでの値を指定できます。*io_size* の初期設定値は 512 キロバイトです。

【アクセス権】操作員権限が必要です。

set network_queue batch_request_run_limit = *run_limit　queue*

queue で指定したネットワークキューに、*run_limit* に指定したリクエストごとのネットワークリクエスト同時実行数を設定します。*run_limit* には、1 から 1000 までの整数値を指定できます。

【アクセス権】操作員権限が必要です。

set network_queue priority = *priority　queue*

queue で指定したネットワークキューにプライオリティ *priority* を設定します。

【アクセス権】操作員権限が必要です。

set network_queue run_limit = *run_limit　queue*

queue で指定したネットワークキューで同時に転送することができるリクエストの最大

数を *run_limit* で指定した数に設定します。

【アクセス権】操作員権限が必要です。

set network_queue staging_extended_buffer_size = *size　queue*

queue で指定したネットワークキューでステージングファイル転送に使用する拡張バッファサイズを *size* に設定します。*size* には、キロバイト単位で、0 から 512 の範囲の整数値が指定可能です。0 を指定した場合は標準のステージングファイル転送用バッファが使用されます。

【アクセス権】操作員権限が必要です。

set network_queue staging_machine = *host_name　queue*

queue で指定したネットワークキューのステージング対象クライアントホストを *host_name* で指定したホストとします。

【アクセス権】操作員権限が必要です。

set network_queue staging_method = *method　queue*

queue で指定したネットワークキューのステージング方式を *method* で指定した方式に設定します。*method* には以下の方式が指定可能です。

internal	バッチサーバ内蔵の標準ステージング方式を使用します
external	ユーザ定義の外部ステージング方式を使用します

【アクセス権】操作員権限が必要です。

set node_group comment = "*comment*" *ngrp_name*

ngrp_name で指定したノードグループのコメントとして *comment* に指定した文字列を設定します。コメントが既に設定されている場合は、上書きされます。

【アクセス権】操作員権限が必要です。

set node_group network = *network_name　ngrp_name*

ngrp_name で指定したクラウドバースティングノードグループの情報を変更します。
*network_name*に起動するクラウドインスタンスが置かれているクラウド環境のネットワークの名前を指定します。クラウド環境のネットワークを区別する必要がある場合に指定します。最大は 255 文字まで指定できます。
このサブコマンドは cloud タイプのノードグループに対してのみ実行できます。

【アクセス権】操作員権限が必要です。

set node_group priority = *priority*　*ngrp_name*

priority にクラウドバースティングノードグループの優先度を指定します。0～63 が指定できます。

このサブコマンドは cloud タイプのノードグループに対してのみ実行できます。

【アクセス権】操作員権限が必要です。

set node_group switch_layer = *layer ngrp_name*

layer にスイッチレイヤーのレベルを指定します。1～4 が指定できます。

このサブコマンドは nw_topo タイプのノードグループに対してのみ実行できます。

【アクセス権】操作員権限が必要です。

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*

template_name で指定した名前の OpenStack テンプレート情報の内容を変更します。但し、テンプレート *template_name* を使用しているリクエストが存在する場合は、変更できません。

テンプレート情報として、以下を指定します。

image	OpenStack の OS ディスクイメージ名。
flavor	OpenStack のフレーバ名。
cpu	割り当てる CPU 数。 １以上の整数を指定します。
memsz	割り当てるメモリサイズ。 1 以上の整数で、単位（B, KB, MB, GB, TB, PB,EB）を付けて指定します。
gpu	割り当てる GPU 数。 ０以上の整数を指定します。
boot_timeout	起動見込時間。 秒単位で、1～2147483647 の整数で指定します。

既定値は 900 秒です。

stop_timeout　停止見込時間。
秒単位で、1～2147483647 の整数で指定します。
既定値は 900 秒です。

custom　起動環境として独自に定義した情報。

comment　テンプレートについてのコメント。

【アクセス権】管理者権限が必要です。

set routing_queue access *queue*

queue で指定した転送キューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを許可します。 モードを変更した場合、それまで設定されていたユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set routing_queue noaccess *queue*

queue で指定された転送キューへの、ユーザリストに設定されたユーザ、グループリストに設定されたグループからのアクセスを禁止します。 モードを変更した場合、それまで設定されていたユーザリスト、グループリストの内容は全て削除されます。

【アクセス権】管理者権限が必要です。

set routing_queue destination = *destination queue*

set routing_queue destination = (*destination*, *destination*, ...) *queue*

queue で指定された転送キューの転送先キューとして、*destination* にキュー名[@キューが存在するバッチサーバホスト名]の形式で指定された 1 つまたはそれ以上のキューを設定します。

転送先キューとして、リモート NQSV システム上の転送キュー、バッチキューを指定することで、NQSV システム間でのリクエスト移動を行うことも可能です。

【アクセス権】操作員権限が必要です。

set routing_queue group_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一グループのリクエストの投入数の最大数を設定します。

キュー*queue* 内に同時に存在する同一グループのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。

既定値は unlimited(0)です。
グループ名を個別指定しての制限が設定されていないグループのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set routing_queue group_submit_limit = *limit*　**groups** = *group*　*queue*
set routing_queue group_submit_limit = *limit*　**groups** = (*group*[, *group* ...])　*queue*
queue で指定したキューに対して、*group* で指定したグループのリクエストの投入数の最大数を設定します。
group には、グループ名を指定します。
キュー*queue* 内に同時に存在するグループ*group* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。
limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set routing_queue groups = *group queue*
set routing_queue groups = (*group* [, *group* ...]) *queue*
queue で指定された転送キューのグループリストに *group* で指定されたグループを設定します。それまでのグループリストの設定は削除されます。グループリストのモードは、**set routing_queue access** | **noaccess** サブコマンドによって設定しておく必要があります。 *group* には、グループ名か、グループ ID を指定します。グループ ID を指定する場合は[]で囲んで指定します。
たとえば、グループ名が group1 でグループ ID が 2001 のグループを追加する場合、*group* として、"group1"か"[2001]"のどちらかを指定します。

【アクセス権】管理者権限が必要です。

set routing_queue hook_function { **on** | **off** } *queue*
queue に指定された転送キューで、NQSV のフックスクリプト機能を使用するかどうかの設定を行います。

on　フックスクリプト機能を使用します。
/opt/nec/nqsv/sbin/hook_prog 下に配置されているフックスクリプトを実行します。

off　フックスクリプト機能を使用しません。（既定値）

【アクセス権】操作員権限が必要です。

set routing_queue priority = *priority queue*

queue で指定された転送キューのプライオリティを *priority* に設定します。

【アクセス権】操作員権限が必要です。

set routing_queue refuse_submission = (*from* [, *from* ...]) *queue*

queue で指定された転送キューへのリクエスト投入に対して、*from* で指定した投入経路からのリクエスト投入を拒否します。*from* に指定できる投入経路は以下のとおりです。

qsub | submit	qsub からのリクエスト投入
qmove	qmove からのリクエスト投入
local_routing	ローカルからのリクエスト転送
remote_routing	リモートからのリクエスト転送

【アクセス権】操作員権限が必要です。

set routing_queue run_limit = *run_limit　queue*

queue で指定した転送キューで同時に転送することができるリクエストの最大数を *run_limit* で指定した数に設定します。

【アクセス権】操作員権限が必要です。

set routing_queue submit_limit = *limit queue*

queue で指定された転送キューに投入できるリクエストの最大数として *limit* に指定した数を設定します。投入できるリクエスト数とは、1 つのキュー内に同時に存在しているリクエストの総数です。 *limit* で指定した数を超えたリクエストの投入はエラーとなります。

limit には、0 から 1000000 までの値を指定できます。 0 は無制限を表します。既定値は 0（無制限）です。

【アクセス権】操作員権限が必要です。

set routing_queue supplementary_groups_check { **on** | **off** } *queue*

queue で指定された転送キューへリクエスト投入する際に行う、グループに対するアクセス権チェックにおいて、補助グループ ID を使用するか否かを設定します。**on** の場合はチェックに補助グループ ID を使用し、**off** の場合は使用しません。

【アクセス権】操作員権限が必要です。

set routing_queue user_submit_limit = *limit*　*queue*

queue で指定したキューに対して、同一ユーザのリクエストの投入数の最大数を設定します。

キュー*queue* 内に同時に存在する同一ユーザのリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には 0 から 1000000 までの値を指定できます。投入できるリクエスト数の制限を行わない場合は、*limit* に 0、または、"unlimited"を指定します。

既定値は unlimited(0)です。

ユーザ名を個別指定しての制限が設定されていないユーザのリクエストは、本設定が適用されます。

【アクセス権】操作員権限が必要です。

set routing_queue user_submit_limit = *limit*　**users** = *user*　*queue*

set routing_queue user_submit_limit = *limit*　**users** = (*user* [,*user*...])　*queue*

queue で指定したキューに対して、*user* で指定したユーザのリクエストの投入数の最大数を設定します。

user には、ユーザ名を指定します。

キュー*queue* 内に同時に存在するユーザ*user* のリクエスト数が*limit* で指定した数を超えてリクエストの投入を行った場合、投入はエラーとなります。

limit には、0 以上の整数を指定できます。0 を指定した場合、無制限として扱います。

【アクセス権】操作員権限が必要です。

set routing_queue users = *user queue*

set routing_queue users = (*user* [, *user* ...]) *queue*

queue で指定された転送キューのユーザリストに *user* で指定されたユーザを設定します。それまでのユーザリストの設定は削除されます。ユーザリストのモードは、**set routing_queue access | noaccess** サブコマンドによって設定しておく必要があります。

user には、ユーザ名か、ユーザ ID を指定します。ユーザ ID を指定する場合は[]で囲んで指定します。

たとえば、ユーザ名が user1 でユーザ ID が 1001 のユーザを追加する場合、*user* として、″user1″か″[1001]″のどちらかを指定します。

【アクセス権】管理者権限が必要です。

show all

バッチサーバ、キュー、ジョブサーバ、ノードグループ、スケジューラについての情報

を表示します。

show all_queue

全種類のキューの一覧を表示します。

show batch_server

バッチサーバの設定情報を表示します。

show execution_queue　[= *que_name*]

バッチキューの一覧を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show interactive_queue　[= *que_name*]

会話キューの一覧を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show job_server

ジョブサーバの一覧を表示します。

show limit

グループ名・ユーザ名を個別指定して設定できる制限に関連する情報を表示します。

show long all_queue

全種類のキューの詳細情報を表示します。

show long execution_queue [= *que_name*]

全バッチキューの詳細情報を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show long interactive_queue [= *que_name*]

全会話キューの詳細情報を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show long network_queue [= *que_name*]

全ネットワークキューの詳細情報を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show long routing_queue [= *que_name*]

全転送キューの詳細情報を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show network_queue [= *que_name*]

ネットワークキューの一覧を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show node_group [= *ngrp_name*]

ノードグループの詳細情報を表示します。*ngrp_name* を指定した場合、指定したノードグループの情報のみを表示します。

show routing_queue [= *que_name*]

転送キューの一覧を表示します。*que_name* を指定した場合、指定したキューの情報のみを表示します。

show scheduler

スケジューラの情報を表示します。

shutdown

NQSV をシャットダウンします。

【アクセス権】管理者権限が必要です。

start all queue

バッチサーバ上のバッチキュー、会話キュー、転送キュー、ネットワークキューを実行可能状態（ACTIVE）にします。すべてのキュー上にあるリクエストの実行が可能になります。

【アクセス権】操作員権限が必要です。

start execution_queue = *queue*

queue で指定されたバッチキューを実行可能状態（ACTIVE）にします。指定したバッチキュー上にあるリクエストの実行が可能になります。

【アクセス権】操作員権限が必要です。

start interactive_queue = *queue*

queue で指定された会話キューを実行可能状態（ACTIVE）にします。指定した会話キュー上にあるリクエストの実行が可能になります。

【アクセス権】操作員権限が必要です。

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

実行ホスト *host_name* 上で、ジョブサーバ ID を *n* としてジョブサーバを起動します。すでに、**attach execution_host** サブコマンドで、実行ホスト *host_name* に対するジョブサーバがバッチサーバに登録されている場合は、job_server_id =に指定する *n* と、登録済みのジョブサーバ ID とが一致している必要があります。
また、**attach execution_host** サブコマンドで、実行ホスト *host_name* に対するジョブサーバがバッチサーバに登録されている場合は、execution_host **=** *host_name*、または、job_server_id = *n* の指定のみでジョブサーバが起動可能です。

node_group =にノードグループ名、*ngrp_name* を指定することで、ノードグループに属している全ジョブサーバを一括して起動することができます。
start job_server all により、**attach execution_host** サブコマンドでバッチサーバに登録済みの全ジョブサーバを一斉に起動することができます。

このコマンドの実行のためには、実行ホスト上にランチャー・デーモン（nqs_lchd）が常駐している必要があります。

【アクセス権】管理者権限が必要です。

start network_queue = *queue*

queue で指定されたネットワークキューを実行可能状態（ACTIVE）にします。指定したネットワークキュー上にあるネットワークリクエストの実行が可能になります。

【アクセス権】操作員権限が必要です。

start routing_queue = *queue*

queue で指定された転送キューを実行可能状態（ACTIVE）にします。指定した転送キュー上にあるリクエストの転送が可能になります。

【アクセス権】操作員権限が必要です。

stop all queue

バッチサーバ上のバッチキュー、会話キュー、転送キュー、ネットワークキューを実行不可状態（INACTIVE）にします。すべてのキュー上にあるリクエストの実行開始ができなくなります。ただし、既に実行中のリクエストはそのまま実行を継続します。

【アクセス権】操作員権限が必要です。

stop execution_queue = *queue*

queue で指定されたバッチキューを実行不可状態（INACTIVE）にします。指定したバッチキュー上にあるリクエストの実行開始ができなくなります。ただし、既に実行中のリクエストはそのまま実行を継続します。

【アクセス権】操作員権限が必要です。

stop interactive_queue = *queue*

queue で指定された会話キューを実行不可状態（INACTIVE）にします。指定した会話キュー上にある会話リクエストの実行開始ができなくなります。ただし、既に実行中のリクエストはそのまま実行を継続します。

【アクセス権】操作員権限が必要です。

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**]

execution_host **=** に対して *host_name* を指定することで、指定した実行ホスト上のジョブサーバをシャットダウンします。
また、job_server_id =に対してジョブサーバ ID として *n* を指定することで、指定したジョブサーバ ID を持つジョブサーバをシャットダウンします。

さらに、node_group =にノードグループ名、*ngrp_name* を指定することで、ノードグループに属している全ジョブサーバを一括してシャットダウンすることができます。
stop job_server all により、**attach execution_host** サブコマンドでバッチサーバに登録済みの全ジョブサーバを一斉にシャットダウンすることができます。
ただし、**force** が指定しない場合、ジョブを持つジョブサーバのシャットダウンは拒否されます。
force を指定した場合は、ジョブサーバ上で実行中のジョブをアボートさせた上で、ジョブサーバをシャットダウンします。

【アクセス権】管理者権限が必要です。

stop network_queue = *queue*

queue で指定されたネットワークキューを実行不可状態（INACTIVE）にします。指定したネットワークキュー上にあるネットワークリクエストの実行開始ができなくなります。ただし、既に実行中のリクエストはそのまま実行を継続します。

【アクセス権】操作員権限が必要です。

stop routing_queue = *queue*

queue で指定された転送キューを実行不可状態（INACTIVE）にします。指定した転送キュー上にあるリクエストの転送開始ができなくなります。ただし、既に転送実行中のリクエストはそのまま転送されます。

【アクセス権】操作員権限が必要です。

unbind execution_queue job_server *queue* **job_server_id** = *n*

queue に指定したバッチキューと、 ジョブサーバ ID が *n* に一致するジョブサーバの接続関係を切り離します。

【アクセス権】管理者権限が必要です。

unbind execution_queue node_group *queue* **node_group** = *ngrp_name*

queue で指定されたバッチキューと、*ngrp_name* で指定されたノードグループの接続関係を切り離します。

【アクセス権】管理者権限が必要です。

unbind execution_queue scheduler *queue*

queue に指定したバッチキューに接続されているスケジューラの接続を切り離します。

【アクセス権】管理者権限が必要です。

unbind interactive_queue job_server *queue* **job_server_id** = *n*

queue に指定した会話キューと、ジョブサーバ ID が *n* に一致するジョブサーバの接続関係を切り離します。

【アクセス権】管理者権限が必要です。

unbind interactive_queue node_group *queue* **node_group** = *ngrp_name*

queue で指定された会話キューと、*ngrp_name* で指定されたノードグループの接続関係を切り離します。

【アクセス権】管理者権限が必要です。

unbind interactive_queue scheduler *queue*

queue に指定した会話キューに接続されているスケジューラの接続を切り離します。

【アクセス権】管理者権限が必要です。

unlock cloud_template = t*emplate_name*

template_name で指定した名前のクラウドバースティングテンプレートのロックを解除します。

【アクセス権】管理者権限が必要です。

unlock container_template = t*emplate_name*

template_name で指定した名前のコンテナ用テンプレートのロックを解除します。

【アクセス権】管理者権限が必要です。

unlock node_group = *ngrp_name*

ngrp_name で指定した名前のクラウドバースティングノードグループのロックを解除します。

【アクセス権】管理者権限が必要です。

unlock openstack_template = t*emplate_name*

template_name で指定した名前の OpenStack テンプレートのロックを解除します。

【アクセス権】管理者権限が必要です。

update license

nqsd.conf のライセンスの設定を読み込み、改めて、ライセンスサーバーからライセンスを取得します。

nqsd.conf のライセンス設定を変更した場合やライセンスサーバーでのライセンス保有数が変更になった場合等に使用します。

エラーが発生した場合は、ライセンスの保有状況を、「qstat -B -L -Pm」コマンドで確認してください。

【アクセス権】管理者権限が必要です。

資源制限

各種 set サブコマンドで指定する資源制限値の指定形式は以下のとおりです。

limit = *max_limit* [, *warn_limit*]
　　max_limit：　資源制限の最大値
　　warn_limit：　資源制限の警告値

1.　時間制限

時間に関する制限は以下の形式で指定します。
　　[[*hours*:]*minutes*:]*seconds*[.*fraction*]
無制限の場合は、"unlimited"という文字列を指定します。

【指定例】
　1234:58:21.29　1234 時間 58 分 21.29 秒
　59:01　59 分 1 秒
　12345　12345 秒
　121.1　121.1 秒

2.　サイズ制限

サイズに関する制限は以下の形式で指定します。
　　integer[.*fraction*][*units*]

units に指定できる単位は、以下のとおりです。
　　b　バイト
　　kb　キロバイト(1024 b)
　　mb　メガバイト(1024 kb)
　　gb　ギガバイト(1024 mb)
　　tb　テラバイト(1024 gb)
　　pb　ペタバイト(1024 tb)
　　eb　エクサバイト(1024 pb)

units の指定がない場合は、バイトと解釈されます。
無制限の場合は、"unlimited"という文字列を指定します。

【指定例】
　　1234　1234 バイト
　　1234kb　1234 キロバイト
　　1234.5gb　1234.5 ギガバイト

環境変数

NQSV_PRIV または NQSII_PRIV
　qmgr 実行時に-P オプションでアクセス権を指定しない場合に使用するアクセス権

PRIV_MGR	管理者権限
PRIV_OPE	操作員権限
PRIV_GMGR	グループ管理者権限
PRIV_SPU	特別利用者権限
PRIV_USR	一般利用者権限

関連項目

nmapmgr(1M), qdel(1), qstat(1), qsub(1), qlogin(1)

注意

キューに対するカーネルパラメータ、資源制限値等の設定は、そのキューにリクエストが投入される際にリクエストに設定されます。したがって、キューの設定を変更しても、既に投入されているリクエストには反映されません。

特記事項

Copyright (c) 1992, 1993
The Regents of the University of California.　All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.　IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) 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.

2.5　sbuedit(1M)

名前

sbuedit -- 課金レートの表示、編集

形式

/opt/nec/nqsv/bin/sbuedit {add|upd}
　　queue_name:*acct_item*=*rat*e_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*=*rat*e_in_prime[/*rate_in_nonprime*][,*acct_item*=...]
/opt/nec/nqsv/bin/sbuedit del -t *template_name*
/opt/nec/nqsv/bin/sbuedit [-N]

機能説明

sbuedit コマンドは課金レートの表示、登録、更新および削除などを行います。

オプションを指定しない場合には、登録されているすべての課金レートが、登録されている分だけ表示されます。下記は 1 キュー分の表示例です。

```
------------------------------------------------------
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
```

Custom Resource Rate が設定されていない場合は、Custom Resource Rate 以降は表示されません。課金レートが設定されていないキューに対して投入されたリクエスト、課金レー

トが設定されていないキューを指定して生成されたリソース予約区間は、予算管理の対象となりません。

オプション

オペレーション、適用範囲、適用項目を下記のように指定します。
(テンプレート名は、実行ホストが SX-Aurora TSUBASA システムの場合は利用できません。)

```
# sbuedit add queue1:CPU=5/10,CR_Power=10.0
# sbuedit add -t template1:ELAPSE=10,RESERVE=10.0
```

add　　課金レートを新たに追加します。キュー名・テンプレート名がすでに存在する場合にはエラーメッセージを表示します。

upd　　課金レートを更新します。キュー名・テンプレート名が存在しない場合にはエラーメッセージを表示します。

del　　課金レートを削除します。キュー名・テンプレート名が存在しない場合にはエラーメッセージを表示します。

-N　　sbuedit のバージョン情報を表示します。他のオプションを指定した場合でも、本オプションが優先され他のオプションはすべて無視されます。

queue_name
　　指定したキューに対して課金レートを設定します。

-t *template_name*
　　指定したテンプレートに対して課金レートを設定します。

acct_item
　　後述の課金項目を指定します。

rate_in_prime
　　プライムタイムの課金レートです。

rate_in_nonprime
　　ノンプライムタイムの課金レートです。省略された場合には、プライムタイムと同じ課金レートとなります。

設定項目

acct_item には以下の項目を指定できます。カスタムリソースの課金レートを設定する場合はカスタムリソースの名前の前に”CR_”を付けて指定します。

CPU　　CPU 消費時間 1 秒当たりの金額です。
この項目は課金レートをプライムタイム、ノンプライムタイムで別々に設定できます。

MEM　　単位メモリ使用量(1 キロバイト×分)当たりの金額です。
この項目は課金レートをプライムタイム、ノンプライムタイムで別々に設定できます。

JOB　　1 ジョブ当たりの金額です。

GPUNUM　　経過時間 1GPU1 秒当たりの金額です。

REQVE　　経過時間 1VE1 秒当たりの金額です。課金額の計算に要求した VE ノード数を用います。本項目は実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

RSVVE　　経過時間 1VE1 秒当たりの金額です。課金額の計算に確保した VE ノード数を用います。本項目は実行ホストが SX-Aurora TSUBASA システムの場合のみ有効です。

ELAPSE　　経過時間 1 ジョブ 1 秒当たりの金額です。

RESERVE　　リソース予約区間 1 ノード 1 秒当たりの金額です。

DEC　　宣言量に対する重みづけを指定します。

ACT　　実績量に対する重みづけを指定します。

PRI_MAX　　優先度が最大の場合の重みづけを指定します。
1.0 以上の値を指定できます。

PRI_MIN　　優先度が最小の場合の重みづけを指定します。
0.0 ～ 1.0 の値を指定できます。

CR_*crname*　　指定したカスタムリソースの消費量 1 当たりの金額です。
カスタムリソースのレートは 20 個まで登録できます。
既に登録されているレートを削除する場合はレートに 0 を指定してください。

関連項目

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

2.6　sceditacct(1M)

名前

sceditacct -- アカウント DB の編集

形式

/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

機能説明

sceditacct コマンドは、アカウンティングサーバで集計されたリクエスト情報、ジョブ情報、予約情報のデータの不要な部分を削除または退避することでデータアクセスのパフォーマンスを向上させるコマンドです。

本コマンドは何らかのオプションを設定することが必須となります。

下記のオプションが指定可能です。

-N　　sceditacct のバージョン情報を出力します。他のオプションを指定した場合でも、本オプションが優先され他のオプションはすべて無視されます。

-j　　ジョブアカウントのデータベースが編集の対象となります。

-r　　リクエストアカウントのデータベースが編集の対象となります。

-y　　予約アカウントのデータベースが編集の対象となります。

-u *user*

指定したユーザのデータを削除、または退避します。

-g *group*

指定したグループのデータを削除、または退避します。

-x *acct-code*

指定したアカウントコードのデータを削除、または退避します。

-s *date*

指定された月日あるいはそれ以降に実行を終了したレコードを削除、または退避します。

-e *date*

指定された月日あるいはそれ以前に実行開始したレコードを削除、または退避します。

-S *date*

指定された月日あるいはそれ以降に実行を開始したレコードを削除、または退避します。

-E *date*

指定された月日あるいはそれ以前に実行が終了したレコードを削除、または退避します。

-q *date*

指定された月日あるいはそれ以降に投入し、かつ実行が開始前に削除されたレコードを削除、または退避します。

-Q *date*

指定された月日あるいはそれ以前に投入し、かつ実行が開始前に削除されたレコードを削除、または退避します。

-h *hostname*

指定されたホスト名のデータを削除、または退避します。このオプションは ジョブアカウントのデータベースに対してのみ有効です。

-d *database*

編集の対象となるデータベースを指定します。予約アカウントデータベースを指定する場合はファイル名を指定します。ジョブ・リクエストアカウントデータベースを指定する場合はディレクトリ名を指定します。

本オプションが指定されない場合はデフォルトのデータベースが操作対象となります。

-j 指定時 /var/opt/nec/nqsv/asv/配下の jacct、jacct_cr、jacct_ps, jacct_gpu
-r 指定時 /var/opt/nec/nqsv/asv/配下の racct、racct_tally、racct_cr、
racct_cr_tally、racct_ps_tally および racct_gpu_tally
-y 指定時 /var/opt/nec/nqsv/asv/rsvacct

また、カンマで区切って指定することで複数のデータベースを指定することができます。

-m database

削除対象のデータを退避する際にデータベースファイル名（-y の場合）またはディレクトリ名（-j と-r の場合）を指定します。本オプションを指定した場合、編集対象の DB から本オプションで指定した DB にデータが移動されます。

-c database

-m オプションと同じく退避先のデータベースファイル名（-y の場合）またはディレクトリ名（-j と-r の場合）を指定しますが、退避元となるデータベースのデータは削除しません。本オプションを指定した場合、編集対象の DB から本オプションで指定した DB にデータがコピーされます。 -m および -c オプションを指定しない場合、編集対象の DB から指定されたデータの削除を行います。

関連項目

scacctjob(1-N), scacctreq(1-N), scacctrsv(1-N).

2.7　scsumacct(1M)

名前

scsumacct -- 日報、月報情報表示

形式

/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

機能説明

指定した日もしくは月のアカウントのサマリ情報が出力されます。表示内容としては、リクエストアカウントのサマリ、予約アカウントのサマリが出力され、最終行に指定した日もしくは月の全データの合計が出力されます。

既定値では下記の項目が出力されます。

リクエストアカウントのサマリ

REQUESTS	リクエストの総数
JOBS	ジョブの総数
EHOST NUM	実行ホストの総数
ELAPSE(H)	ジョブの ELAPSE 時間の総和 [時間]
CPU(S)	CPU 消費時間（システム＋ユーザ）[秒]
MEM	のべメモリ使用量[KB×min]

予約アカウントのサマリ

RESERVES	予約の総件数
EHOST NUM	実行ホストの総数
RSVTIME	リソース予約区間の総時間 [時間]
USETIME	リソース予約区間の総使用時間 [時間]

また、次のオプションが指定可能です。

-N　　scsumacct のバージョン情報を出力します。他のオプションを指定した場合でも、本オプションが優先され他のオプションはすべて無視されます。

-s *MM*[/*DD*[/*YY*]]

出力したい日を指定します（本オプションおよび-S が指定されない場合は当日のサ

マリが出力されます)。

-S *MM*[/*YY*]
出力したい月を指定します。

-t　　CPU 消費時間をユーザ/システム別に表示します。

-b　　予定課金額と実課金額を表示します。

-r database
集計の対象となるリクエストアカウントデータベースのディレクトリ名を指定します。本オプションが省略された場合は、デフォルトのデータベース(/var/opt/nec/nqsv/asv/配下の racct および racct_tally)が出力の対象となります。

-y database
集計の対象となる予約アカウントデータベースを指定します。
本オプションが省略された場合は、デフォルトのデータベース(/var/opt/nec/nqsv/asv/rsvacct)となります。

なお、未来の日もしくは月が指定された場合、エラーメッセージが表示されることがあります。

関連項目

sceditacct(1M-N)

注意

蓄積されたアカウントデータの量が著しく多い場合、本コマンドの実行に時 間が かかる場合があります。 sceditacct コマンドで必要のないデータを削除して実行されることをお勧めします。

2.8　scturnacct(1M)

名前

scturnacct -- アカウントデータの退避

形式

/opt/nec/nqsv/bin/scturnacct [-N]

機能説明

scturnacct コマンドは、アカウントファイルを退避するコマンドです。コマンド実行時に、アカウントファイルが設定されたサイズを超えた場合、設定されている世代分アカウントデータを退避します。

退避の対象となるファイルサイズ、世代数は /etc/opt/nec/nqsv/acmd.conf に設定します。デフォルトの値は 500MB、10 世代です。

TURN_SIZE=500 退避を開始する各アカウントのサイズ（単位：MB）
TURN_SAVE=10 退避する世代数

-N オプションが指定された場合、 scturnacct のバージョンが表示されます。

関連項目

なし。

2.9　smgr(1M)

名前

smgr -- JobManipulator の構成管理および運用管理

形式

smgr　[**-P** *privilege*] [**-V**] [**-h** *server_host*] [**-s** *scheduler_id*] [--help] [--version]

機能説明

smgr(1M)コマンドは、システム管理者とシステム操作員が、*server_host* 上の JobManipulator を制御するために使用するプログラムです。

以下のオプションが指定できます。

-P *privilege*

コマンド実行時のアクセス権を指定します。*privilege* には、次の指定ができます。

m	管理者権限となります。
o	操作員権限となります。
g	グループ管理者権限となります。
s	特別利用者権限となります。
u	一般利用者権限となります。

アクセス権は管理者権限がもっとも高く、一般利用者権限が最低となります。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数 **JM_PRIV** で指定されたアクセス権となります。環境変数が設定されていない場合は、一般利用者権限となります。

-V　コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

-h *server_host*

JobManipulator が実行されるサーバホストを指定します。本オプションが指定されない場合、デフォルトの JobManipulator サーバホスト(/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf 内に指定したホスト名)を使用します。デフォルトの JobManipulator サーバホストがない場合は、ローカルホストとなります。

-s *scheduler_id*

JobManipulator のスケジューラ ID を指定します。 本オプションが指定されない場合、スケジューラ ID は、環境変数 **JM_SCHED_ID** で指定されたスケジューラ ID となります。 環境変数が設定されていない場合は、/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定した JobManipulator スケジューラ ID を使用します。いずれの設定もされていない場合は 0 となります。

--help smgr の使い方を表示して終了します。

--version

コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

入力コマンド

smgr コマンドを起動すると、**smgr** はプロンプト **Smgr:**を返します。
その後に後述するサブコマンドを入力できます。
smgr コマンドを終了するには **exit** コマンドを使います。

サブコマンドを複数行にわたって入力する場合は改行の前に行継続文字（バックスラッシュ）を入力します。

入力コマンド中の編集コマンド

Smgr:プロンプト行では以下の編集コマンドが使用できます。

Ctrl-f
　順方向に一文字進みます
Ctrl-b
　逆方向に一文字戻ります
Ctrl-h
　前の文字を削除します
Ctrl-p
　一つ前の入力コマンドを表示します
Ctrl-n
　一つ後の入力コマンドを表示します
<TAB>
　入力されている文字列に一致するサブコマンドを表示もしくは補完します

アクセス権

smgr サブシステムにはアクセス権のタイプ 4 種類あります。
管理者権限はすべてのコマンドを使用することが許されていますが、操作員権限はある一部の制限されたコマンドのみ使用することができます。
一般利用者権限、特別利用者権限およびグループ管理者権限では、**exit**,**help**,**list** サブコマンドのみ使用できます。

ページャ機能

PAGER 環境変数にページングを行うためのプログラムを指定できます。
その際、**help** サブコマンドの出力がそのプログラムの標準入力として使用されます。
たとえば

```
PAGER=/usr/bin/more
```

と設定することで、**help** の出力がページングされて表示されます。
また、指定されたプログラムが実行できないときは、そのまま出力します。

サブコマンド

以下の節に各 **smgr** サブコマンドの構文規則を記述します。
[]は、[]内に示された部分が省略可能であることを示します。

add complex_queue queue = (*queue-name* [,*queue-name...*]) *complex-queue-name*

complex-queue-name で指定されたコンプレックスキューに *queue-name* で指定された実行キューを追加します。
管理者権限が必要です。

create complex_queue = *complex-queue-name* **queue** = (*queue-name* [,*queue-name...*])

complex-queue-name で指定されたコンプレックスキューを作成します。
queue-name には、作成するコンプレックスキューに属する実行キューを指定します。
管理者権限が必要です。

create eco_schedule starttime = *start_time*　**endtime** = *end_time*　**hostname** = *host_name*

計画省電力区間を作成します。

start_time には、計画省電力区間の開始時刻を指定します。現時刻より前の時刻は指定できません。
end_time には、計画省電力区間の終了時刻を指定します。*start_time* との間隔が「ノード停止マージン＋ノード起動マージン」以上となる値を指定してください。
host_name には、対象となる実行ホスト名を指定します。

操作員権限が必要です。

start_time や *end_time* の形式は次のとおりです。

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

各々の 2 桁は次のように表されます。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時間がすでに過ぎている場合は、次の日となります。
本オプションが省略された場合は、現在時刻となります。
開始時間は、分単位に切り捨てられます。

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*

リソース区間を予約します。予約には、キュー指定とホスト指定、あるいは、ノードグループ指定があります。
ホスト指定とノードグループ指定は、どちらかを指定できます。

キュー指定の場合、*queue-name* に指定されたキューにバインドされている実行ホストがリソース区間を予約する対象となります。 *hostnum* には、予約するホスト数を指定します。省略された場合は、全ホストとなります。*group* には、当該予約を利用するグループを指定します。
cpunum には、予約するホスト毎の CPU 台数を指定します。省略された場合は、全 CPU 台数となります。

ホスト名指定の場合は、*host_name* に指定されたホストの全 CPU 台数が対象となります。
host_name には以下の指定が可能です。

- ホスト名

- （ホスト名,ホスト名,ホスト名……）
- ALL_HOSTS(全ホスト)

ノードグループ指定の場合は、*node_group_name* に指定されたノードグループに属する実行ホストに対して一括して予約区間を作成します。
なお、後からノードグループに実行ホスト追加しても追加された実行ホストに対して自動的に予約区間を作成しません。
また、ノードグループから実行ホストを削除しても、作成した予約から当該実行ホストを自動的に削除しません。

start_time には予約区間の開始時刻を指定します。現時刻より前の時刻が指定された場合は、予約されません。
block_time には予約区間の期間を秒単位で指定します。
resource_reservation_name には予約名を指定します。

start_time の形式は次のとおりです。

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

各々の 2 桁は次のように表されます。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時間がすでに過ぎている場合は、次の日となります。
本オプションが省略された場合は、現在時刻となります。
開始時間は、分単位に切り捨てられます。

[例] 17 時開始、予約区間の期間 240 秒(4 分)、キューexecqueu1、CPU 数 1 の場合
create resource_reservation starttime=1700 blocktime=240 queue=execque1 cpunum=1

操作員権限が必要です。

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*]

テンプレート指定のリソース区間を予約します。
start_time には予約区間の開始時刻を指定します。現時刻より前の時刻が指定された場合は、予約されません。
block_time には予約区間の期間を秒単位で指定します。
queue-name に指定されたキューにバインドされている実行ホストがリソース区間を予約する対象となります。
template_name にテンプレート名を指定します。
machine_num に、*template_name* で指定されたテンプレートで起動するマシン数を指定します。値は 1 から 10240 まで指定できます。
resource_reservation_name には予約名を指定します。
group には、当該予約を利用するグループを指定します。

start_time の形式は次のとおりです。

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

各々の 2 桁は次のように表されます。

CC　西暦年のはじめの 2 桁
YY　西暦年の後ろの 2 桁
MM　月(01-12)
DD　日(01-31)
hh　時(00-23)
mm　分(00-59)
SS　秒(00-61)

hhmm で指定された時間がすでに過ぎている場合は、次の日となります。
本オプションが省略された場合は、現在時刻となります。
開始時間は、分単位に切り捨てられます。

[例] 17 時開始、予約区間の期間 240 秒(4 分)、キューexecqueu1、テンプレート名 template1、マシン数 10 の場合

```
create resource_reservation starttime=1700 blocktime=240 queue=execque1
template=template1 machinenum=10
```

操作員権限が必要です。

delete complex_queue = *complex-queue-name*

complex-queue-name で指定されたコンプレックスキューを削除します。

管理者権限が必要です。

delete eco_schedule = *eco_schedule_id*

計画省電力区間を削除します。

eco_schedule_id には、計画省電力 ID を指定します。

操作員権限が必要です。

delete global_group_cpu_run_limit groups = *group-name-list*

group-name-list で指定されたスケジューラ単位でのグループ個別の同時実行 CPU 台数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete global_group_run_limit groups = *group-name-list*

group-name-list で指定されたスケジューラ単位でのグループ個別の同時実行リクエスト数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete global_group_ve_run_limit groups = *group-name-list*

group-name-list で指定されたスケジューラ単位でのグループ個別の同時実行 VE ノード数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete global_user_cpu_run_limit users = *user-name-list*

user-name-list で指定されたスケジューラ単位でのユーザ個別の同時実行 CPU 台数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete global_user_run_limit users = user*-name-list*

user-name-list で指定されたスケジューラ単位でのユーザ個別の同時実行リクエスト数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete global_user_ve_run_limit users = *user-name-list*

user-name-list で指定されたスケジューラ単位でのユーザ個別の同時実行 VE ノード数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete queue group_cpu_run_limit groups = *group-name-list queue-name*

group-name-list に指定したグループの *queue-name* に指定したキュー内でのグループ個別の同時実行 CPU 台数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete queue group_run_limit groups = *group-name-list queue-name*

group-name-list に指定したグループの *queue-name* に指定したキュー内でのグループ個別の同時実行リクエスト数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete queue group_ve_run_limit groups = *group-name-list queue-name*

group-name-list に指定したグループの *queue-name* に指定したキュー内でのグループ個別の同時実行 VE ノード数制限設定を削除します。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

操作員権限が必要です。

delete queue user_cpu_run_limit users = *user-name-list queue-name*

user-name-list に指定したユーザの *queue-name* に指定したキュー内でのユーザ個別の同時実行 CPU 台数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete queue user_run_limit users = *user-name-list queue-name*

user-name-list に指定したユーザの *queue-name* に指定したキュー内でのユーザ個別の同時実行リクエスト数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete queue user_ve_run_limit users = *user-name-list queue-name*

user-name-list に指定したユーザの *queue-name* に指定したキュー内でのユーザ個別の同時実行 VE ノード数制限設定を削除します。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名，ユーザ名，ユーザ名……）

操作員権限が必要です。

delete resource_reservation = *resource_reservation_id* [**force**]

予約区間を削除します。*resource_reservation_id* には、予約区間 ID を指定します。

force が指定された場合は、予約区間内にリクエストが存在する場合でも予約区間を削除します。 このときリクエストも削除されます。

操作員権限が必要です。

リクエストが予約区間内に存在する場合、**force** 指定で削除するには管理者権限が必要です。

exit

smgr コマンドを終了します。

help [*command*]

help 情報を表示します。引数なしで **help** サブコマンドを使用すると、使用できるサブコマンドについての情報を表示します。

引数を付けて **help** サブコマンドを使用すると、指定されたサブコマンドについての情報を表示します。
サブコマンドの指定は、それがユニークである限りは部分的に指定してもかまいません。
より詳しい形で *command* 引数を指定すれば、受け取る情報もより詳しくなります。

remove complex_queue queue = (*queue-name* [,*queue-name...*]) *complex-queue-name*

complex-queue-name で指定されたコンプレックスキューから *queue-name* で指定された実行キューを削除します。
管理者権限が必要です。

resume request = *request_id*

request_id で指定されたリクエストに対してリジューム要求を行います。*request_id* の形式は以下のとおりです。
request_number[.batch_server]
request_number
リクエスト ID のシーケンス番号、パラメトリックリクエストの場合は、その後に[]、サブリクエストの場合は、その後に[n]の形式で指定します。n にサブリクエスト番号を指定します。
batch_server
バッチサーバ名
batch_server が省略された場合は、JobManipulator サーバが使用している NQSV バッチサーバ名となります。
パラメトリックリクエスト ID を指定した場合は、パラメトリック内の全てのサブリクエストに対して、リジューム要求を行います。
リジューム要求を受けたリクエストは、再スケジューリングの対象となり、予定開始時刻になるとリジュームされます。
本サブコマンドでリジューム要求可能なリクエスト状態は SUSPENDED でかつ **smgr** コマンドの **suspend request** サブコマンドによりサスペンドされたリクエストのみです。
操作員権限が必要です。

set assign_policy = *assign-policy*

CPU 台数集中/分散ポリシーを設定します。
assign-policy には以下の指定が可能です。
resource_balance
リソース分散アサインに設定します。
リクエストをアサインする場合その時点でのノードのリソース使用量が一番少ないノードを選択してアサインされます。
CPU_concentration
CPU 台数集中アサインに設定します。

ジョブの使用 CPU 台数が各ノードに設定されている使用可能 CPU 台数制限まで詰めるようにしてジョブをアサインします。この設定の場合は、ジョブの使用 GPU 台数が各ノードの搭載 GPU 台数まで詰めるようにしてジョブをアサインします。

初期設定値は、resource_balance です。
操作員権限が必要です。

set assign_policy_priority = *priority* assign_policy **=** *assign_policy*

下記のアサインポリシーの優先度、あるいは、無効化を設定します。
assign-policy には設定対象のアサインポリシーを指定します。以下のアサインポリシーを指定可能です。

network_topology
　ネットワークトポロジーを考慮したアサイン
staging_job
　予定実行開始時刻がクリアされたステージング中ジョブがあるノード以外のノードの優先アサイン

priority には上記アサインポリシーの優先度 low/high、あるいは、無効化 disable を設定します。disable と設定した場合は、本ポリシーは無効となります。
運用中に変更した場合は、変更したアサインポリシーをそれ以降のスケジューリングに適用します。
初期設定値は、以下の通りです。

network_topology：high
staging_job　　：　low

操作員権限が必要です。

set auto_delete_resource_reservation = on | off

自動予約区間削除機能を有効にするか無効にするかを設定します。
on の場合は、自動予約区間削除機能が有効になります。
off の場合は、自動予約区間削除機能が無効になります。
初期設定値は、**off** です。
操作員権限が必要です。

set cloud_bursting priority_weight_custom_resource = *weight cr_name*

cr_name に指定されたカスタムリソースに対する重み付けを *weight* に指定します。この値はクラウドバースティングプライオリティの計算に使用されます。*weight* に指定可能な値は 0～10000 です。
操作員権限が必要です。

set cloud_bursting priority_weight_resoruce = *weight*

リクエストの要求リソース量(CPU 台数、GPU 台数、メモリ使用量)に対する重み付けを *weight* に指定します。この値はクラウドバースティングプライオリティの計算に使用されます。 *weight* に指定可能な値は 0～10000 です。
初期設定値は、0 です。
操作員権限が必要です。

set cloud_bursting priority_weight_scheduling_priority = *weight*

リクエストのスケジューリングプライオリティに対する重み付けを *weight* に指定します。この値はクラウドバースティングプライオリティの計算に使用されます。 *weight* に指定可能な値は 0～10000 です。
初期設定値は、0 です。
操作員権限が必要です。

set cloud_bursting priority_weight_scheduling_wait_time= *weight*

実行待ち時間に対する重み付けを *weight* に指定します。この値はクラウドバースティングプライオリティの計算に使用されます。 *weight* に指定可能な値は 0～10000 です。
初期設定値は、1 です。
操作員権限が必要です。

set cloud_bursting request_number = *value*

value に一度にクラウドバースティングの対象として選択するリクエストの上限値を設定します。 *value* に指定可能な値は 0～2147483647 です。
クラウドバースティングノードグループで定義された最大のインスタンス数が、実際に一度に利用できる実行ホストの上限値となるため、そのインスタンス数の範囲内で実行できるリクエスト数を指定してください。
初期設定値は、0 です。
操作員権限が必要です。

set complex_queue run_limit = *run_limit complex-queue-name*

run_limit に *complex-queue-name* で指定されたコンプレックスキュー内での同時実行リクエスト数制限を指定します。
run_limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。
操作員権限が必要です。

set complex_queue user_assign_limit = *assign_limit complex-queue-name*

assign_limit に *complex-queue-name* で指定されたコンプレックスキュー内でのユーザ一律の同時アサインリクエスト数制限を指定します。
assign_limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。

操作員権限が必要です。

set complex_queue user_run_limit = *run_limit complex-queue-name*

run_limit に *complex-queue-name* で指定されたコンプレックスキュー内でのユーザ一律の同時実行リクエスト数制限を指定します。

run_limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set dc-off_limit = *number_of_times*

動的省電力運用機能における 1 日あたりのノード停止回数制限値を *number_of_times* に設定します。 1 日に動的省電力により実施したノード停止は、設定した回数以内となります。ただし、「ピークカット」のためのノード停止は停止回数としてカウントしますが、ノード停止は可能です。計画省電力停止も同様です。

number_of_times に指定可能な値は 1-200 です。初期設定値は、5 です。

操作員権限が必要です。

set device_group_topology = on | off

HCA デバイスグループトポロジを意識したスケジューリングを設定します。

on の場合は、HCA デバイスグループトポロジを意識してできるだけ同一 HCA デバイスノードグループに閉じてジョブをアサインします。

off の場合は HCA デバイスグループトポロジを意識せず、下位のポリシーである CPU 台数リソース集中・分散ポリシーに従ってアサインします。例えば、集中ポリシーの場合は VI 内の使用可能 VE ノード数制限まで詰めてジョブをアサインします。

初期設定値は **off** です。

操作員権限が必要です。

set dynamic_dc_control　= on | off

動的省電力運用機能の利用を設定します。

off から **on** に変更した場合は、動的電源制御を開始します。

on から **off** に変更した場合は、動的電源制御を停止します。

JobManipulator とバインドしているキューにバインドしているすべてのノードを即時に起動します。ただし、HW 障害ノードと計画省電力停止中のノードを除きます。

初期設定値は、**off** です。

操作員権限が必要です。

set escalation_interval = *escalation-interval*

エスカレーションの実行間隔をセルサイズ単位で *escalation-interval* に指定します。

escalation-interval は 1 以上が指定可能です。初期設定値は、1 です。

操作員権限が必要です。

set estimated_dc-off_time = *seconds*

seconds に省電力停止見込み時間の閾値を設定します。動的省電力機能によりノードが省電力停止見込み時間の閾値以上停止可能な場合、つまり、現在時刻から省電力停止見込み時間の閾値の間に実行開始予定のジョブがない場合のみ、ノードを停止させます。

省電力停止見込み時間は、秒単位で設定し、設定可能な値は 2-2147483647 です。設定値は「ノード停止マージン + ノード起動マージン」以上に設定してください。初期設定値は、3600 秒 です。
操作員権限が必要です。

set executionhost cpunum_limit_ratio = *limit-ratio execution-host-name* |
node_group = *node_group_name*

execution-host-name に指定された実行ホスト、あるいは、*node_group_name* に含まれている実行ホストの使用可能 CPU 台数制限を設定します。
使用可能 CPU 台数制限は実行ホストの実装している CPU 台数に対する割合を *limit-ratio* に指定します。

例えば、*limit-ratio* に 1 を指定した場合、実装 CPU を全て使用します。
limit-ratio に 0 を指定した場合、本制限は無効となり、CPU 台数はチェックされません。
本制限の初期設定値は、1 です。
なお、*limit-ratio* に指定された値で CPU 台数に変換した結果、端数が生じた場合は切り捨てられます。
例えば、CPU 台数が 1.2 台となった場合は 1 台 となります。
操作員権限が必要です。

set executionhost rsg_cpunum_limit_ratio = *limit-ratio* **rsg_number** = *rsg-number execution-host-name* | **node_group** = *node_group_name*

execution-host-name に指定された実行ホスト、あるいは、*node_group_name* に含まれている実行ホストの *rsg-number* に対応する RSG の使用可能 CPU 台数制限を設定します。
使用可能 CPU 台数制限は RSG 単位で設定している CPU 台数(Icpu)に対する割合を *limit-ratio* に指定します。
例えば、*limit-ratio* に 1 を指定した場合、指定された RSG の CPU 台数を全て使用します。
limit-ratio に 0 を指定した場合、本制限は無効となり、CPU 台数はチェックされません。
本制限の初期設定値は、1 です。なお、*limit-ratio* に指定された値で CPU 台数に変換した結果、端数が生じた場合は切り捨てられます。
例えば、CPU 台数が 1.2 台となった場合は 1 台 となります。
操作員権限が必要です。

set executionhost memsz_limit_ratio = *limit-ratio execution-host-name* | **node_group** = *node_group_name*

execution-host-name に指定された実行ホスト、あるいは、*node_group_name* に含まれている実行ホストの使用メモリ量制限を設定します。
使用メモリ量制限は実行ホストの実装しているメモリ量に対する割合を *limit-ratio* に指定します。
例えば、*limit-ratio* に 1 を指定した場合、実装メモリ量を全て使用します。*limit-ratio* に 0 を指定した場合、本制限は無効となり、メモリ量はチェックされません。
本制限の初期設定値は、無効です。
操作員権限が必要です。

set executionhost rsg_memsz_limit_ratio = *limit-ratio* **rsg_number** = *rsg-number execution-host-name* | **node_group** = *node_group_name*

execution-host-name に指定された実行ホスト、あるいは、*node_group_name* に含まれている実行ホストの *rsg-number* に対応する RSG の使用メモリ量制限を設定します。
使用メモリ量制限は RSG 単位で設定しているラージページのメモリ量(Imem)に対する割合を *limit-ratio* に指定します。
例えば、*limit-ratio* に 1 を指定した場合、指定された RSG のメモリ量を全て使用します。
limit-ratio に 0 を指定した場合、本制限は無効となり、メモリ量はチェックされません。
本制限の初期設定値は、無効です。
操作員権限が必要です。

set forced_rescheduling = **on** | **off**　[**cloud**]

強制再スケジューリングの設定を行います。cloud が指定されていない場合は、クラウドバースティングされない通常のリクエストに対して強制再スケジューリングの設定を行います。
cloud が指定された場合は、クラウドバースティングされたリクエストに対して強制再スケジューリングの設定を行います。

on
　実行中のリクエストが障害に遭遇しストール状態となったとき、再スケジューリングされます。このときリクエストはリランされます。

off
　実行中のリクエストが障害に遭遇しストール状態となったとき、ストールから復旧されるまで再スケジューリングされません。

　初期設定値は、**off** です。
　操作員権限が必要です。

set global_group_cpu_run_limit = *run-limit*　[**groups** = *group-name-list*]

run-limit にスケジューラ単位でのグループ一律/グループ個別の同時実行 CPU 台数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行 CPU 台数制限値となりま

す。**groups** を指定する場合は、指定グループのグループ個別の同時実行 CPU 台数制限値となります。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_group_run_limit = *run-limit* [**groups** = *group-name-list*]

run-limit にスケジューラ単位でのグループ一律/グループ個別の同時実行リクエスト数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行リクエスト数制限値となります。 **groups** を指定する場合は、指定グループのグループ個別の同時実行リクエスト数制限値となります。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_group_ve_run_limit = *run-limit* [**groups** = *group-name-list*]

run-limit にスケジューラ単位でのグループ一律/グループ個別の同時実行 VE ノード数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行 VE ノード数制限値となります。**groups** を指定する場合は、指定グループのグループ個別の同時実行 VE ノード数制限値となります。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_run_limit = *run-limit*

run-limit にスケジューラ単位での同時実行リクエスト数制限を指定します。

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_user_assign_limit = *assign-limit*

assign-limit にスケジューラ単位でのユーザ一律の同時アサインリクエスト数制限を指定します。*assign-limit* に 0 を指定した場合、本制限は無制限となります。本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_user_cpu_run_limit = *run-limit* [**users** = *user-name-list*]

run-limit にスケジューラ単位でのユーザ一律/ユーザ個別の同時実行 CPU 台数制限を指定します。**users** を指定しない場合は、ユーザ一律の同時実行 CPU 台数制限値となります。**users** を指定する場合は、指定ユーザのユーザ個別の同時実行 CPU 台数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_user_run_limit = *run-limit* [**users** = *user-name-list*]

run-limit にスケジューラ単位でのユーザ一律/ユーザ個別の同時実行リクエスト数制限を指定します。 **users** を指定しない場合は、ユーザ一律の同時実行リクエスト数制限値となります。 **users** を指定する場合は、指定ユーザのユーザ個別の同時実行リクエスト数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set global_user_ve_run_limit = *run-limit* [**users** = *user-name-list*]

run-limit にスケジューラ単位でのユーザ一律/ユーザ個別の同時実行 VE ノード数制限を指定します。**users** を指定しない場合は、ユーザ一律の同時実行 VE ノード数制限値となります。**users** を指定する場合は、指定ユーザのユーザ個別の同時実行 VE ノード数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名

- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。

操作員権限が必要です。

set half_reduce_period = *half-reduce-period*

half-reduce-period に半減期を指定します。
半減期の初期設定値は、604800 秒です。
操作員権限が必要です。

set interruption_method = *method*

割り込まれたリクエストを中断させる方法を設定します。
なお、割込み位置が current と指定された場合のみ、本設定が有効になります。
method には以下の指定が可能です。

suspend
　サスペンドする。

rerun
　リランする。
　なお、対象リクエストがリラン禁止に設定されている場合でも、強制的にリランします。

初期設定値は、**suspend** です。
操作員権限が必要です。

set interrupt_to_where = *where*

優先リクエストが割り込む場所（割込み位置）を設定します。
where には以下の指定が可能です。

next_run
　現在実行中のリクエストに影響しないように、アサイン済みリクエストの先頭に割込みます。

current
　現在時刻に割込みます。実行中のリクエストがある場合は、割込まれたリクエストを中断させる方法の設定にしたがって、処理します。

初期設定値は、**next_run** です。
操作員権限が必要です。

set interrupt_assign_block = *on* | *off*

実行待ちの緊急リクエストがアサインされている実行ホストへの新規投入された下位タイプリクエストのアサイン禁止を設定します。

on

実行待ちの緊急リクエストがアサインされている実行ホストへの新規投入された下位タイプリクエストのアサインを禁止します。

off

実行待ちの緊急リクエストがアサインされている実行ホストへの新規投入された下位タイプリクエストのアサインを許可します。

初期設定値は、**on** です。
操作員権限が必要です。

set keep_forward_schedule = *second*

second に、ノード障害時リクエストの開始予定時刻を維持する起点の時刻を、障害発生時刻からの秒数で指定します。予定実行開始時刻が「障害発生時刻+*second* 秒」以降のリクエストの開始予定時刻を維持します。
second に 0 を指定した場合、マップを維持しません。
初期設定値は、0 です。
操作員権限が必要です。

set logfile [**file** = *logfile-name*] [**level** = *log-level*] [**size** = (*logfile-size*)] [**save** = *save-num*]

JobManipulator のログファイルの設定をします。
logfile-name には、ログファイル名を指定します。
log-level には、出力されるログのレベルを指定します。ログレベルには、1 から 5 まで指定できます。
logfile-size には、ログファイルの上限サイズを指定します。
上限サイズには、1024 から 2^31-1 バイトまで指定できます。
上限サイズの指定には、単位の指定(B,KB,MB,GB)ができます。
単位の指定がない場合はバイトとなります。
上限サイズが 0 のときは、無制限となります。
サイズの上限値に達すると自動的にリセットし、以前のログファイルはバックアップファイルに保存します。
save-num には、バックアップファイルの保存数を指定します。
操作員権限が必要です。

set mapsize [**sched_time** = *sched_time*] [**sched_interval** = *sched_interval*]

マップ幅とセルサイズの設定をします。
sched_time には、マップ幅を秒単位で指定します。
sched_interval には、セルサイズを秒単位で指定します。
マップ幅、セルサイズのいずれかが変更された場合、実行中のリクエスト以外は全て再スケジューリングされます。

sched_time の初期設定値は、86400 秒であり、設定可能な値は 3 - 2147483647 秒です。
sched_interval の初期設定値は、60 秒であり、設定可能な値は、2 - 2147483646 です。
sched_time には **sched_interval** より大きい値を設定する必要があります。
操作員権限が必要です。

set max_operation_hosts = *hostnum*

動的省電力運用機能における最大稼働ノード数を *hostnum* で指定します。動的省電力運用機能では、最大稼働ノード数以下となるようにノードを停止します。
hostnum に指定可能な値は 0-10240 です。初期設定値は、10240 です。
操作員権限が必要です。

set min_idle_time = *seconds*

seconds に最低アイドル時間を設定します。最低アイドル時間の計測開始は下記の通りです。
最低アイドル時間の計測開始から設定した最低アイドル時間経過後にノードを停止させます。

- 最低アイドル時間の計測開始は、下記の中で最も遅いタイミングです。
  - ノード上の実行中ジョブがなくなったとき
  - ノードを起動したとき
  - JobManipulator を起動したとき
  - **smgr** コマンドの set dynamic_dc_control サブコマンドにより動的省電力機能を有効に設定したとき
  - ジョブサーバを JobManipulator とバインドしているキューにバインドしたとき
  - ジョブサーバをバインドしているキューに JobManipulator をバインドしたとき

最低アイドル時間は、秒単位で設定し、設定可能な値は 0-2147483647 です。初期設定値は、300 秒 です。
操作員権限が必要です。

set overtake allow_small_request = *on* | *off*

小規模リクエストの追い越し割り当て機能の利用可否を設定します。*on* にした場合は、追い越し制御の閾値を超えたスケジューリングプライオリティ値を持つリクエストが存在しても、そのリクエストの前方の空き領域に収まる、追い越し制御の閾値を超えていないスケジューリングプライオリティ値を持つリクエストをアサイン可能とします。*off* にした場合は、追い越し制御の閾値を超えたスケジューリングプライオリティ値を持つリクエストが存在すれば、追い越し制御の閾値を超えていないスケジューリングプライオリティ値を持つリクエストをアサイン不可とします。

既定値は off です。
操作員権限が必要です。

set overtake_priority = *priority queue-type*

priority に *queue-type* で指定されたキュー種別内での追い越しを不可にするスケジューリングプライオリティ値を設定します。

queue-type には以下の指定が可能です。

normal

　normal タイプ

special

　special タイプ

操作員権限が必要です。

set peak_cut_urgency = **right_now** | **wait_run**

動的省電力運用機能における最大稼働ノード数調整機能によりノードを停止する場合に実行中のジョブがあっても即時に停止するかどうかを設定します。

right_now の場合は、実行中のジョブがあればリランし、即時にノードを停止します。

wait_run の場合は、実行中のジョブの終了を待ち合わせてからノードを停止します。

初期設定値は、**wait_run** です。

操作員権限が必要です。

set priority baseup_interrupted = *baseup* **processing_pattern** = *pattern*

緊急リクエストによって SUSPEND されたリクエストに対するベースアップ値を *baseup* に指定します。

この値は、スケジューリングプライオリティの計算へ使用されます。

pattern には以下の指定が可能で、指定された処理毎にベースアップ値を設定できます。

assign

　アサイン時

escalation

　エスカレーション時

初期設定値は、0 です。

操作員権限が必要です。

set priority baseup_reschedule = *baseup* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素の再スケジューリングリクエストに対するベースアップ値を *baseup* に指定します。

この値はスケジューリングプライオリティの計算へ使用されます。

pattern には以下の指定が可能で、指定された処理毎にベースアップ値を設定できます。

assign

　アサイン時

escalation

　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_request_priority = *weight* **processing_pattern** = *pattern*

実績値のリクエストプライオリティに対する重み付けを *weight* に指定します。この値はスケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_cpu_number = *weight* **processing_pattern** = *pattern*

実績値の CPU 台数に対する重み付けを *weight* に指定します。
この値はスケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_elapse_time = *weight* **processing_pattern** = *pattern*

実績値の経過時間に対する重み付けを *weight* に指定します。
この値はスケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_memory_size = *weight* **processing_pattern** = *pattern*

実績値のメモリサイズに対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_ve_number = *weight* **processing_pattern** = *pattern*

実績値の VE 台数に対する重み付けを *weight* に指定します。
この値はスケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority pastusage_weight_userrank_request_priority = *weight* **processing_pattern** = *pattern*

ユーザランク算出時におけるリクエストプライオリティ実績値に対する重み付けを *weight* に指定します。
この値はユーザランクの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority pastusage_weight_userrank_cpu_number = *weight* **processing_pattern** = *pattern*

ユーザランク算出時における CPU 台数実績値に対する重み付けを *weight* に指定します。
この値は、ユーザランクの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority pastusage_weight_userrank_elapse_time = *weight* **processing_pattern** = *pattern*

ユーザランク算出時における経過時間実績値に対する重み付けを *weight* に指定します。
この値は、ユーザランクの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority pastusage_weight_userrank_memory_size = *weight* **processing_pattern** = *pattern*

ユーザランク算出時におけるメモリサイズ実績値に対する重み付けを *weight* に指定します。
この値はユーザランクの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority pastusage_weight_userrank_ve_number = *weight* **processing_pattern** = *pattern*

ユーザランク算出時における VE 台数実績値に対する重み付けを *weight* に指定します。
この値は、ユーザランクの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority weight_user_rank = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザランクに対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_user_share = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザシェアに対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_request_priority = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のリクエストプライオリティに対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、1 です。
操作員権限が必要です。

set priority weight_cpu_number = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザが宣言した CPU 台数に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_elapse_time = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザが宣言した経過時間に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_memory_size = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザが宣言したメモリサイズに対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign

　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_job_number = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のジョブ数に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。*pattern* には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
アサイン時
escalation
エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_ve_number = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素のユーザが宣言した VE 台数に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。

assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_restart_wait_time = *weight* **processing_pattern** = *pattern*

スケジューリングプライオリティ要素の再開待ち時間に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時

escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_run_wait_time = *weight* **processing_pattern** = *pattern*
スケジューリングプライオリティ要素の、キューに投入してからの実行待ち時間に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set priority weight_assign_wait_time = *weight* **processing_pattern** = *pattern*
スケジューリングプライオリティ要素の、アサイン可能になってからの実行待ち時間に対する重み付けを *weight* に指定します。
この値は、スケジューリングプライオリティの計算へ使用されます。
pattern には、以下の指定が可能で、指定された処理毎に重み付けを設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set provisioning_start_retry_time = *value*
プロビジョニング環境により起動された、仮想マシン（VM）、ベアメタルサーバ、およびコンテナの起動失敗実行ホストの再スケジューリング待ち時間を秒単位で *value* に設定します。
設定変更時には、設定変更前から再スケジューリング待ちをしている実行ホストを含めて変更後の値が有効になります。
初期設定値は、0 です。値 0 はすぐに再スケジューリングすることを意味します。
操作員権限が必要です。

set queue assign_policy = *assign-policy queue-name*

queue-name で指定されたキューにリクエストアサインポリシーを設定します。*assign-policy* には、以下の指定が可能です。

resource_balance

リクエストをアサインする場合その時点でのノードのリソース使用量が最も少ないノードを選択してアサインされます。

CPU_concentration

ジョブの使用 CPU 台数が各ノードに設定されている使用可能 CPU 台数制限まで詰めるようにしてジョブをアサインします。

DEFAULT

スケジューラ単位に設定されているアサインポリシーに従ってリクエストをアサインします。

初期設定値は、**DEFAULT** です。

操作員権限が必要です。

set queue cloud_bursting display_bursting_priority= off | on *queue-name*

queue-name で指定されたキューに投入されたリクエストのクラウドバースティングプライオリティの表示、非表示を設定します。

off

sstat [-R] -f の実行結果にリクエストのクラウドバースティングプライオリティを表示しません。なお、クラウドバースティング可能なキューに投入されたリクエストは本設定が off の場合でもクラウドバースティングプライオリティを表示します。

on

sstat [-R] –f の実行結果にリクエストのクラウドバースティングプライオリティを表示します。*queue-name* で指定されたキューではクラウドバースティングが不可の場合でも、キューに投入されたリクエストのクラウドバースティングプライオリティを表示します。

初期設定値は、**off** です。

操作員権限が必要です。

set queue cloud_bursting occupancy_ratio = *value queue-name*

queue-name で指定されたキューにクラウドバースティングを実施するかを決める、スケジューラマップの占有率の閾値を設定します。v*alue* に指定された値を超えた場合は、*queue-name* で指定されたキューに投入されたリクエストがクラウドバースティングできます。

0～1 の範囲の小数が指定できます。0 が指定された場合、当該キューに投入されたリクエストは常にクラウドバースティングできます。1 が指定された場合、当該キューに投入されたリクエストは常にクラウドバースティングしません。

初期設定値は、1 です。

操作員権限が必要です。

set queue deadline mode = off | on *queue-name*

queue-name で指定されたキューのデッドラインスケジューリングの動作を指定します。

off

デッドラインスケジューリングを行いません。

on

デッドラインスケジューリングを行います。

初期設定値は、**off** です。

操作員権限が必要です。

set queue deadline reduce rate = (*R1*, *R2*, *R3*) **time** = (*T1*, *T2*, *T3*, *T4*) *queue-name*

queue-name で指定されたキューのデッドラインスケジューリング 減免率調整パラメータの設定を行います。

減免率調整パラメータは、以下の値を指定します。

R1

減免率下限　（0～1.0 初期設定値 1.0）

R2

オンタイム減免率　（0～1.0 初期設定値 1.0）

R3

減免率上限　（0～1.0 初期設定値 1.0）

T1

減免率減少終了時間　（0 以上の整数 初期設定値 0）

T2

減免率減少開始時間　（0 以上の整数 初期設定値 0）

T3

減免率増加開始時間　（0 以上の整数 初期設定値 0）

T4

減免率増加終了時間　（0 以上の整数 初期設定値 0）

操作員権限が必要です。

set queue dynamic_jsv_priority control = off | on *queue-name*

queue-name で指定されたキューの Dynamic JSV Priority 機能の動作を設定します。

off　　Dynamic JSV Priority 機能が無効になります。

on　　Dynamic JSV Priority 機能が有効になります。

初期設定値は **off** です。

操作員権限が必要です。

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*

queue-name で指定されたキューに Dynamic JSV Priority 機能のプライオリティ計算のための設定をします。

item-name に指定した要素に対して設定を行います。

priority_value には最大プライオリティ値を指定します。初期設定値は 0 です。

item_value には要素の最大値を指定します。初期設定値は 1 です。

base_num には基数を指定します。初期設定値は 0 です。

order には昇順の場合 **asce** を、降順の場合 **desc** を指定します。初期設定値は **asce** です。

item-name には値を設定する要素として以下の文字列の指定が可能です。

running_ve_job	実行中 VE ジョブの有無
remain_elapse	実行中 VE ジョブの残経過時間（割合）
elapse_time_limit	実行中 VE ジョブの経過時間制限値
elapsed_time	実行中 VE ジョブの経過時間
ve_job_num	実行中 VE ジョブの個数
カスタムリソースの名前	実行中 VE ジョブのカスタムリソースで指定したリソース量。カスタムリソースの消費単位は job である必要があります。

操作員権限が必要です。

set queue elapse_margin = *elapse_margin queue-name*

queue-name で指定されたキューの Elapse マージンを *elapse_margin* に指定します。

本設定を行うと、設定された *elapse_margin* を反映させるため、実行中のリクエスト以外は再スケジューリングされます。

elapse_margin は秒単位で設定し、設定可能な値は 0 - 2147483647 秒です。

elapse_margin の初期設定値は 0 です。

操作員権限が必要です。

set queue escalation_limit [**jobs** = *number*] [**period** = *second*] [**forward** = *second*] *queue-name*

queue-name で指定されたキューに横方向エスカレーションの対象となるリクエストを選択する条件を設定します。

設定可能な条件は、以下のとおりです。

jobs

横方向エスカレーションジョブ数制限値。

エスカレーションを禁止するノード変更が発生するジョブ数を指定します。

エスカレーション時に指定されたジョブ数より多くノード変更が発生する場合、横方向エスカレーションは実施しません。
本機能は、リクエストの開始予定時刻が **period** で指定されたエスカレーション開始時間制限内となったとき有効となります。
0 を指定した場合、本機能は無効となります。

period
横方向エスカレーション開始時間制限値。
横方向エスカレーションを禁止する現在時刻からの期間を秒単位で指定します。
0 を指定した場合、**jobs** の値に関係なく本機能は無効となります。

forward
横方向エスカレーション開始時間差制限値。
横方向エスカレーションを禁止するエスカレーション実行前後のリクエストの開始予定時刻差を秒単位で指定します。
0 を指定した場合、本機能は無効となります。

初期設定値は、全て**無効**です。
操作員権限が必要です。

set queue group_cpu_run_limit = *run-limit*　[**groups** = *group-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのグループ一律/グループ個別の同時実行 CPU 台数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行 CPU 台数制限値となります。**groups** を指定する場合は、指定グループのグループ個別の同時実行 CPU 台数制限値となります。
group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue group_run_limit = *run-limit*　[**groups** = *group-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのグループ一律/グループ個別の同時実行リクエスト数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行リクエスト数制限値となります。**groups** を指定する場合は、指定グループのグループ個別の同時実行リクエスト数制限値となります。
group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue group_ve_run_limit = *run-limit*　[**groups** = *group-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのグループ一律/グループ個別の同時実行 VE ノード数制限を指定します。**groups** を指定しない場合は、グループ一律の同時実行 VE ノード数制限値となります。**groups** を指定する場合は、指定グループのグループ個別の同時実行 VE ノード数制限値となります。

group-name-list は以下のように指定します。

- グループ名
- （グループ名，グループ名，グループ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue interrupt_to_where = *where queue-name*

queue-name で指定されたキューに優先リクエストが割り込む場所を設定します。*where* には以下の指定が可能です。

next_run

現在実行中のリクエストに影響しないように、アサイン済みリクエストの先頭に割込みます。

current

現在時刻に割込みます。

実行中のリクエストがある場合は、割込まれたリクエストを中断させる方法の設定にしたがって、処理します。

DEFAULT

スケジューラ単位に設定されているアサインポリシーに従ってリクエストをアサインします。

初期設定値は、**DEFAULT** です。

操作員権限が必要です。

set queue jsv_assign_priority = *priority* **job_server_id** = *job_server_id* | **node_group** = *node_group_name queue*

queue で指定したキューに投入されたリクエストのスケジューリングに関して、*jobserver_id* で指定されたジョブサーバ、あるいは、*node_group_name* で指定したノードグループに含まれるジョブサーバの JSV アサインプライオリティを *priority* に設定します。

リクエストをアサインする際、*priority* の高いジョブサーバを優先して選択します。

priority は整数で設定し、設定可能な値は -2147483648 ～ 2147483647 です。
queue には、JobManipulator にバインドしているキューのみを指定できます。また、*job_server_id* には、ATTACH された実行ホスト上のジョブサーバのジョブサーバ ID のみを指定できます。
初期設定値は、0 です。
操作員権限が必要です。

set queue mapsize sched_time = *sched_time queue-name*

queue-name で指定されたキューでのマップ幅を *sched_time* に指定します。*sched_time* に、スケジューラで設定しているマップ幅を超えて指定した場合はエラーとなります。
本設定を行わないキューについては、スケジューラで設定しているマップ幅となります。
操作員権限が必要です。

set queue min_operation_hosts = *hostnum queue-name*

queue-name で指定されたキューでの最小稼働ノード数を *hostnum* に指定します。
動的省電力運用機能においてノードを停止する場合、最小稼働ノード数を下回らないようにします。
hostnum に指定可能な値は 0-10240 です。初期設定値は、10240 です。
操作員権限が必要です。

set queue network_topology min_group = **off** | **on** *queue-name*

queue-name で指定されたキューに対して、当該キューに投入されたリクエストの最小ネットワークトポロジーノードグループ選択を指定します。

off
　最小ネットワークトポロジーノードグループ選択を行いません。

on
　最小ネットワークトポロジーノードグループ選択を行います。

初期設定値は、off です。
操作員権限が必要です。

set queue resource_reservation pre-margin = *seconds* | **post-margin** = *seconds queue-name*

queue-name で指定されたキューに対して、予約区間マージン（PRE-MARGIN、あるいは、POST-MARGIN）を秒単位で設定します。該当キューの予約区間が存在する場合は、現在の設定値より大きい値に変更できません。現在の設定値より小さい値に変更した場合は、すでに作成した予約に対しても適用します。
初期値は両者とも 0 です。
操作員権限が必要です。

set queue retry_time [**staging** = *staging_retry_time*] [**pre-running** = *pre-running_retry_time*] *queue-name*

queue-name で指定されたキューに対して、ステージイン処理失敗時に再度スケジューリング対象になるまでの待ち時間を *staging_retry_time* で、PRE-RUNNING 処理失敗時に再度スケジューリング対象になるまでの待ち時間を *pre-running_retry_time* で秒単位で設定します。
初期値は両者とも 0 です。
操作員権限が必要です。

set queue run_limit = *run-limit queue-name*
run-limit に *queue-name* で指定されたキュー内での同時実行リクエスト数制限を指定します。
run-limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。
操作員権限が必要です。

set queue schedule_type = *backfill* | *fifo queue-name*
スケジュール種別を設定します。backfill が指定された場合は、バックフィルスケジューリングを行います。fifo が指定された場合は、FIFO スケジューリングを行います。

初期設定値は、backfill です。
操作員権限が必要です。

set queue type = *queue-type queue-name*
実行キューの種類を設定します。*queue-type* には以下の指定が可能です。
normal
　normal タイプ
special
　special タイプ
urgent
　urgent タイプ

初期設定値は、**normal** タイプのキューとなります。
操作員権限が必要です。

set queue user_assign_limit = *assign-limit queue-name*
assign-limit に *queue-name* で指定されたキュー内でのユーザ一律の同時アサインリクエスト数制限を指定します。
assign-limit に 0 を指定した場合、本制限は無制限となります。
本制限の初期設定値は、無制限です。
操作員権限が必要です。

set queue user_cpu_run_limit = *run-limit* [**users** = *user-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのユーザ一律/ユーザ個別の同時実行 CPU 台数制限を指定します。

users を指定しない場合は、ユーザ一律の同時実行 CPU 台数制限値となります。**users** を指定する場合は、指定ユーザのユーザ個別の同時実行 CPU 台数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue user_run_limit = *run-limit* [**users** = *user-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのユーザ一律/ユーザ個別の同時実行リクエスト数制限を指定します。

users を指定しない場合は、ユーザ一律の同時実行リクエスト数制限値となります。**users** を指定する場合は、指定ユーザのユーザ個別の同時実行リクエスト数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue user_ve_run_limit = *run-limit* [**users** = *user-name-list*] *queue-name*

run-limit に *queue-name* で指定されたキュー内でのユーザ一律/ユーザ個別の同時実行 VE ノード数制限を指定します。

users を指定しない場合は、ユーザ一律の同時実行 VE ノード数制限値となります。**users** を指定する場合は、指定ユーザのユーザ個別の同時実行 VE ノード数制限値となります。

user-name-list は以下のように指定します。

- ユーザ名
- （ユーザ名,ユーザ名,ユーザ名……）

run-limit に 0 を指定した場合、本制限は無制限となります。

本制限の初期設定値は、無制限です。

操作員権限が必要です。

set queue wait_stageout = *wait_stageout queue-name*

wait_stageout に *queue-name* で指定されたキューのワークフロー内のリクエストおよび、ベアメタルサーバで実行するリクエストのステージアウトにかかる見積もり時間（ステージアウト待ち時間)を秒単位で設定します。
前記リクエストの予定実行終了時刻からこの見積もり時間を空けた時刻以降に後続リクエストをアサインします。ワークフロー内のリクエストの場合、ワークフロー内の後続リクエストのみが適用対象となります。
初期設定値は、0 です。
操作員権限が必要です。

set request baseup_user_definition = *baseup* **processing_pattern** = *pattern request_id*

request_id に対応するリクエストのスケジューリングプライオリティ要素のユーザ定義に対するベースアップ値を *baseup* に指定します。
この値はスケジューリングプライオリティの計算へ使用されます。
request_id の形式は以下のとおりです。
request_number[.batch_server]
request_number
　リクエスト ID のシーケンス番号、パラメトリックリクエストの場合は、その後に[]、サブリクエストの場合は、その後に[n]の形式で指定します。
　n にサブリクエスト番号を指定します。
batch_server
　バッチサーバ名
　batch_server が省略された場合は、JobManipulator サーバが使用している NQSV バッチサーバ名となります。
　パラメトリックリクエスト ID を指定した場合は、パラメトリック内の全てのサブ」リクエストに対して、ユーザ定義ベースアップ値を設定します。
pattern には以下の指定が可能で、指定された処理毎にベースアップ値を設定できます。
assign
　アサイン時
escalation
　エスカレーション時

初期設定値は、0 です。
操作員権限が必要です。

set realtime_scheduling = on |off

リクエスト即時スケジューリングを設定します。on の場合は、スケジューリングインターバルを待たずにリクエストをスケジューリングし、すぐ実行開始します。off の場合は、スケジューリングインターバル毎にリクエストをスケジューリングして実行開始します。

初期設定値は、on です。

操作員権限が必要です。

set realtime_scheduling mode = default | always

リクエスト即時スケジューリングのモードを設定します。default の場合は、JSV の LINKDOWN や LINKUP 等のイベントを処理する時間(イベント時間)を除いた時間帯にリクエストを即時にスケジューリングします。always の場合は、イベント時間を含むすべての時間においてリクエストを即時にスケジューリングします。

初期設定値は default です。
操作員権限が必要です。

set scheduling_method assign_mode = continue |reset

リクエストのアサインモードを設定します。reset の場合は、スケジューリングに時間がかかる場合、一定の時間で処理を打ち切ります。continue の場合は、この処理を打ち切らず、継続して行います。

初期設定値は reset です。
操作員権限が必要です。

set scheduling_method non_scheduled_request_cache = on | off

スケジューリング不可リクエストのキャッシュ機能を有効にするか無効にするかを設定します。
on の場合は、スケジューリングキャッシュ機能が有効になります。
off の場合は、スケジューリングキャッシュ機能が無効になります。

初期設定値は **off** です。
操作員権限が必要です。

set scheduling_method ve_degradation = continue | exclude | auto

VE ノード縮退時のスケジューリング方式を設定します。
continue の場合は最新の VE ノード数に追随してスケジューリングを行います。
exclude の場合は VE ノードが縮退した VI をスケジューリング対象から除外します。
auto の場合は 1 台以上の VE ノードが縮退すると、当該 VI をスケジューリング対象から除外します。縮退した全ての VE ノードが復旧すると、当該 VI が直ちにスケジューリング対象に戻ります。
初期設定値は continue です。
操作員権限が必要です。

set stage-in_margin additional_margin_for_escalation = *additional_margin_for_escalation* | **stage-in_threshold** = *stage_in_threshold* | **first_stage-in_time** = *value*

additional_margin_for_escalation に横方向エスカレーション時のステージング時間の見積もり調整時間を秒単位で設定します。
stage_in_threshold にステージイン開始時間閾値を秒単位で設定します。
value に初回ステージイン時間を秒単位で設定します。
初期設定値は 0 です。
操作員権限が必要です。

set standard_cpu_number = *cpu-number*

cpu-number に標準 CPU 台数を指定します。
標準 CPU 台数の初期設定値は、8 です。
操作員権限が必要です。

set standard_job_number = *job-number*

job-number に標準ジョブ数を指定します。
標準ジョブ数の初期設定値は、10240 です。
操作員権限が必要です。

set standard_memory_size = (*memory-size*)

memory-size に標準メモリサイズを指定します。
標準メモリサイズの初期設定値は、32Gb です。
操作員権限が必要です。

set standard_ve_number = *ve-number*

ve-number に標準 VE 台数を指定します。
標準 VE 台数の初期設定値は、8 です。
操作員権限が必要です。

set treat_unbookable_request = *method*

指定した時間にアサインできない場合、そのリクエストをどのようにするかを設定します。
method には以下の指定が可能です。

delete
　削除する
after
　指定された時間より後ろの最も近い時刻にアサインする

初期設定値は、**after** です。
操作員権限が必要です。

set use_elapstim_unlimited = on | off

資源制限値に elapstim_req が UNLIMITED に設定されているリクエストのスケジューリング可否を設定します。
on に設定した場合、スケジューリングを実施し、**off** に設定した場合はスケジューリングしません。
初期設定値は **off** です。
操作員権限が必要です。

set use_escalation = forward | all | off

エスカレーションの動作を設定します。
forward
　ノード変更なしの前方エスカレーションを行います。
all
　ノード変更の有無にかかわらず前方エスカレーションを行います。
　前方エスカレーションが横方向エスカレーションより優先します。
off
　エスカレーションを行いません。

初期設定値は、**off** です。
操作員権限が必要です。

set use_overtake_priority = on | off *queue-type*

queue-type で指定されたキュー種別内での追い越しを不可にするスケジューリングプライオリティ値を有効にするか無効にするかを設定します。
on の場合は、追い越しを不可にするスケジューリングプライオリティ値を有効にします。
off の場合は無効にします。*queue-type* には以下の指定が可能です。
normal
　normal タイプ
special
　special タイプ

初期設定値は、**off** です。
操作員権限が必要です。

start scheduling

スケジューリングを開始します。
操作員権限が必要です。

stop scheduling

スケジューリングを停止します。
操作員権限が必要です。

stop waiting_retry executionhost = *hostname*

プロビジョニング環境により起動された、仮想マシン（VM）、ベアメタルサーバ、およびコンテナの起動失敗実行ホストの再スケジューリング待ちを解除します。

hostname にはプロビジョニングの実行ホスト名を指定します。

操作員権限が必要です。

stop waiting_retry request = *request_id*

request_id で指定されたリクエストに対して、再スケジューリング待ちを解除します。

request_id の形式は以下のとおりです。

request_number[.*batch_server*]

request_number

リクエスト ID のシーケンス番号、パラメトリックリクエストの場合は、その後に[]、サブリクエストの場合は、その後に[n]の形式で指定します。

n にサブリクエスト番号を指定します。

batch_server

バッチサーバ名

batch_server が省略された場合は、JobManipulator サーバが使用している NQSV バッチサーバ名となります。

パラメトリックリクエスト ID を指定した場合は、パラメトリック内の全てのサブリクエストに対して、再スケジューリング待ちを解除します。

操作員権限が必要です。

suspend request = *request_id*

request_id で指定されたリクエストに対して経過時間を停止するサスペンドの要求を行います。*request_id* の形式は以下のとおりです。

request_number[.*batch_server*]

request_number

リクエスト ID のシーケンス番号、パラメトリックリクエストの場合は、その後に[]、サブリクエストの場合は、その後に[n]の形式で指定します。

n にサブリクエスト番号を指定します。

batch_server

バッチサーバ名

batch_server が省略された場合は、JobManipulator サーバが使用している NQSV バッチサーバ名となります。

パラメトリックリクエスト ID を指定した場合は、パラメトリック内の全てのサブリクエストに対して、経過時間が停止するサスペンドの要求を行います。

サスペンドされたリクエストは、Elapse 経過時間が停止し、**smgr** コマンドの resume request サブコマンドでリジュームされるまで再スケジューリングの対象から外されます。
本サブコマンドでサスペンド可能なリクエスト状態は RUNNING のみです。
本サブコマンドでサスペンドされたリクエストについて、CPU は解放されますが、プロセスは残存しているためメモリは確保したままとなります。したがってサスペンド中に他のリクエストが実行されてもメモリもしくはスワップは十分に確保できる状態にしておく必要があります。他のリクエスト実行時にメモリ不足が発生した場合、ジョブがアボートする原因となります。
本サブコマンドでサスペンドされたリクエストに対して、管理者は **qsig** コマンドでリジュームすることができます。
このときリジュームされたリクエストは直ぐに実行を再開するため、既に実行されているリクエストとリソースの競合が発生する可能性があります。
操作員権限が必要です。

list all [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator、実行ホスト、実行キュー、ノードグループに関する情報を **smgr** のサブコマンドの形に変換して保存します。
出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。
file_name を省略した場合、標準出力に出力します。 また、すでに同名のファイルが存在する場合は、上書きします。

list executionhost [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator が使用しているノードグループに関する情報を **smgr** のサブコマンドの形に変換して保存します。
出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。
file_name を省略した場合、標準出力に出力します。
また、すでに同名のファイルが存在する場合は、上書きします。

list node_group [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator が使用している実行ホストに関する情報を **smgr** のサブコマンドの形に変換して保存します。
出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。
file_name を省略した場合、標準出力に出力します。
また、すでに同名のファイルが存在する場合は、上書きします。

list priority [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator が使用しているスケジューリングプライオリティに関する情報を **smgr** のサブコマンドの形に変換して保存します。

出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。

file_name を省略した場合、標準出力に出力します。

また、すでに同名のファイルが存在する場合は、上書きします。

list queue [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator が使用している実行キューに関する情報を **smgr** のサブコマンドの形に変換して保存します

出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。

file_name を省略した場合、標準出力に出力します。

また、すでに同名のファイルが存在する場合は、上書きします。

list server [file=*file_name*]

file_name で指定したファイルに、現在運用されている JobManipulator に関する情報を **smgr** のサブコマンドの形に変換して保存します。

出力されたファイルを **smgr** に読み込ませることで、JobManipulator 環境の復旧が容易に行えます。

file_name を省略した場合、標準出力に出力します。

また、すでに同名のファイルが存在する場合は、上書きします。

環境変数

JM_PRIV

デフォルトのコマンド実行時のアクセス権

JM_PRIV_MGR 管理者権限

JM_PRIV_OPE 操作員権限

JM_PRIV_GMGR グループ管理者権限

JM_PRIV_SPU 特別利用者権限

JM_PRIV_USR 一般利用者権限

JM_SCHED_ID

デフォルトの JobManipulator スケジューラ ID

関連項目

sstat(1).

ファイル

/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf

特記事項

Copyright (c) 1992, 1993 The Regents of the University of California.
All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of
Cornell University.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are
met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
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)

名前

sushare -- JobManipulator ユーザシェア管理

形式

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**

機能説明

1. **-D** オプションを指定、もしくはコマンド動作を示すオプション(C,U,S,L,I,M)を指定しない場合
   user-share-config-file に指定されたユーザシェア設定ファイルを読み込み、各ユーザのユーザシェア値を表示します。
   user-share-config-file は、絶対パス名で指定してください。
   user-share-config-file が省略された場合は、/etc/opt/nec/nqsv/jm_sharedb.conf が対象となります。
   ユーザシェアファイルの修正方法については以下を参照してください。
   NQSV 利用者の手引[JobManipulator 編]
   第 3 章 運用管理 3.1.3.2 スケジューリングプライオリティ計算式

2. **-C** オプションを指定した場合

   user-share-config-file に指定されたユーザシェア設定ファイルの整合性チェックを行います。
   user-share-config-file は、絶対パス名で指定してください。
   user-share-config-file が省略された場合は、/etc/opt/nec/nqsv/jm_sharedb.conf が対象となります。

エラーメッセージと意味は、以下の通りです。

エラーメッセージ	意味
Double definition	ユーザを二重定義（複数のグループで定義）
No entry	グループにユーザを定義していません。
Belong to no group	グループが TOP_GROUP に定義していません。

3. **-U** オプションを指定した場合

user-share-config-file に指定されたユーザシェア設定ファイルのユーザシェア値を（スケジューラ）のスケジューリングプライオリティ計算に使用するように登録し

ます。
user-share-config-file は、絶対パス名で指定してください。
user-share-config-file が省略された場合は、/etc/opt/nec/nqsv/jm_sharedb.conf が対象となります。 以下のオプションが指定できます。

-h *server_host*

JobManipulator が実行されるサーバホストを指定します。
本オプションが指定されない場合、デフォルトの JobManipulator サーバホスト(/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定したホスト名)を使用します。
デフォルトの JobManipulator サーバホストがない場合は、ローカルホストとなります。

-s scheduler_id

JobManipulator のスケジューラ ID を指定します。本オプションが指定されない場合、スケジューラ ID は、環境変数 **JM_SCHED_ID** で指定されたスケジューラ ID となります。環境変数が設定されていない場合は、
/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定した JobManipulator スケジューラ ID を使用します。いずれの設定もされていない場合は 0 となります。

4. **-S** オプションを指定した場合

user-name[:Account-code]に指定されたユーザの使用実績値（ShareDB のマージ機能を使用した場合は"マージ後"の実績値）を表示します。user-name[:Account-code]が省略された場合は、全ユーザ使用実績値及びグループ毎の合計使用実績値を出力します。-f オプションを指定しなかった場合、グループ名の先頭にはグループ毎の合計使用実績を示す*が付加されます。

以下のオプションが指定できます。

-b 値を単位変換せずに表示します。

-f　　詳細情報を表示します。

-h *server_host*

JobManipulator が実行されるサーバホストを指定します。本オプションが指定されない場合、デフォルトの JobManipulator サーバホスト (/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定したホスト名)を使用します。デフォルトの JobManipulator サーバホストがない場合は、ローカルホストとなります。

-s *scheduler_id*

JobManipulator のスケジューラ ID を指定します。本オプションが指定されない場合、スケジューラ ID は、環境変数 JM_SCHED_ID で指定されたスケジューラ ID となります。環境変数が設定されていない場合は、/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定した JobManipulator スケジューラ ID を使用します。いずれの設定もされていない場合は 0 となります。

表示される以下の値はユーザ毎の使用実績値（半減期による減衰も含む）を蓄積した値です。

PU_cpunum　　: 使用 CPU 台数 * 経過時間
PU_venum　　: 使用 VE 台数 * 経過時間
PU_memsz　　: 使用メモリ量 * 経過時間
PU_elapstim　　: 経過時間
PU_reqpri　　: リクエストプライオリティ * 経過時間

表示される値に付加されている "B","K","M"等は単位を表しています。

- PU_cpunum、PU_venum、 PU_reqpri に表示される単位について

  "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)

- PU_memsz に表示される単位について

  "B" = バイト
  "K" = キロバイト（1K = 1024B）

"M" = メガバイト（1M = 1024K）
"G" = ギガバイト（1G = 1024M）
"T" = テラバイト（1T = 1024G）
"P" = ペタバイト（1P = 1024T）
"E" = イクサバイト（1E = 1024P）

5. **-L** オプションを指定した場合

ローカルの使用実績を表示します。

6. **-I** オプションを指定した場合

user-name[:*Account-code*]に指定されたユーザの使用実績値を変更します。*user-name*[:*Account-code*]が省略された場合は、全ユーザが対象となります。
以下のオプションが指定できます。

-c *pastusage_cpunum*
使用実績値の CPU 台数を *pastusage_cpunum* で指定された値に変更します。

-v *pastusage_venum*
使用実績値の VE 台数を *pastusage_venum* で指定された値に変更します。

-e *pastusage_elapstim*
使用実績値の経過時間を *pastusage_elapstim* で指定された値に変更します。

-h *server_host*
JobManipulator が実行されるサーバホストを指定します。本オプションが指定されない場合、デフォルトの JobManipulator サーバホスト (/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定したホスト名)を使用します。デフォルトの JobManipulator サーバホストがない場合は、ローカルホストとなります。

-m *pastusage_memsz*
使用実績値のメモリサイズを *pastusage_memsz* で指定された値に変更します。

-r *pastusage_reqpri*
使用実績値のリクエストプライオリティを *pastusage_reqpri* で指定された値に変更します。

-s *scheduler_id*
JobManipulator のスケジューラ ID を指定します。本オプションが指定されない場合、スケジューラ ID は、環境変数 **JM_SCHED_ID** で指定されたスケジューラ ID となります。環境変数が設定されていない場合は、/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf　内に指定した JobManipulator スケジューラ ID を使用します。いずれの設定もされていない場合は 0 となります

-z　使用実績の全要素をゼロクリアします。

7. **-M** オプションを指定した場合

config-file で指定された内容にしたがって ShareDB ファイルのマージを実行します。
config-file が省略された場合は、/etc/opt/nec/nqsv/jm_merge_sharedb.conf が対象となります。

本オプションの実行には、操作員権限が必要です。
本オプションでは以下のオプションが指定できます。

-l [*logfile_name*]
logfile_name で指定されたファイルパスにマージ処理の詳細結果を出力します。
logfile_name を省略した場合、カレントディレクトリの nqs_jmd_sharedb_merge.log に出力されます。 **-l** と *logfile_name* の間に空白文字は入れないで下さい。

8. **--help** オプションを指定した場合

sushare の使い方を表示して終了します。

9. **--version** オプションを指定した場合

コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

共通で以下のオプションが指定できます。

-P *privilege*
コマンド実行時のアクセス権を指定します。*privilege* には、次の指定ができます。

m	管理者権限となります。
o	操作員権限となります。
s	特別利用者権限となります。
u	一般利用者権限となります。

アクセス権は、管理者権限がもっとも高く、一般利用者権限が最低となります。
本オプションが指定されない場合、コマンド実行時のアクセス権は、環境変数
JM_PRIV で指定されたアクセス権となります。
環境変数が設定されていない場合は、一般利用者権限となります。

-V

コマンドのバージョンおよび使用している JobManipulator API のバージョンを表示して終了します。

アクセス権

sushare コマンドを使用するには操作員権限以上が必要です。

環境変数

JM_PRIV
　デフォルトのコマンド実行時のアクセス権
　　JM_PRIV_MGR　　管理者権限
　　JM_PRIV_OPE　　操作員権限
　　JM_PRIV_SPU　　特別利用者権限
　　JM_PRIV_USR　　一般利用者権限

JM_SCHED_ID
　デフォルトの JobManipulator スケジューラ ID

ファイル

/etc/opt/nec/nqsv/jm_sharedb.conf
/etc/opt/nec/nqsv/nqs_jmd_cmdapi.conf
/etc/opt/nec/nqsv/jm_merge_sharedb.conf

付録 A　発行履歴

A.1　発行履歴一覧表

2018 年　2 月	初版
2023 年　6 月	第 20 版
2023 年　9 月	第 21 版
2024 年　3 月	第 22 版
2025 年　7 月	第 23 版
2026 年　7 月	第 24 版

A.2　追加・変更点詳細

第 20 版

- smgr(1M)コマンドのサブコマンドset queue network_topology min_nwgroupを修正
- qlogin(1), qrsh(1), qsub(1)コマンドの-vオプションによって変更できない環境変数を修正

第 21 版

- qlogin(1), qrsh(1),qsub(1)コマンドの-U,-vおよび-wオプションに関する説明を充実
- qsub(1)コマンドの-Cオプションに関する説明を充実

第 22 版

- scacctjob(1), scacctreq(1)コマンドの--long-request-nameオプションに関する説明を追加

第 23 版

- qstat(1)コマンドの--venodeの形式を修正
- qmgr(1M)コマンドの一部のサブコマンドのアクセス権を訂正
- qmgr(1M)コマンドの一部のサブコマンドの指定可能な値の範囲を訂正

第 24 版

- scacctjob(1), scacctreq(1) コマンドにGPUアカウンティング機能に伴い追加された出力項目を追加
- sceditacct(1M)コマンドの退避対象DBファイルを追加
- qmgr(1M)コマンドの一部のサブコマンドの指定可能な値の範囲を訂正

索　引

B

BMC vi
budgetcheck 2
budgetedit 201

H

HCA vi

I

IB vi

M

MPI vi

N

NIC vi
nmapmgr 204
nqs_ntfr 206

Q

qalter 4
qattach 14
qcat 15
qcmdconf 18
qdel 21
qdetach 23
qhold 25
qlogin 27
qmgr 208
qmove 40
qrerun 43
qrls 45
qrsh 47
qsig 60
qstat 62
qsub 103
qwait 132
qwait2 136

R

racctreq 150
racctrsv 161
rbudgetcheck 2

S

sbuedit 313
scacctreq 150
scacctrsv 161
sceditacct 317
scsumacct 320
scturnacct 322
smgr 323
sstat 166
sushare 367

V

VE vi
VI クラスター vi

W

wdel 194
wstart 196
wstat 198

あ

アサインマップ情報 190

き

キュー情報 183, 184

さ

サーバ情報 177

し

実行ホスト情報 187, 188

す

スケジューリングプライオリティ情報 182

と

同時実行リクエスト数制限 177, 183

へ

ページャ機能 325
ベクトルエンジン vi
ベクトルホスト vi

ゆ

ユーザランク詳細 174

よ

予約区間情報表示 190, 191

り

リクエストスケジューリング情報 173
リクエストスケジューリング情報（詳細）
.....174

わ

割り込まれたリクエストを中断させる方法
.....340

NEC Network Queuing System V (NQSV)
利用の手引［リファレンス編］

2026年 7月 第24版

日本電気株式会社

東京都港区芝五丁目７番１号

TEL(03)3454-1111（大代表）

© NEC Corporation 2018
日本電気株式会社の許可なく複製・改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。