Perl API Reference

qb::block

Purpose	Sets job state to blocked.	
Prototype	qb::block(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Notes		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;qb::block(@ids)
```

qb::bottom

Purpose	Moves jobs to end of execution order queue.	
Prototype	qb::bottom(*ids*)	
Parameters	ids	a list of job or subjob ids.
Result		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::bottom(@ids)
```

qb::genchunks

Purpose	Generates a work agenda based on an input frame specification divided into fixed width chunks.	
Prototype	qb::genchunks(*chunksize*, *range*)	
Parameters	chunksize	number of frames in a single chunk
	range	frame range format string. See qb::rangesplit() for range format.
Result	Reference to a work agenda hash.	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

my $frames = qb::genchunks(5, "1-100");
```

qb::genframes

Purpose	Generates a work agenda based on an frame specification string.	
Prototype	qb::genframes(*range*)	
Parameters	range	Frame range format string. See qb::rangesplit() for range format.
Result	Reference to a work agenda hash.	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

my $frames = qb::genframes("1-100");
```

qb::hist

Purpose	Returns a list of history objects corresponding to the job query.	
Prototype	qb::hist(*ids*)	
Parameters	ids	Job or subjob ids.
Result	Reference to a array of hash references containing history information: "comment": comment string "stamp": timestamp "jobid": job ID "subid": subjob ID	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::hist(@ids)
```

qb::hostinfo

Purpose	Returns information from list of hosts.	
Prototype	qb::hostinfo(*query*)	
Parameters	query	Hash containing host query information

Results	Array of hash references containing information about hosts meeting query: "name": host name "address": host IP address "state": host state "cluster": host cluster membership "groups": host group membership "stats": host statistics "properties": Worker properties for host "resources": Worker resources for host "restrictions": host restrictions	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::hostinfo(%query)
```

qb::interrupt

Purpose	Forces running jobs back to pending state immediately.	
Prototype	qb::interrupt(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::interrupt(@ids)
```

qb::jobinfo

Purpose	Generates a list of job objects matching query filter.	
Prototype	qb::jobinfo(*query*)	
Parameters	query	Hash containing job query parameters.

Result	Reference to an array of hashes containing information regarding jobs that meet the query (all jobs if no query): "prototype": job type "ID": job ID "priority": job priority "user": job owner "label": job label "account": job accounting "pid": process ID "pgrp": process group "cpus": cpus "reservations": job reservations "requirements": job requirements "restrictions": job restrictions "cluster": job cluster "hosts": job hosts "groups": host groups "name": job name "package": job package data hash reference "data": job package data string "status": current job status	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::jobinfo(%query)
```

qb::joborder

Purpose	Returns a list of jobs eligible to run on specified hosts.	
Prototype	qb::joborder(*host*)	
Parameters	host	host name.
Results	Reference to an array of hashes containing job information for jobs scheduled to run on host: "prototype": job type "ID": job ID "priority": job priority "user": job owner "label": job label "account": job accounting "pid": process ID "pgrp": process group "cpus": cpus "reservations": job reservations "requirements": job requirements "restrictions": job restrictions "cluster": job cluster "hosts": job hosts "groups": host groups "name": job name "package": job package data hash reference "data": job package data string "status": current job status	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::joborder($host)
```

qb::kill

Purpose	Kills jobs.	
Prototype	qb::kill(*ids*)	
Parameters	ids	list of job or subjob ids.
Results		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::kill(@ids)
```

qb::migrate

Purpose	Interrupt a running job and force to run on a different host.	
Prototype	qb::migrate(*ids*)	
Parameters	ids	a list of job or subjob ids.
Result		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::migrate(@ids)
```

qb::modify

Purpose	modifies job parameters	
Prototype	qb::modify(*parameters*, *ids*)	
Parameters	parameters	Hash containing job parameters and new values.

	ids	list of job or subjob ids to be modified
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;qb::modify({ priority => 1, name => "hello world" }, 1002);
```

qb::preempt

Purpose	Forces running jobs back to pending state after agenda item is completed.	
Prototype	qb::preempt(*ids*)	
Parameters	ids	list of job or subjob ids.
Results		
Notes	Will release host gracefully if Job Type supports an agenda.	

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::preempt(@ids)
```

qb::rangechunk

Purpose	Converts an frame range format into an array of frame ranges of a specified length.	
Prototype	qb::rangechunk(*chunksize*, *range*)	
Parameters	chunksize	Number of frames in a single chunk
	range	Frame range format string. See qb::rangesplit() for range format
Result	List containing the individual frames in mode order.	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::rangechunk($chunksize, $range)
```

qb::rangejoin

Purpose	Converts a list of frames into frame range format string.	
Prototype	qb::rangejoin(*frames*)	
Parameters	frames	List of frames.
Result	Frame range format string corresponding to the frame list	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
my $range = qb::rangejoin(1,2,3,4,5);result: $range = "1-5";
my $range = qb::rangejoin(1,3,5,7,9,11,13);
result: $range = "1-13x2";
```

qb::rangeorder

Purpose	Takes an input range and converts it into an array of individual numbers sorted/ordered in the method specified.	
Prototype	qb::rangeorder(*mode*, *range*)	
Parameters	mode	Sort order mode.Valid modes: 1. binary, 2. reverse 3. rawbinary 4. ascend 5. descend
	range	Frame range format string. See **qb::rangesplit ()** for range format
Result	List containing the individual frames in mode order.	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;qb::rangeorder($mode, $range)
```

qb::rangesplit

Purpose	Takes an input range and converts it into an array of individual numbers.	
Prototype	qb::rangesplit(*range*)	
Parameters	range	Frame range format string.

Results	list of individual frame numbers	
Comments	Range Format: <start>~~<end>x<step>[,<start>~~<end>x<step>...] Ex. 1-10x2,12,20	

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

my @frame_numbers = qb::rangesplit("1-100");

result: @frame_numbers = (1,2,3,4..98,99,100);

my @frame_numbers = qb::rangesplit("1-100x10");

result: @frame_numbers = (1,11,21,31,41,51,61,71,81,91);

my @frame_numbers = qb::rangesplit("10-10");result: @frame_numbers = (-10,-9,-8,-7..6,7,8,9,10);my
@frame_numbers = qb::rangesplit("-10-5");

result: @frame_numbers = (-10,-9,-8,-7,-6,-5);

my @frame_numbers = qb::rangesplit("1-5x2,10-12");

result: @frame_numbers = (1,3,5,10,11,12);
```

qb::remove

Purpose	Removes jobs from the Supervisor database cache.	
Prototype	qb::remove(*ids*)	
Parameters	ids	list of job or subjob ids.
Results		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;qb::remove(@ids)
```

qb::requeue

Purpose	Resets a failed, complete or killed job back to a initial blocked state.	
Prototype	qb::requeue(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;qb::requeue(@ids)
```

qb::resource

Purpose	Queries Supervisor for state of resources.	
Prototype	qbresource(*resources*)	
Parameters	resources	Hash describing resource keys and values
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qbresource(%resources)
```

qb::resume

Purpose	resumes suspended jobs.	
Prototype	qb::resume(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;qb::resume(ids)
```

qb::retry

Purpose	Resets failed, complete, or killed jobs back to the pending state.	
Prototype	qb::retry(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Notes		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;qb::retry(@ids)
```

qb::stderr

Purpose	Retrieves job STDERR log file output.	
Prototype	qb::stderr(*ids*)	
Parameters	ids	list of job or subjob ids
Results	Array of hash references referring to the STDERR logs: "data": log file contents "jobid": job ID "subid": subjob ID	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;qb::stderr(@ids)
```

qb::stdout

Purpose	Retrieves job STDOUT log file output.	
Prototype	qb::stderr(*ids*)	
Parameters	ids	list of job or subjob ids
Results	Array of hash references referring to the STDOUT logs.	
Comments	Hash contains keys: "data": log file contents "subid": subjob ID "jobid": job ID	

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::stderr(@ids)
```

qb::submit

Purpose	Submits a list of jobs to be dispatched by the Supervisor.	
Prototype	qb::submit(*job*)	
Parameters	job	Hash reference containing the job parameters. Must contain at minimum a $job{prototype} value containing the Job Type to execute.
Results	$_ is a pointer to the job hash.$_->{ID} is the ID of the submitted job.	
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

my $job = { "name" => "job name", "priority" => "12433", "cluster" => "/project/rnd", "requirements" =>
"host.name == qb003 \ and host.os eq Linux", "prototype" => "cmdline", "package" => {"cmdline" => "sleep
100"}};

qb::submit($job)
```

qb::submitcallback

Purpose	Submit a job callback to be executed on certain events.	
Prototype	qb::submit(*callback*)	
Parameters	callback	Hash containing the callback parameters.
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";

use qb;

qb::submit(%callback)
```

qb::suspend

Purpose	Sends the SUSPEND signal to running jobs	
Prototype	qb::suspend(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::suspend(@ids)
```

qb::top

Purpose	Moves jobs to the head of execution order queue.	
Prototype	qb::top(*ids*)	
Parameters	ids	a list of job or subjob ids.
Result		
Comments		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::top(@ids)
```

qb::unblock

Purpose	Unblocks jobs so they can begin executing when hosts become available.	
Prototype	qb::unblock(*ids*)	
Parameters	ids	a list of job or subjob ids.
Results		
Notes		

Example

```
use lib "$ENV{QBDIR}/api/perl";
use qb;
qb::unblock(@ids)
```