

__SimpleCmd__AdvancedJobControl

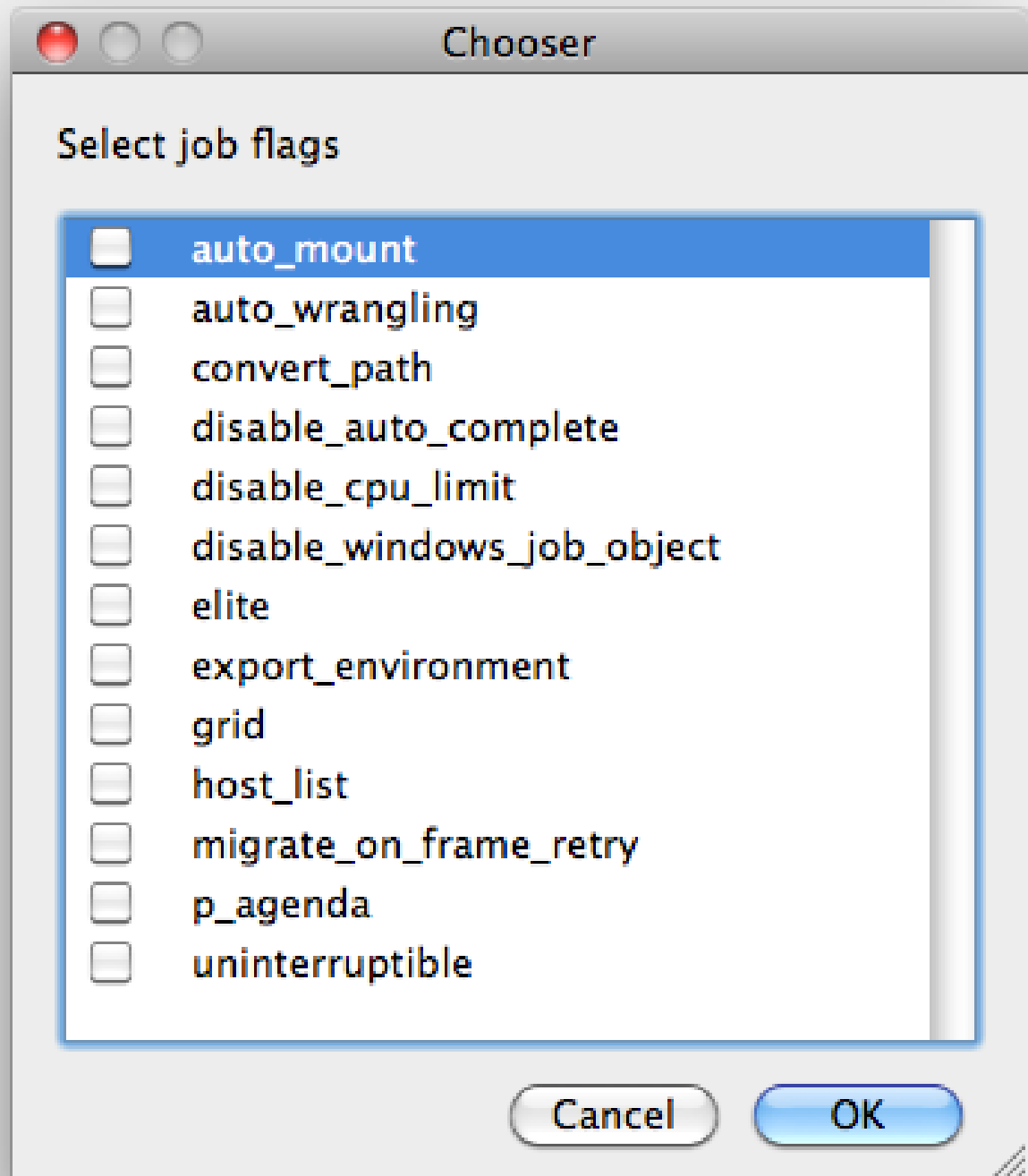
Qube Advanced Job Control

Flags	<input type="text"/>	<input type="button" value="Browse"/>
Dependency	<input type="text"/>	<input type="button" value="Add"/>
Email (job complete)	<input type="checkbox"/>	<input type="text" value="briank"/>
Email (failed frames)	<input type="checkbox"/>	<input type="text" value="briank"/>
Blocked	<input type="checkbox"/>	
Stderr->Stdout	<input type="checkbox"/>	
Job Label	<input type="text"/>	
Job Kind	<input type="text"/>	
Process Group	<input type="text"/>	
Retry Frame/Instance	<input type="text" value="0"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
Retry Work Delay	<input type="text" value="0"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
Subjob Timeout	<input type="text" value="-1"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
Frame Timeout	<input type="text" value="-1"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>

▼ [Click here for details...](#)

Flags

List of submission flag strings (comma separated). Click 'Browse' to choose required job flags.

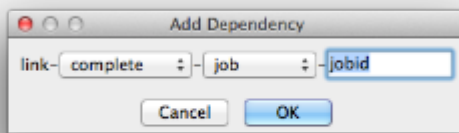


See [this page](#) for a full explanation of flag meanings

Dependency

Wait for specified jobs to complete before starting this job

(comma-separated). Click 'Add' to create dependent jobs.



You can link jobs to each other in several ways:

- "complete" means only start this job after designated job completes
- "failed" means only start this job if the designated job fails
- "killed" means only start this job if the designated job has been killed
- "done" means start this job if the designated job is killed/failed/complete

The second menu chooses between "job" (the entire set of frames) and "work" (typically a frame). So to link frame 1 of one job to frame 1 of a second, job, you would choose "work" in this menu. If you want to wait for all the frames of one job to complete before starting a second, then choose "job". The other option, "subjob", refers to the instance of a job. This is much less common, but means that, for example, the instance of Maya that was running frames has completed.

For a complete description on how to define complex dependencies between jobs or frames, please refer to the [Callbacks](#) section of the Developers Guide.

Email (job complete)

Send email on job completion (success or failure). Sends mail to the designated user.

Email (failed frames)

Sends mail to the designated user if frames fail.

Blocked

Set initial state of job to "blocked".

Stderr->Stdout

Redirect and consolidate the job stderr stream to the stdout stream. Enable this if you would like to combine your logs into one stream.

Job Label

Optional label to identify the job. Must be unique within a Job Process Group. This is most useful for submitting sets of dependent jobs, where you don't know in advance the job IDs to depend on, but you do know the labels.

Job Kind

Arbitrary typing information that can be used to identify the job. It is commonly used to make sure only one of this "kind" of job runs on a worker at the same time by setting the job's requirements to

include "not (job.kind in host.duty.kind)". See [How to restrict a host to only one instance of a given kind of job, but still allow other jobs](#)

Process Group

Job Process Group for logically organizing dependent jobs. Defaults to the jobid. Combination of "label" and "Process Group" must be unique for a job. See [Process group labels](#)

Retry Frame/Instance

Number of times to retry a failed frame/job instance. The default value of -1 means don't retry.

Retry Work Delay

Number of seconds between retries.

Subjob Timeout

Kill the subjob process if running for the specified time (in seconds). Value of -1 means disabled. Use this if the acceptable instance/subjob spawn time is known.

Frame Timeout

Kill the agenda/frame if running for the specified time (in seconds). Value of -1 means disabled. Use this if you know how long frames should take, so that you can automatically kill those running long.