Job Dependencies

Job Parents

Qube! allows parent/child relationships between jobs. This does not imply dependency. With proper use of callbacks, it can be used to set up complex abstract dependencies, without the use of process groups. Qube! allows the user to specify the parent ID through the API or on the command line.

Example

```
% qbsub --pid 12314 Render myscenefile.ma
```

Job Dependencies

Suppose you want to link up two or more jobs so that when one finishes, the next one will run. Qube! provides two methods for performing this task:

1. --waitfor option in qbsub
2. job callbacks with the API

Using the --waitfor option with qbsub is by far the easier of the two approaches; however, it is not without some limitations in flexibility. The technique is straightforward:

1. Submit the first job with qbsub
2. Collect the job ID, say *id*
3. Submit the dependent job with qbsub --waitfor *id*
4. Collect the job ID if more dependent jobs are required
5. This can also be accomplished using the job dependency field feature. A job will wait until the specified condition is reached. The format of the dependency field is exactly the format used in the callback trigger field (See Triggers)

What happens

The first job is submitted normally. If the second job is submitted with a --waitfor option, it will be submitted in the blocked state. When the first job finishes, it will unblock and run.

Example

Using the waitfor option:

```
% qbsub --name jobA sleep 60  (this creates job 220)
% qbsub --name jobB --waitfor 220 sleep 60
```

Using the dependency option:

```
% qbsub --name jobA sleep 60  (this creates job 220)
% qbsub --name jobB --dependency link-complete-job-220 sleep 60
```

Process Groups

Qube! process groups are created upon job submission. Jobs are associated in the same process group by submitting them all in the same qb::submit API call, or they may be connected by specifying the "pid" number for the job during submission.

Example

```
% qbsub --pgrp 1002 sleep 100
```

Wait On

For developers who prefer to use Qube! as a remote procedure mechanism for executing commands and waiting for them to complete before continuing with the execution of their scripts/tools, the `qbsub` command allows a user to submit a job and expect `qbsub` to wait until the job completes, fails or is killed. This is very easily done by using the `--waiton` option.

Example

The command will block until the submitted job finishes:

```
% qbsub --waiton sleep 10
```

Example Script

```
#!/bin/sh
# Create new rib files
mtor etc. etc. etc..
#Render the files
qbsub –waiton prman myfiles.rib
# Cleanup
rm myfiles.rib
```