

_jobLabels

Qube introduces the concept of job **labeling**. A job label is a separate field in a Qube job which is used to help other jobs **refer to that job by name rather than its job ID**.

When designing a job dependency graph, developers were previously forced to submit the jobs in order of precedence, collecting job ids and using them to initialize the child jobs. This technique is messy and takes a significant amount of development to implement. It also limits the dependencies to a directed graph, and will not lend itself to a feedback loop job easily.

Another alternative is to use the job's name to identify the job's dependency relationship. This method doesn't work well because a user is then committed to a strict naming convention when submitting to the farm.

To solve this, Qube uses its **process group** job attribute in combination with the job's label. The only prerequisite is that the jobs be submitted with the same process group ID. All jobs submitted with the same `qb.submit()` call are automatically joined into a new pgrp; each has the same process group ID, which is the job ID of the first job submitted (also known as the **pgrp leader**).



Job Labels **must be unique within the process group**.

The Qube Supervisor automatically enforces the uniqueness requirement and will not allow duplicate job labels to be submitted, instead the submission will be rejected

During submission, a developer *may* link jobs to the same process group by collecting the lead job's process group ID and then using that to submit the successive jobs, setting each job's **pgrp** value. A simpler method is to submit all the jobs under the same API submit call, which automatically attaches all jobs to the same process group.

This process group/label system solves 2 problems:

1. The developer isn't forced to collect the job ids.
2. The developer isn't required to use a naming convention for their jobs.

The system also offers several major benefits:

1. Resubmission, cloning and storage of a process group are simpler.
2. Feedback loop job relationships are made possible.