

Job Structure

Synopsis

All jobs in Qube! have a consistent structure. It is through this structure the Supervisor is able to communicate to the Worker the appropriate instructions for executing a job.

Job Type

The job's type is the mechanism by which Qube! indicates to the Worker what it should do with the job. For example, a job with type "command line" will have discrete strings for each agenda item that will be run at a command prompt (e.g. "dir", "ls", "cinema4D -frame 1 5 1 -render /path/to/file.c4d"). A job with type "maya" will, instead, store a discrete set of instructions for Maya for each agenda item that will be run at a mel prompt. In the case of the former, the application is launched when the agenda item starts and then exits when the agenda item is complete. In the case of the latter, the application is launched when the instance starts, but then does not exit until all agenda items in the job have been completed (either by this instance and/or others).

Job ID

Once a job has been submitted and accepted by the Supervisor, it is assigned an ID. The ID is unique to the job, and serves two purposes: it helps the Supervisor keep track of each job individually, and without any other priority information, the Supervisor can dispatch jobs in chronological ("first-come, first-served") order based upon the ID.

Job Parameters

Job parameters act as instructions to the Qube! Supervisor to help it determine how best to dispatch the job to the Workers. The parameters are simply the values used in the submission dialog (or through the API or command line).

Job Package

The job package contains parameters that are specific to the application that is being rendered. For example, with a cmdline job, the package will contain the command to be executed and possibly the frame range. For a maya job, the package will contain the frame range, and things like the path to the maya scene file, the output path (if specified), the thread count, etc. Every job will have a package, but every package may be different, both in number and type of variables.

Job Agenda

The job agenda contains a number of discrete items – often frames – that are transmitted by the Supervisor to the Worker on an "as needed" basis. An agenda could be considered synonymous with a task.

Job Callbacks

Callbacks are one of the most powerful features of Qube!, enabling it to perform well beyond other similar render farm management systems. Any event that takes place in the system can trigger the Supervisor to perform a series of actions; some are internal like activating a dormant job, others are external like sending out an email message or executing a python script.

Instances

In reality, a job is a large data structure that contains all the information the Supervisor needs to monitor, control, and dispatch a task for a remote host. The actual execution of that task is performed by the Worker by what we call an *instance* (previously, these were called *subjobs*). In other words, a worker runs one or more job instances in parallel and each instance runs one or more agenda items (typically frames) in series.