Glossary of Terms

Qube! Related Terms

Agenda	Agenda items (or Frames) refers to the granular list of items to be processed.
AppFinder	AppFinder Jobs will "find" a particular version of a 3rd-party application on the worker when the job runs. The submitting user is not required to know that application's installation path on the remote worker and •since the application does not include a hard-coded path to the 3rd-party application, the job is able to run across machines of differing operating systems at the same time.
ArtistView	ArtistView GUI is designed to be a more artist-centric view of Qube! that delivers fast feedback on your rendering pipeline.
AutoWrangling	Some of the most common render wrangling tasks are handled automatically by the Supervisor through a set of global parameters. Qube! has added built-in logic that detects faulty jobs and workers.
Callbacks	Job Callbacks allow Qube events to trigger many actions including unblocking jobs, emailing the user, or running custom scripts.
Chunk	When the application startup and scene load time can contribute significantly to the processing time of a single piece of work, it becomes advantageous to send more than 1 piece of work to be processed each time the application is started. This is a “chunk”.
Client	Each user application that interacts with the Qube Supervisor is referred to as a Client (artist_user machines). These interfaces include the QubeGUI, commandline, and some job-specific in-application interfaces.
Cluster	The clusters in a Qube farm are laid out in a tree-like structure. In the same way that folders in a directory tree contain files and other folders, clusters can contain other clusters. Clusters can also contain Workers and jobs. This is the key to Qube’s priority scheme.
Cmdline Job	A commandline job is the simplest type of job; you submit a command-line that is run on a machine in the farm. The command is run exactly as entered, no part of it is substituted at the time the job runs. If you submit a cmdline job with multiple instances, the identical command-line is run on each instance.
Cmdrange Job	A command range job has Agenda items over a frame range.
Dependencies	Jobs can have *dependencies* they require to be satisfied in order to work properly, such as the correct versions of software. Or *dependencies* can mean having one (or more) job's execution dependent on one (or more) other jobs finishing first.
Dynamic Allocation	Dynamic Allocation reduces application startup and scene load overhead to the absolute minimum while automatically load balancing the work to be done across different speed workers. With Dynamic Allocation, the application is only started once on each worker for the duration of the job, and work is sent “on demand” in the smallest units possible.
Hostid	The hostid is the MAC address of the Supervisor machine.
Instance	Instances or Job Processes ("Subjobs" pre 6.4) are numbered 0 through n, where n is the number of processes that the job can run at the same time. Therefore if you want to have the farm render 10 frames at a time, the number of Instances for that job should be 10.
Job	A Job is what's submitted by a client to the supervisor. It's a little bundle of information that includes all the information the worker will need to execute the job - what program(s) to run, where the source files are, who's running it, how many cpus it wants, what OS it can run under, all that kind of stuff.
Job ID	A unique number assigned to each job by the Supervisor. It is used to keep track of jobs.
Job Instance	(known as *subjobs* in older versions of Qube) When you wish to run a job on more than 1 machine, you submit a job with multiple instances; each instance is a copy of the same job, but usually each instance will be processing a different frame. A job instance occupies 1 or more **job slots** on a worker, depending on the job's **reservation** value.
Job Package	A generic data structure, included in the job object, that can be accessed by the Worker's backend job module and used by that module to set up the job.
Job Slots	The Job Slots (or Process Slots) for a Worker determine how many processes a given Worker can run at a given time. The Job Slots have no direct relation to the number of processor cores on a given machine. It is often desirable when using a multithreaded renderer like Maya to have the Worker run 1 Instance (or process) at a time.
Job Types	Qube has the unique ability to allow a developer to create not only job submission front ends, but also develop the back end execution modules. The coupling of a front end interface and a back end implementation is called a Job Type. Qube comes with a built- in Job Type for executing traditional command lines and scripts. It also comes with a full API for a number of common scripting languages if you decide you need more control than is provided by the basic command Job Type.

MobileView	Qube! MobileView is a light-weight webserver that serves up mobile-enhanced web pages from any machine in your facility Note: Touch devices required for interaction.
Priority	Jobs have a priority value that tells the Supervisor which jobs are more important and which are less. Priority runs from 1 to 9999, with 1 being the highest (most important) priority.
QBLocker	QBLocker gives artists the ability to control how much of their workstation is available to the render farm and shows which jobs are running on their workstation at any point in time.
Qube!	Qube! is the leading render farm management system for digital media creation. Built on a client-server architecture, Qube! generates, prioritizes, dispatches and oversees render and build jobs.
Qubeproxy	By default, each Qube Worker is configured to run in what we call “proxy” mode. This means the Worker will attempt to execute the job as a proxy user, rather than the user that submitted the job (“user” mode). When Qube is installed, a local user called “qubeproxy” is installed, and the Worker is pre-configured to run in proxy mode, and to designate qubeproxy as the proxy user.
Render Farm	A render farm is high performance computer system, e.g. a computer cluster, built to render computer-generated imagery (CGI), typically for film and television visual effects.
Render Wrangling	There are a number of manual tasks typically performed when overseeing jobs on the renderfarm and the renderfarm itself. The render wrangler performs these tasks.
Requirements	Jobs can (and usually do) have requirements that they need satisfied by a Worker before they can run on it. These can include the operating system, a number of processors, an amount of RAM, specific application licenses, or any of a number of things.
SimpleCmd:	The SimpleCmd framework provides an easy way to add new submission interfaces to the QubeGUI or tailor existing ones.
Supervisor	The Supervisor (server node) is the central server daemon and is responsible for all functions involving the submission, dispatch, monitoring, and control of jobs in the system.
Worker	Workers (render nodes) are daemons (or services in Windows) running on each execution host that is configured to execute jobs under the Qube management system. There can be several jobs running on a single Worker host.
WranglerView	WranglerView GUI is the newly renamed Qube! GUI. This frontend is more focused at advancedl users and render wranglers/admins. It gives access to user permissions, license installation, Supervisor testing and repair, charts and control of user jobs.