System-Wide Resource Tracking

Qube! offers automatic resource tracking to monitor and control the farm's total usage of finite compute resources. For example, it may be used to track resources such as floating software licenses, or the number of simultaneous jobs accessing a particular file server.

Jobs are submitted with their **reservations** property specifying the required resources, and the supervisor will review it when making dispatch decisions.

Qube! can manage the following kinds of resources:

- **Global** – Resources available and managed over the whole render farm system.
- **License** – Resources that are controlled and managed external to Qube but are consumed when jobs execute.
- **Per-Host Global** – Resources available and managed over the whole render farm system, where a single resource is shared by multiple subjobs running on the same worker.
- **Per-Host License** – Resources that are controlled and managed external to Qube but are consumed when jobs execute. Each resource is shared by multiple subjobs running on the same worker.
- **Scoped** – Resources that are managed over the whole render farm system, but only available to specific hosts.

Global vs License Resources

The difference between "global" and "license" is whether or not any entity external to Qube! is also drawing from the same license pool as Qube!. If the license resources are only in use by Qube!, and are never checked out by users, other queueing systems or automated pipeline processes running outside of Qube!, then you can use the "global" naming scheme, and you do **not** need to implement polling of the license server with the periodic updating of Qube! with qbupdateresouce.

If there is ever the possibility that any entity external to Qube! will consume a license from the pool, then you must use the "license" naming scheme, and implement a mechanism to poll the license server and update Qube! via qbupdateresource whenever the actual license count usage changes.

Per-Host or Not?

The difference between *_host naming scheme and the regular one without the "_host" is to differentiate between the 3rd-party software vendor's licensing scheme, and whether they allow sharing of a license across multiple instances on the same host; Nuke's nukeR and Houdini's hbatch licenses are examples of "shared per-host" licenses. If the license is "shared per-host", then you should use the "_host" naming type, otherwise just use either "global.<name>" or "license.<name>", depending on whether you're polling the license server or not.

Related Pages

- Global Resources
- License Resources
- Per-Host Global Resources
- Per-Host License Resources
- Jobs Reserving Resources
- Externally Updatable Worker Resources and Properties
- Scoped Resources