

# Qube 6.9-1 Release Notes

#####

@RELEASE: 6.9-1

#####

@SUMMARY: This is a maintenance release of 6.9, and includes a number of fixes and improvements to 6.9-0. Recommended upgrade for all 6.9 customers.

#####

==== CL 17696 ====

@UPDATE: add explanation for "deferTableCreation" to the python qb.submit() API routine.

JIRA: QUBE-2400

==== CL 17692 ====

@FIX: another memory leak plugged in the startHost()-related routine, startQualifiedJobsOnHost(). This was causing successful iterations of startHost() (i.e., an instance was dispatched to a worker) to cause memory bloats. Among other places, it was affecting the background helper thread (when it does the "requeuing host" routine).

JIRA: QUBE-2382

==== CL 17649 ====

@FIX: memory leak in preemption code, especially when preemption policy is set to passive or is disabled by the algorithm.

QUBE: JIRA-2382

==== CL 17634 ====

@FIX: memory leak in one of the host-triggered dispatch routines startQualifiedJobsOnHost(), which is called from startHost().

Among other things, this was bloating the memory usage inside the helper routine running in a background thread/process (cleanermain()).

JIRA: QUBE-2382

ZD: 16952

==== CL 17610 ====

@FIX: memory corruption that would cause python or perl to crash when the function was called inside jobs.

JIRA: QUBE-2389

==== CL 17595 ====

@FIX: fixed memory leak in QbPack::store() and storeXML() methods, which were causing, among other things, supervisor threads to bloat when processing large job submissions

JIRA: QUBE-2382

==== CL 17594 ====

@FIX: plugged a potential memory leak in QbDaemon communication code, affecting all server (supervisor, worker) programs

JIRA: QUBE-2382

==== CL 17593 ====

@FIX: plugged memory leak in dispatch code

JIRA: QUBE-2382

==== CL 17592 ====

@FIX: plugged potential memory leak in user permission-check routine, specifically in the group-access check code

JIRA: QUBE-2382

==== CL 17566 ====

@NEW: qbrwk.conf loading optimization (and thus "qbadmin w -reconfig" speed up) by explicitly listing template names and non-existing hostnames in the new [global\_config] section

\* added [global\_config] section to the qbrwk.conf file, and allow new config parameters "templates" to list all qbrwk.conf template section names,

and "non\_existent" to list all non-existent hostnames

\* supe skips ip-address resolution for all section names included in "templates" and "non\_existent", and all reserved names, i.e.: "global\_config", "default", "linux", "osx", and "winnt", thus speeding up the loading of qbwrk.conf file, which in turn speeds up supervisor boot time and "qbadm w -reconfig" operation.

JIRA: QUBE-2346

==== CL 17540 ====

@CHANGE: removed unnecessary submit-time check/rejection of omithosts and omitgroups.

ZD: 16907, 16908

JIRA: QUBE-2366

==== CL 17449 ====

@FIX: directory deletion during log cleanup can fail if the supervisor is updating the job history file at the same time

==== CL 17435 ====

@FIX: supervisor process handling a qbping request should always reread the license file before replying

There was a code path that instructs the supe thread to force-read the license file, but the read was not happening under certain conditions; the code was returning the old cached data if available, or the default count of 2 if the cache isn't available.

\* add a few more informational lines to print to the supelog at license re-reading.

JIRA: QUBE-2317

==== CL 17422 ====

@FIX: make formatting and object instantiation compatible with Python 2.6

==== CL 17416 ====

@FIX: remove unnecessary error message in the schema upgrade routine

JIRA: QUBE-2283

==== CL 17414 ====

@CHANGE: Add more text to describe the subtle yet significant difference between "retry" and "requeue" Python API routines

JIRA: QUBE-2049

==== CL 17403 ====

@FIX: jobs with status "registering" appears when submissions are rejected due to incorrect requirements specifications

ZD: 16408

JIRA: QUBE-2034

==== CL 17402 ====

@FIX: intermittent bug where some supe threads won't properly read the supervisor license key from qb.lic

\* add warning message to print to supelog when the license file reader returns zero-length data

ZD: 16828

JIRA: QUBE-2317

==== CL 17390 ====

@FIX: post-flight should only be run when qbreportwork() is invoked with an agenda-item with terminal-state

JIRA: QUBE-2032

ZD: 16412

==== CL 17376 ====

@FIX: Triggers incorrectly executing multiple times

When a composite (i.e, using && or ||) trigger is specified for a job's callback, such as "done-job-job1 && done-job-job2", the callback would erroneously get run multiple times.

ZD: 16282

JIRA: QUBE-1881

==== CL 17369 ====

@FIX: issue introduced in 6.9 where requestwork() jobtype backend routine will crash when frame padding is 40 or greater.

Python jobtype backend, in particular, was found to crash during a call to the API routine qb.requestwork(), with a "\*\*\*\* stack smashing detected \*\*\*."

error message and a backtrace.

ZD: 16759

JIRA: QUBE-2318

==== CL 17290 ====

@TWEAK: license-reading routine prints the total license count to the supelog

JIRA: QUBE-2003

==== CL 17289 ====

@TWEAK: "ping" handler to print out more info to supelog

Every "qbbing" will print out something like the following supelog now:

```
[Nov 18, 2016 16:25:55] shinyamp[11662]: INFO: responded to ping request from [127.0.0.1]: 6.9-0 bld-custom osx - - host - 0/11 unlimited licenses (metered=0/0) - mode=0 (0)
```

JIRA: QUBE-2002

==== CL 17231 ====

@FIX: disabled verbose option for logging libcurl actions

==== CL 17208 ====

@CHANGE: Populate the subjob (instance) objects with more data (like status), and not just the IDs, when subjob info is requested via "qbhostinfo" (qb.hostinfo(subjobs=True) for python API)

Previously, only jobid, subid, and host info (name, address, macaddress) were filled. Now, things like "status", "timestart", "allocations", etc. are properly filled in.

JIRA: QUBE-2073

ZD: 16541

==== CL 17206 ====

@FIX: When "migrate\_on\_frame\_retry" job flag is set, prevent backend from doing further processing (especially another requestwork()) after a work failed

This was causing race-conditions that will get agenda items to be stuck in "retrying" state, while there are no instances processing them.

Now the reportwork() API routine is modified so that if it's invoked to report that a work "failed", and the "migrate\_on\_frame\_retry" is set on the job, it will stop processing (does a long sleep), and let the worker/proxy do the process clean up.

JIRA: QUBE-2202

ZD: 16553

==== CL 17186 ====

@FIX: "VirtualBox Host-Only Ethernet Adapter" now when daemons (supe, worker) try to pick a primary mac address

JIRA: QUBE-2149

ZD: 16561

==== CL 17182 ====

@CHANGE: all classes that inherit from QbObject print as a regular dictionary, no longer have a \_\_repr\_\_ which prints the job data as a single flat string

@NEW: add qb.validatejob() function to python API, help find malformed jobs that crash the user interfaces

==== CL 17141 ====

@FIX: Any job submitted from within a running job picks up the pgrp of the submitting job

By design, if the submission environment has QBGRPID and QBJOBID set, the API's submission routine will set the job's pgrp and pid, respectively to the values specified in the environment variables.

One couldn't override this "inheritance" behavior even by explicitly specifying "pgrp" or "pid" in the job being submitted, for instance with the "-pgrp" command-line option of qbsub.

Fixed, so that setting "pgrp" to 0 on submission means that the job should generate its own pgrp instead of inheriting it from the environment.

JIRA: QUBE-2141

ZD: 16545

==== CL 17101 ====

@NEW: add "-dying" and "-registering" options to qbjobs.

@CHANGE: also add dying and registering jobs to the "-active" filter.

JIRA: QUBE-2091

ZD: 16469

==== CL 17083 ====

@FIX: Python API: qbping(asDict=True) crashes when used against older (pre-6.9) supe

Among other things, this was causing WV to crash and AV to note an exception (but not crash) when starting up with an older supervisro.

JIRA: QUBE-2084