

# Qube 7.0-0 Release Notes

#####

@RELEASE: 7.0-0

#####

==== CL 19673 ====

@CHANGE: allow Metered Licensing (ML) with just a valid, unexpired supe license, and no worker license

JIRA: QUBE-2823

==== CL 19637 ====

@FIX: Perl API: added proper qb::version() support

==== CL 19636 ====

@NEW: add support for Perl 5.18, 20, 22, 24, and 26 on Windows.

JIRA: QUBE-749

==== CL 19529 ====

@NEW: add paexec.exe and ntrights.exe to aid with proper installation of postgresql server

==== CL 19507 ====

@NEW: add com.pipelinefx.postgresql.plist file to enable launchd support of PostgreSQL DB server on macOS

JIRA: QUBE-3100

==== CL 19504 ====

@NEW: add\_pfx\_account\_osx.sh script that creates the "pfx" account that runs the PostgreSQL DB server.

The script is run from the "postinstall" process of the pkg installer.

JIRA: QUBE-3100

==== CL 19478 ====

@FIX: workers are always "auto-remove"d, even if "auto\_remove" is not set in worker\_flags.

ZD: 18512

JIRA: QUBE-3174

==== CL 19475 ====

@FIX: issue where instances would be stuck in "QB\_PREEMPT\_MODE\_FAIL", causing the supervisor to tell instances to "wait and retry later" in response to retryWork() indefinitely.

Issue was caused when the preemptJobNetwork() routine determines that the instance has started but has NOT yet started working on an agenda item, in which case it would mark the QB\_PREEMPT\_MODE\_FAIL in order to interrupt (i.e. aggressively preempt) the instance; However, the interrupt was not being triggered properly.

Issue was apparently introduced in CL19126.

==== CL 19462 ====

@FIX: issue where some daemon (supe/worker) threads exit early, after processing less client requests than specified via max\_clients (e.g. 65, not 256).

Early exits should now only happen when "max threads" happened earlier.

==== CL 19457 ====

@TWEAK: add a couple of useful supelog lines to pring in assignjob(), regarding result of calling converseWorker() for dispatch

==== CL 19454 ====

@FIX: add call to sendHostReport() so that a statusHost message is sent to the supe when the worker "received kill order for unassigned job". This should eliminate some of the jobs that stay in "dying"(or allow "kill" of jobs that are stuck in "dying")

==== CL 19443 ====

@NEW:Add KeyShot commandline render script for batch rendering

==== CL 19437 ====

@FIX: workid is not duplicated by QbHistory copy constructor

==== CL 19436 ====

@FIX: "down" workers not always detected properly

JIRA: QUBE-3155  
ZD: 18425

==== CL 19425 ====

@FIX: issue when supe thread doesn't hear back from worker during a dispatch. Related to CL19243.

Also fixed an issue (probably harmless) where an extra call to queue.releaseJob() was sometimes made in the findSubjobAndReserveJob() method.

==== CL 19415 ====

@CHANGE: add qbsub support for jobtypes "pyCmdrange" and "pyCmdline".

==== CL 19263 ====

@FIX: log directories for jobs submitted after the utility has been started but before the orphaned log removal is begun are erroneously removed

==== CL 19258 ====

@FIX: not running --use-frm when first-pass repair fails when message has different line-endings than OS X

==== CL 19243 ====

@FIX: add code to avoid mixed-up job instance status when worker-supervisor communications are dropped during job dispatch on an intermittently unreliable network

It was found that network hiccups can cause a worker to not respond to the supervisor during the dispatch of a job instance, but still start running the instance anyway. The worker would send the "running" instance report to the supervisor, which is processed by a separate thread, which updates the DB, causing a status mix-up.

Added code to detect such situations, and allowed the system to let the job run (instead of force-removing it from the duty table) on the worker in question.

Also added error-checking code on the worker side-- if worker detects that it couldn't respond to the supe for a dispatch order, it will give up on that job and release resources that it had just reserved for it.

ZD: 17868

==== CL 19236 ====

@FIX: jobs submitted by non-admin user without a specified priority attempt to submit at priority -1

JIRA: QUBE-3015

==== CL 19209 ====

@FIX: "down" workers would not be detected properly by the supervisor even when the supervisor\_heartbeat\_timeout expired.

ZD: 18057

JIRA: QUBE-3018

==== CL 19178 ====

@FIX: timing issue causing workers to get stuck with job instances.

Issue was seen on a very busy farm with intermittently drops in network communications, when many supe threads would try to dispatch a single instance at the same time.

ZD: 17868

==== CL 19164 ====

@CHANGE: On Unix, by default, supe uses a Unix domain socket to connect to the PostgreSQL server, unless the "database\_host" parameter is set.

The default value of database\_host is "" on Unix (Linux/macOSX), and "localhost" on Windows.

==== CL 19163 ====

@FIX: fix an issue where a worker can sometimes get stuck with a job instance that it's not running any longer

\* Issue was seen when job instances are migrated and there are intermittent networking issues between the supe and worker causing job updates to NO come thru in an expected, orderly fashion.

ZD: 17868

==== CL 19126 ====

@FIX: on a network with intermittent worker-supe communication issues, bad timing can cause job instances to get stuck in "running" state

\* In a bunch of routines that handle job-command executions (i.e., migrate,

kill, etc.) in QbSupervisorCommand, add code to do one last check when a worker is unreachable, to see if the instance still belongs to the worker before updating the instance on DB. It was found that, since a thread dealing with down workers can spend quite a long time, sometimes instances that a worker was processing can be moved off of it and the DB updated by another thread (for example, assigned and running on another worker)-- the check is designed to prevent our thread from overwriting such updates.

ZD: 17868

==== CL 19121 ====

@FIX: job instances can get into an odd state when dispatch routine doesn't hear back from the worker ("found dead").

Networking hiccups can cause this communication drop, which in turn may cause job instances to be "stuck" in the running state on a worker, and be unkillable.

ZD: 17868

==== CL 19118 ====

@FIX: Systemctl unit files for worker and supervisor not installed into correct location

==== CL 19109 ====

@FIX: optimize job cleanup script

@CHANGE: only scan log directories if log removal necessary

@CHANGE: removal of large number of orphaned log directories does not require skipping sanity checks

==== CL 18985 ====

@FIX: 'No database selected' MySQL error when removing ghost jobs

ZD: 17882

==== CL 18911 ====

@TWEAK: add workerlog to show the host's available properties when inspecting a newly dispatched job (when "checking job requirements").

==== CL 18910 ====

@INTERNAL FIX: supervisor patches to help cut down on the number of threads, and reduce chances of repeated worker rejections on some farms due to race-conditions/timing issues.

ZD17713

==== CL 18831 ====

@NEW: add support for retrieval of only a specified range of jobs (IDs, date, N most recent, etc) in the qbjobinfo() API

Changed the "sign" field of QbFilter class to be a QbString rather than a char, to support SQL operators that are longer than a single character, such as ">=", "<>" or "!=".

Added a "limit" and "order\_by" fields to the QbQuery class, so that any query can limit the number of jobs returned, and specify the sort order.

Made change to db-support code (QbDatabase.cpp) and supervisor code (QbSupervisorQuery.cpp and Queue) to take advantage of the above changes and implement the desired range-specific queries.

JIRA: QUBE-2658

==== CL 18822 ====

@FIX: a bug in the startHost() dispatch routine causing the supervisor NOT to always dispatch jobs to workers when they became available.

@INTERNAL: QbServer::printMemUsage() modified to only kick in if QB\_DEBUG\_SERVER\_MEM\_USAGE is defined

ZD: 17713

==== CL 18802 ====

@FIX: Correct 'restrictions' variable name and 'Restrictions' label

==== CL 18717 ====

@FIX: Job instances can become unkill-able with QB\_PREEMPT\_MODE\_FAIL internal status

JIRA: QUBE-2819

==== CL 18680 ====

@FIX: supervisor rpm uninstall leaves the mysql/mariadb service in a stopped state instead of restarting it