Qube 6.10-0 Complete Release Notes

###

@RELEASE: 6.10-1

###
==== CL 20025 ====

@FIX: extremely inaccurate cumulative cpu time for agenda items

JIRA: QUBE-3375
ZD: 18841

==== CL 20014 ====
@FIX: worker is added to the worker_dim dimension table as many times as there are expired entries for that same worker

==== CL 19931 ====
@FIX:Fix relative movie paths in images_to_move.py

==== CL 19897 ====
@FIX: auto_remove worker flag missing from worker config dialogs

==== CL 19834 ====
@FIX: supe/worker RPMs should be installable onto a system with core of the same major.minor mode installed

JIRA: QUBE-3332

==== 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 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 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 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 thur in an expected, orderly fashion.

ZD: 17868

==== CL 19126 ====
@FIX: on a network with intermittent worker-supe commnuication 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 cane 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 18351 ====
@CHANGE: background helper thread improvements

* limit the number of workers that are potentially recontacted by the background helper routine to 50 per iteration.

* background thread exits and refreshes after running for approximately 1 hour, as opposed to 24 hours

ZD: 17124

##
@RELEASE: 6.10-0a
##

@SUMMARY: This is a supervisor-only patch release of 6.10-0 that includes the following key fixes.

NOTE regarding dependencies on Linux:

Installation of this updated supervisor package on a linux system requires the use of rpm with the

--nodeps argument; the yum utility does not support disabling the dependency checks during

installation, only removal.

==== 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 18822 ====
@FIX: a bug in the startHost() dispatch routine causing the supervisor NOT to always dispatch jobs to workers when they became available.

ZD: 17713

==== CL 18717 ====
@FIX: Job instances can become unkill-able with QB_PREEMPT_MODE_FAIL internal status
JIRA: QUBE-2819

##

@RELEASE: 6.10-0

##

==== CL 18422 ====
@UPDATE: Shotgun API from v3.0.1 to v3.0.32
@CHANGE: images_to_movie.py - simplified options and syntax
@CHANGE: qube_imagesToMovie.py - simplified options and syntax
@CHANGE: simplecmd.py - Add "Upload Movie" option to Shotgun parameters
@CHANGE: shotgun_submitVersion.py - fixed movie upload functionality, general code cleanup

==== CL 18356 ====
@FIX: QBDIR set to null-string in job runtime environment

JIRA: QUBE-2611

==== CL 18351 ====
@CHANGE: background helper thread improvements

* limit the number of workers that are potentially recontacted by the background helper routine to 50 per iteration.

* background thread exits and refreshes after running for approximately 1 hour, as opposed to 24 hours

ZD: 17124

==== CL 18340 ====
@FIX: allow special characters in job name field at submissions

JIRA: QUBE-2748

==== CL 18324 ====
@CHANGE: output of "qbadmin s -config" and "qbadmin w -config hostname" now sorted alphabetically.

JIRA: QUBE-2654

==== CL 18285 ====
@FIX: add better error-checks in cmdrange jobtype's log-parsing code, in case the log file is not readable.

In some situations, fseek() was causing crashes in the parseFileStream() routine.

ZD: 17442

==== CL 18221 ====
@FIX: prevent "host.processors" to be unset when jobs are modified.

JIRA: QUBE-2649

==== CL 18185 ====
@CHANGE: make deferred table creation ON by default for all submissions via the APIs (C++: qbsubmit() , Python: qb.submit())

JIRA: QUBE-2603

==== CL 18157 ====
@FIX: shortened the timeout for "qbreportwork" when it reports a "failed" work that has migrate_on_frame_retry from 600 seconds to 20.

This was causing long 10-minute pauses on the job instance when a frame
fails after exhausting all of its retry counts.

Original change was made in CL17206, for QUBE-2202/ZD16553.

ZD: 17447

==== CL 18147 ====
@FIX: Windows worker wouldn't properly release automounted drives at the end of running a job instance

ZD: 17400

==== CL 18107 ====
@FIX: memory leak in a DB-querying supervisor routine.

==== CL 18001 ====
@FIX: Pytnon API's qb.ping(asDict=True) was broken when metered licensing was unauthorized, because of the minus sign

==== CL 17984 ====
@CHANGE: add description of "disable_submit_check" flag to qb.conf.template comment

JIRA: QUBE-2560

==== CL 17982 ====
@CHANGE: Python API: license_provider_name and license_provider_key added to data returned by qb.hostinfo()

JIRA: QUBE-2549

==== CL 17944 ====
@CHANGE: Disable the two free worker licenses for any Qube installation.

JIRA: QUBE-2554

==== CL 17942 ====
@FIX: Some agenda item's "timestart" field doesn't reset when they are killed and then later retried.

JIRA: QUBE-2555

==== CL 17938 ====
@CHANGE: added verbosity in log entries about jobs that are "modified"

JIRA: QUBE-1473

==== CL 17898 ====
@NEW: add "no_defaults" job flag support to Python API files

JIRA: QUBE-2365

==== CL 17897 ====
@NEW: add no_defaults job flag, which tells the system to bypass the supervisor_job_flags.

If a job is submitted with no_defaults set in the job flag, the supervisor will NOT apply supervisor_job_flags.

JIRA: QUBE-2365

==== CL 17889 ====
@CHANGE: job queries requesting for subjob and/or work details now must explicitly provide job IDs.

Both qbjobinfo() C++ and qb.jobinfo() Python APIs now reject such submissions and return an error.

For example, the Python call "qb.jobinfo(subjobs=True)" will raise a runtime exception. It must be now called like "qb.jobinfo(subjobs=True, id=12345)" or "qb.jobinfo(subjobs=True, id=[1234,5678])"

JIRA: QUBE-244

==== CL 17863 ====
@FIX: Qube language callback command "mail-status" wasn't working properly, setting the smtp "TO" field to an incorrect string.

==== CL 17858 ====
@FIX: qb.deleteworkerproperties() and qb.deleteworkerresources() fn should return an error when used with the wrong 2nd arg (must be a list)

ZD: 16932
JIRA: QUBE-2381

==== CL 17856 ====
@FIX: misleading "invalid key" error message in supelog when supervisor_max_metered_licenses set to 0

JIRA: QUBE-2397

==== CL 17821 ====
@FIX: data warehouse worker table updates throttled to a single record at a time when multiple workers simultaneously change their defined slot counts

==== CL 17797 ====
@FIX: ignore any ethernet interface with "virutal" in its description when detecting the primary MAC address on Windows.

ZD 17072

==== CL 17790 ====
@FIX: issue where the background helper thread frequently sends 2 or more update requests (QB_MESSAGE_REQUEST_UPDATE) to a single "questionable" worker (i.e., one that has missed enough heartbeats, and potentially down) at once.

ZD: 17124

==== CL 16491 ====
@NOTES:Add support for AfterEffects point release scheme (2015.3)

==== CL 17763 ====
Supervisor and worker now use correct startup scripts for CentOS 7+, untested yet on CentOS 6.

==== CL 17744 ====
@CHANGE: Add a third paramter, "user", to Custom Policy's qb_approve_modify() routine, so the policy script can allow/disallow modification to a job based on the user name of the requestor.

For example, the routine can now allow certain users to only change priority between 7000 and 8000.

Note that ordinary users are still only allowed to modify his/her own jobs, while admins are allowed to modify anybody's jobs in any way, and are NOT subject to the "approve modify" custom policy routine.

With user groups defined (via "qbusers"), group admins are allowed to modify any job within its group. In that case, the "approve modify" routine does come into play.

JIRA: QUBE-2277

==== CL 17737 ====
@NEW: add 'pgrp' to job data stored in the data warehouse job_fact table.

==== CL 17735 ====
@FIX: badlogin jobs can't be retried or killed (previously fixed in CL15011, but regressed)

JIRA: QUBE-642
ZD: 12699, 17010

==== 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 itereations 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: qbwrk.conf loading optimization (and thus "qbadmin w -reconfig" speed up) by explictly listing template names and non-existing hostnames in the new [global_config] section

* added [global_config] section to the qbwrk.conf file, and allow new config parameters "templates" to list all qbwrk.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 "qbadmin 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 17450 ====
@INTEG: rel-6.9 -> main

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

==== 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 17399 ====
@CHANGE: MSI no longer starting the worker service, qubeInstaller will start if required

==== 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 17375 ====
LEGACY>>>>
@RELNOTES : NO
@INTERNAL: remove even more left-over files from initial metered license tracking

==== CL 17374 ====
LEGACY>>>>
@RELNOTES : NO
@INTERNAL: remove even more left-over files from initial metered license tracking

==== CL 17373 ====
LEGACY>>>>
@RELNOTES : NO
@INTERNAL: remove more left-over files from initial metered license tracking, where db was local to each machine

==== 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 "qbping" will print out something like the following supelog now:

```
[Nov 18, 2016 16:25:55] shinyambp[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 17286 ====
@NEW: exposed Python's qb.admincommand() API routine, and add support for "reverify"

---- Sample Usage ----

```
cmd = {}
cmd['action'] = qb.CONST("QB_ADMIN_ORDER_ACTION_REVERIFY_WORKERS")
cmd['workers'] = ["shinyambp"] # optional

ret = qb.admincommand(cmd);
if(ret == None) :
print "ERROR: qb.admincommand() returned None";
else:
print "INFO: successfully sent admin order";
```

JIRA: QUBE-2159

==== CL 17285 ====
@NEW: add support for "reverify" in Perl's qb::admincommand() API routine

---- Sample Usage ----

```
my $command =
{
"action" => qb::CONST("QB_ADMIN_ORDER_ACTION_REVERIFY_WORKERS"),
"workers" => ["shinyambp"] # optional;
};
my $result = qb::admincommand($command);
if(not defined($result)) {
print STDERR "ERROR: qb::admincommand() returned undef\n";
} else {
print "INFO: successfully sent admin order\n";
}
```

JIRA: QUBE-2159

==== CL 17281 ====
@NEW: add 'qbadmin w -reverify [worker,...]' option to force the supervisor to reverify workers' license provider info.

JIRA: QUBE-2159

==== CL 17231 ====
@FIX: disabled verbose option for logging libcurl actions

==== CL 17208 ====
@CHANGE: Popluate 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 17199 ====
@NEW: add "auto_remove" worker_flag, which indicates to the supervisor that this worker should be automatically removed when it goes "down"

JIRA: QUBE-1058

==== CL 17198 ====
@NEW: add Partner Licensing support to supervisor

JIRA: QUBE-1911, QUBE-1912, QUBE-1913, QUBE-1914, QUBE-1915

==== 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