Flight-Checks (Pre and Post)

New in Qube 6.6

Overview

Flight checks, AKA preflights and postflights, new in Qube 6.6, are programs and/or scripts that are run on the worker just before and after the actual job or agenda starts.

The exit codes of flight checks determine the processing and exit status of their respective job instances or agendas.

System-wide vs Job-specific

System-wide flight checks are installed by the site administrator into predefined locations on the worker systems, and are run for every job. By default, system-wide flight checks are installed in one of four subdirectories under **$QBDIR/flightCheck** on the worker. More precisely:

- $QBDIR/flightCheck/instance/pre (instance-level preflights)
- $QBDIR/flightCheck/instance/post (instance-level postflights)
- $QBDIR/flightCheck/agenda/pre (agenda-level preflights)
- $QBDIR/flightCheck/agenda/post (agenda-level postflights)

where **$QBDIR** is the system's Qube install location, which is, by default, "/usr/local/pfx/qube" on Linux, "/Applications/pfx/qube" on Mac OS X, and "C:\Program Files\Pfx\Qube" on Windows.

The worker parameter **worker_flight_check_path** may be set in qb.conf (or qbwrk.conf), to override the default "$QBDIR/flightCheck"

Any file with a ".txt" extension in those subdirectories are ignored, but every other file will be attempted to run at their respective timings. **It is expected that those files are all proper executables** for the platforms on which they are intended to run-- **otherwise, they will fail and thus jobs will fail**.

Individual Jobs may also specify their own flight checks at submission. For example, the "qbsub" command now supports options such as "-preflights", "-postflights", "-agenda_postflights" and "-agenda_postflights" that may be used to specify a **comma-separated list of full-paths to f light check executables** on a per-submission basis. See the "qbsub" documentation for details. The Qube API's job submit routine also may be used to specify flight checks when submitting jobs. These API job attribute names are "preflights" and "postflights" for instance-level, and "agenda_preflights" and "agenda_postflights" for agenda-level flight checks.

As with the system-wide flight checks (see: Universal Callbacks), these job-specific ones also need to be proper executables for the target platform.

Multiple flight checks may be installed or specified for a single job. Job-specific flight checks are run before the system-wide ones. Note that if any flight check program fails (i.e., returns non-zero), all subsequent flight checks for that job instance or agenda are not run.

The postflight checks are still run if an instance or agenda-item fails.

Instance-level vs Agenda-level flight checks and the effect of their exit codes

Instance-level preflights and postflights are run on the worker just before and after job instances are processed, respectively. Similarly, agenda-level preflights and postflights are run just before and after agendas are processed, respectively.

A non-zero exit code returned from a flight check will abort the processing of any additional flight checks for the respective job instance or agenda, and also affect their exit status. More specifically:

- If an instance-level preflight exits non-zero, any additional preflights for the job instance are skipped, the execution of the job instance is canceled, and the job instance is reported to the supervisor as "failed".
- If an agenda-level preflight program exits non-zero, any addtional preflights for the agenda are skipped, the agenda will be unprocessed, and reported as "failed". The job instance will move onto processing the next agenda.
- If an instance-level postflight exits non-zero, any additional postflights for the job instance are skipped, and the job instance is reported to be "failed".
- If an agenda-level postflight exits non-zero, any additional postflights for the agenda are skipped, and the agenda is reported as "failed". The job instance will move onto processing the next agenda.

Qube-specific environment variables set in the flightCheck environments

Variables in both instance-level and agenda-level flightChecks:

- **QBDIR**
- **QBJOBID**
- **QBPGRPID**
- **QBSUBID**
- **QB_ALLOCATIONS** (a reservation-style string, e.g, '`host.processors=1`')
- **QB_DIR**
- **QB_DOMAIN**
- **QB_FRAME_PADDING**
- **QB_JOBSLOTS** (how many job slots got allocated, can vary when using the "host.processors=N+" reservation)
- **QB_SUPERVISOR**

Additional variables in agenda-level flightChecks:

- **QB_FRAME_END**
- **QB_FRAME_NUMBER**
- **QB_FRAME_PADDING**
- **QB_FRAME_RANGE**
- **QB_FRAME_START**
- **QB_FRAME_STEP**
- **QB_PADDED_FRAME_END**
- **QB_PADDED_FRAME_NUMBER**
- **QB_PADDED_FRAME_START**
- **QB_PADDED_FRAME_STEP**

Addtional variables in postFlight checks

- **QB_INSTANCE_STATUS**
- **QB_FRAME_STATUS**
- **QB_SYSTEM_EXIT_CODE (**only set by cmdline and cmdrange jobs**)**

In postflight scripts, access the status of the last-processed frame or instance via the **QB_FRAME_STATUS** and **QB_INSTANCE_STATUS** envir onment variables, respectively. These will be set to strings such as "complete" and "failed".

Universal Callbacks vs. FlightChecks

At first glance, Universal Callbacks and the Job Pre- and Post-FlightChecks appear similar, but they have an important difference:

- **Universal callbacks** are run by and on the **supervisor** host.
- **Flight checks** are only run on the **worker** hosts.

Tips

When writing postflight scripts for **cmdline** and **cmdrange** jobs, you can query the environment variable **QB_SYSTEM_EXIT_CODE** to find out the numerical exit code of the last command that was run.