Integration and Integration Blockers

Most HotSpot development work is done in `jdk/hs`. Larger projects are recommended to use a project repository which is synced with `jdk/hs` on a regular basis.

ⓘ **When is a change integrated?**

Changes in `jdk/hs` are bulk integrated to `jdk/jdk` once a week. A change in HotSpot will go through these steps on the way to `jdk/jdk`:

1. After proper testing HotSpot developers push changes to the HotSpot repo.
   - Pushes are direct and done using `hg push`
2. When a push is done **Continuous Integration (CI)** will automatically start a set of test jobs on the HotSpot repo.
   - These tests include hs-tier1, hs-tier2, jdk-tier1, jdk-tier2, jdk-tier3, and builds-tier1.
   - It is obviously a good idea to have run the relevant tests before pushing to ensure that no failures happen in CI.
   - If a CI job is already running, the next job will start once the current is done. If several pushes have been made in this time, all will be tested in the same CI job.
   - CI failures are considered **critical failures** and must be dealt with immediately.
3. Every weeknight (Mon - Fri) a **nightly test** job is started at 8 pm PT.
   - The nightly testing runs hs-tier3, hs-tier4, and hs-tier5.
   - New failures that are encountered in the nightly testing are always labeled as **integration_blockers**.
4. Each Friday the changes included in the most recent clean nightly are merged with the most recent clean CI build of master. This merge is then sent to **PIT** (Product Integration Testing).
   - A clean nightly is one where no integration blockers were open when the nightly started, and no new ones were found in the same.
   - If all integration blockers in a nightly snapshot have been fixed before the PIT is integrated, that snapshot can be chosen provided the extra fixes are included in the integration.
5. Once the PIT result has been analysed (usually early the next week), and no integration blockers were found, the merge is **pushed to master** (using `hg push`)

All in all this means that if you need a change to be in master at a specific date, you need to make sure your change is pushed to `jdk/hs` in time to go through the PIT and be pushed to master before the actual date.

ⓘ **What is an integration blocker?**

An integration blocker is **a bug that is present in a repository, but has not yet spread to the upstream repository**. We want to make sure not to spread problems, so if there are new failures in a repository, hold upstream integration.

There is a JBS label that is used to help keep track of bugs that are blocking integration, integration_blocker.

⊘ **Never integrate if there are open integration blockers**

Use the JBS filter Hotspot Integration blockers to find out if there are any blockers.