JBS Label Dictionary

This table contains some frequently used JBS labels and their meaning. Please help keeping this dictionary up to date by adding your favorite labels. This table does not dictate how to use labels, but rather document how they are used. That said, obviously it will help everyone if we try to follow a common standard and use similar labels in the same way across all entities that use JBS.

Labels are case sensitive

When using labels in Jira gadgets (like pie charts, heat maps, and statistics tables) Jira will be case sensitive and treat **`OpenJDK`** and **`openjdk`** as two different labels. Searching however is case insensitive. This means that if you filter out a set of issues based on a label, and then click that label to see a list of the issues, that list will contain more results than the filtered result shows if there are usages of the label with different casing. This can be very confusing and for this reason the recommendation is to stick with the commonly used case for all labels, regardless of your personal taste for upper or lower case letters.

Label	Description
(Area)-interest	Used to indicate that an "area" (usually a team or project) is interested in the issue. This label doesn't indicate ownership of the issue. E.g. **`redhat-interest`**, **`azul-interest`**, **`coin-interest`**
(Area)-related	Used to indicate that an issue is related to a specific area (usually a feature or project). This label doesn't indicate ownership of the issue. E.g. **`graal-related`**, **`testcolo-related`**, **`doc-related`**
(Rel)-bp	Used to indicate that a bug would be suitable for backport to *(Rel)*. This is not a decision to backport, just a suggestion / recommendation. E.g. **`11-bp`**
(Rel)-critical-request • *(Rel)*-critical-approved	Used in the rampdown phases to request approval of changes that requires project lead approval (or similar) to include. *(Rel)* is the release in question, e.g. **`jdk11-critical-request`** *(Rel)*-critical-approved is used to signal that the change has been approved for inclusion. E.g. **`jdk11-critical-approved`**
(Rel)-defer-request • *(Rel)*-defer-approved	Used to request deferral of changes that requires project lead approval (or similar) to defer. *(Rel)* is the release in question, e.g. **`jdk12-defer-request`** *(Rel)*-defer-approved is used to signal that the deferral has been approved. E.g. **`jdk12-defer-approved`** Further details are found in JDK Release Process.

(Rel)-enhancement-request • *(Rel)*-enhancement-yes • *(Rel)*-enhancement-no	Used in the rampdown phases to request the late inclusion of an enhancement. *(Rel)* is the release in question, e.g. **jdk10-enhancement-request** *(Rel)*-enhancement-yes and *(Rel)*-enhancement-no are used to indicate the response on the request. E.g. **jdk10-enhancement-yes**, **jdk10-enhancement-no** Further details are found in JDK Release Process.
(Rel)-fix-request • *(Rel)*-fix-SQE-ok • *(Rel)*-fix-yes	Used to indicate that an issue would be of interest to get integrated in *(Rel)*. E.g. **jdk12u-fix-request** • *(Rel)*-fix-SQE-ok is used to indicate that the issue will be covered by the test plan for *(Rel)*. E.g. **jdk12u-fix-SQE-ok** • *(Rel)*-fix-yes is used to indicate that an issue has been approved for backport to *(Rel)*. E.g. **jdk12u-fix-yes** The labels are placed (only) on the main JBS issue. Further details are found in JDK Release Process and JDK Update Releases.
(Rel)-na	Used to indicate that the issue does not affect *(Rel)* or later. Could for instance be a bug in code that was removed in *(Rel)*.
(Team)-triage[-*(Rel)*|d]	Used to indicate that *(Team)* has triaged this issue for *(Rel)*. The release is optional, but it is encouraged that all bugs are triaged regularly. Using the release will make it easier to keep track of which bugs has been triaged for each release. E.g. **rt-triage-13**, **jmx-triaged** There are many label variants that include the word triage in some form. The form described above is the only one recommended. Please refrain from using other forms.
aot	Used to indicate that an issue is related to ahead of time compilation (AoT).
~~**appcds**~~	**Deprecated.** Was used to indicate that an issue was related to application class-data sharing (AppCDS). The **cds** label is now used instead.
c1	Used to indicate that an issue is related to the C1 JIT compiler.
c2 • c2-.*	Used to indicate that an issue is related to the C2 JIT compiler. c2-.* labels are used to identify different c2 features. E.g. **c2-intrinsic**, **c2-loopopts**
cds	Used to indicate that an issue is related to class data sharing (CDS).
cleanup	A **cleanup** is an enhancement that has no semantic changes, who's only purpose is to make the code more maintainable or better looking.
gc-.*	Used to indicate that an issue is related to a specific garbage collector in the JVM. E.g. **gc-g1**, **gc-shenandoah**, **gc-serial**, **gc-epsilon** There are also labels in use to identify different GC features or areas rather than GC algorithms. E.g. **gc-g1-fullgc**, **gc-largeheap**, **gc-performance**

graal	Used to indicate that this is a Graal issue. (Something that needs to be fixed in Graal rather than OpenJDK.)
graal-integration	Reserved for Graal integration umbrella bugs. The automated integration script will break if this label is used for other bugs.
hgupdate-sync	Used to identify backport issues automatically created by HG Updater (a script that monitors the hg repositories for changes).
~~**hs-nightly**~~	**Deprecated.** Was used to tag bugs found in the HotSpot nightly testing. Since we are now running tiered testing there is no more nightly HotSpot testing. See `tier[1-8]`.
hs-sbr	Used to tag bugs that are found in the "same binary runs", a stress testing method used to find intermittent failures.
~~**hs-tier[1-8]**~~	**Deprecated.** Was used to identify which HotSpot tier a test failure was seen in. We don't separate HotSpot tiers from other JDK tiers anymore. See `tier[1-8]`.
i18n	Used to indicate that an issue is related to internationalization (i18n).
integration_blocker	Used to indicate that a bug is present in a downstream repository but not present in the upstream repository and is therefore blocking integration of downstream changes into upstream.
intermittent • **intermittent-environment** • **intermittent-hardware**	An **intermittent** issue is one that fails sometimes but not always. The exact reason for the intermittent failure is per definition unknown. Once the reason has been identified the issue is no more considered intermittent. An issue isn't intermittent if some characteristics has been found that triggers the failure consistently, even if the actual cause for the failure has not been found. For instance if a test fails every time it is executed on a specific host but not on other hosts it wouldn't be considered intermittent as it fails consistently on that specific host. In other cases it may be that we know that a test sometimes is unlucky in some respect and fails due to this. This test could still be considered intermittent even though we know what the reason is if the reason itself appears intermittently. Some issues may seem intermittent when looking at test results, even though the reason for failing is actually known. One example is where a test fails consistently on a specific host, or due to specific conditions in the environment. These failures should not be considered intermittent but it may still be valuable to tag these in JBS with one of the labels **intermittent-hardware** or **intermittent-environment**. This will help to faster identify that the cause of the failure is known without having to read through the entire bug. A test that should be platform agnostic but is consistently failing on a specific OS would for instance be labeled with **intermittent-environment**, while a test that fails every time it is run on some specific hardware would be labeled with **intermittent-hardware**.
maintainer-pain	Bugs that for some reason is wasting engineering time just by existing, or in other ways are causing pain for the maintainers of the JDK. Examples are bugs that occur frequently in testing or test failures that are time consuming to investigate before determining that it is a pre-existing bug.

`noreg-.*`
 	
`nounit-.*`
	The **noreg-.*** and **nounit-.*** labels are used to explain why a bugfix doesn't need/have a regression test or a unit test. A detailed explanation of all the different cases are found in The OpenJDK Developers' Guide.

performance	Used to identify a bug with noticeable performance impact. Either positive or negative.
~~**pit**~~	**Deprecated.** Was used to indicate that a failure happened in product integration testing (PIT). Since we are now running tiered testing there is no more PIT. See **tier[1-8]**.
problemlist	One or more tests has been problemlisted due to this bug.
regression	A **regression** is a bug that did not exist in the previous release. Ideally all regressions must be fixed in order to release the next major version.

<table>
<tr><td>release-note</td><td>Used to indicate that the issue is a release note. The release-note issue is a sub-task to the main JBS issue containing the text to be used in the release note. The release note must also have one of the following labels:<table>
<tr><td>RN-NewFeature</td><td>New Feature or enhancement.</td></tr>
<tr><td>RN-IssueFixed</td><td>A significant issue which has been fixed, would normally be a regression or an issue which unknowingly released in a new feature.</td></tr>
<tr><td>RN-KnownIssue</td><td>An issue that was not possible to fix by the time the release was GA'd.</td></tr>
<tr><td>RN-Removed</td><td>Covers an API, feature, tool etc. which has been removed from the JDK.</td></tr>
<tr><td>RN-Deprecated</td><td>Covers an API that has been marked as deprecated in the release.</td></tr>
<tr><td>RN-Important</td><td>Used to indicate that the release note should be highlighted in some fashion, such as listing it at the beginning of the release notes.</td></tr>
<tr><td>RN-(distro)</td><td>Used to indicate that the release note is only relevant for a specific JDK distribution. E.g. RN-Oracle</td></tr>
<tr><td><s>RN-Change</s></td><td>Deprecated.</td></tr>
</table></td></tr>
<tr><td>release-note=yes

release-note=no

<s>release-note=done</s></td><td>Used to indicate wether a change requires a release note or not. The labels are (only) placed on the main JBS issue.

release-note=done is deprecated and should no longer be used.</td></tr>
<tr><td>starter</td><td>A starter bug is a well contained, small issue that is suitable for someone new to the codebase.</td></tr>
<tr><td>startup</td><td>Used to identify a bug as affecting Java SE startup.</td></tr>
<tr><td>tck-red-(Rel)

<s>tck-red</s></td><td>Used to identify TCK conformance stoppers (e.g. failure of a valid TCK test that exists in a shipped TCK). The release number indicates which release of the TCK that failed. E.g. tck-red-11

There are tck-red labels without the release number out there as well. This usage is deprecated.</td></tr>
<tr><td><s>test</s>

<s>test-only</s>

<s>testbug</s></td><td>The labels test, test-only, and testbug label are deprecated and should no longer be used. Use noreg-self to indicate that an issue is a bug in test code.</td></tr>
<tr><td>tier[1-8]</td><td>Used to indicate which tier a test failure has been seen in. Lower tiers would in general mean higher urgency to fix the issue. E.g. tier1, tier2</td></tr>
<tr><td>webbug</td><td>Used to identify a bug as submitted on bugs.java.com.</td></tr>
<tr><td>zgc</td><td>Used to indicate that an issue is related to ZGC.</td></tr>
</table>