

Jira

Hadoop Jira

Hadoop uses [Jira](#) as its issue tracker. All changes are discussed and reviewed in Jira before committing to subversion.

Hadoop Special Fields

"Release Note" Text

Add a brief comment (1-3 sentences) if this item is likely to be of interest to application programmers. New features and changes to APIs or operating procedures require a release note. Don't bother with a simple "fixed bug" message. Do reference other documentation, if applicable, but don't bother with any reference to the Jira item itself. These comments are collected semi-automatically into the release notes document as the release nears completion. They may be edited for content and style by the folks helping to prepare the release. Do be careful with grammar and spelling as release notes are seen by people who do not browse Jira. Hadoop style is to begin the first sentence with a verb. For example:

Changed the output of the "fs -ls" command to more closely match familiar Linux format. Additional changes were made by HADOOP-3459. Applications that parse the command output should be reviewed.

"Incompatible Change" Flag

Set this flag if the change will result in an incompatibility with existing code. An incompatible change should also have a release note!

- Changing an API or the semantics of a system call
- Moving or renaming a file
- Changing the format of a log message (logs are processed by automated tools)
- Adding or removing or renaming a configuration parameter
- Changing metrics
- Changing the format of a shell command response (output is processed by automated tools)
- Modifying server web pages

"Reviewed" Flag

A reviewer should set this flag when no further reviews are required. A change that is "patch available," "reviewed," and has passed the automated tests is ready to be committed.

Jira Best Practices

Please observe the following guidelines when using Jira for Hadoop.

- Give issues a meaningful name. "Hadoop Bug I have found" is not appropriate. Better something like "NPE during datanode startup on OS/X"
- Keep issue descriptions short. The issue description field should briefly describe a *problem*, not a solution. Detailed descriptions, stack traces and proposed solutions should be subsequently added in the comments. This is preferred for several reasons. The description is included in every email Jira sends about an issue, so a long description generates big messages. Also, solutions should be arrived at through consensus-building discussion, so it is inappropriate to present a solution at the outset.
- Don't edit comments. If you make a mistake, add another method clarifying things. This makes it easier for folks to follow the discussion.
- Don't delete comments. If you have changed your opinion, add a new comment noting that rather than deleting your original comment.
- Use the issue name as the patch file name. As you update a patch, re-use the same name for the file. Do not remove old versions of the patch, but rather overwrite them with a new patch file of the same name. The standard name for patch files is the name of the issue with ".patch" appended, e.g., HADOOP-1.patch.
- Assign the issue to yourself when you're the primary author of a patch. This lets folks easily list the patches a contributor has written, to decide whether they are qualified to become a committer.
- Limit Jira email volume as follows:
 1. Configure your email client to filter all but new issue creation messages, skipping messages sent to the -dev list from Jira whose subject does not begin with "Created";
 2. For new issues that you wish to follow, use Jira's "watch" feature, so that emails are sent directly to you.
- Configure the JIRA portal to show only the project(s) that you care about; consider adding the Issues in Progress and Assigned To Me portlets.