

# Committer's Guide

## Welcome, New Committers!

To get started:

1. Send the following information to cwilper (at) duraspace.org:
  - Your duraspace.org userid. This is used to grant you committer-level permission on the resources hosted on the website (JIRA, Confluence, Bamboo, etc)
  - Your sourceforge.net userid (If you don't have one, create one [here](#)). We use sourceforge.net to distribute releases, so we need your userid there in order to grant you write access.
  - Your GitHub userid (If you don't already have one, sign up [here](#). GitHub hosts our source code repositories for the project; we'll need your id to grant you write access.
  - If you're a Skype user, your Skype userid. If you don't use Skype, please provide a phone # and/or Instant Messaging info.
2. [Sign up to the Fedora Development list](#).
3. [Sign up to the Fedora Codewatch list using your userid@sourceforge.net address](#). This is a public list that all committers subscribe to. It's used solely for automatic notification of Subversion commits and automated test results. It is necessary to sign up using your sourceforge.net id; commit notification will not work otherwise. Make sure you have set up your Sourceforge account so that emails to userid@sourceforge.net are forwarded to your usual address.

## How We Keep In Touch

The team keeps in touch on a day-to-day basis via the development list and Skype. We generally try to keep significant discussions on the list, but prefer Skype for higher-bandwidth discussions.

We also have weekly, virtual [Fedora Developer Meetings](#) where we discuss our recent work on the FCRepo project. This also gives us a way to share what we're doing with interested members of the community.

## What To Work On

While we collectively prefer the high-priority tracker items (as determined by the user community and committers) to be worked on first, all contributions are appreciated. You may work on any unclaimed items that exist in the [FCREPO tracker in JIRA](#).

If you want to work on a new feature that isn't in the FCREPO tracker yet, please feel free to submit it, and open it if it's an obvious bug or improvement that needs work. If not, leave it in the "Received" state – this will trigger a discussion of the issue in an upcoming [Fedora Developer Meeting](#).

## Claiming an Issue

Once you have an issue to work on, simply assign yourself as the owner in JIRA. This lets everyone know that you plan to begin working on the issue soon. If you find that you cannot complete an issue, you should 1) remove yourself as the owner to give other developers an opportunity to work on it and 2) if you've created a branch for it, remove the branch.

## Coding Guidelines

### • Code Style

The project's coding style is currently best described by the Eclipse settings in src/build. Here are the major rules:

- Four space indents for Java, and 2-space indents for XML. NO TABS.
- K&R style braces:

```
if (code) {
    // code
} else {
    // code
}
```

- Don't use wildcard imports
- Write Javadocs for public methods and classes. Keep it short and to the point.
- Avoid public instance variables; use accessors.
- Use public methods sparingly; implementation details are not public.

### • Logging

We currently use SLF4J for logging. When writing log messages (especially INFO, FATAL, and WARN), consider your audience. Don't over-use INFO; these messages are always logged by default.

### • Testing

Use JUnit for unit, integration, and system tests. When making changes, please take the time to write tests as needed. We are not looking for 100% coverage here, but the developers will have a lot more confidence in your code if you can show that it works. See the readme.txt in the root of the source tree as a jumping-off point for adding your tests to one of the suites. Here's a brief description of the basic types of tests we run:

- **Unit** - Tests a single class. These tests are typically very quick to run, and therefore get the most use by developers. Unit tests don't touch the disk or interface with classes other than mocks. If you can cover a scenario adequately in a unit test, you should prefer implementing it as such (and not as an integration or system test)
- **Integration** - Tests multiple classes working together. May touch disk or interact with a "test" database.
- **System** - Works against a running Fedora server in a given configuration. These are "block box" style tests against the public APIs of Fedora. These take much longer to run than unit or integration tests because they require significant setup.

## Using Git

Please see [Git Guidelines and Best Practices](#)

## Closing an Issue

Once the relevant code has been committed and passes the automated tests, you can resolve the issue in JIRA. Prior to resolving an issue, double check to make sure the metadata is accurate in the tracker.

## How we do Releases

See the [Fedora Release Process](#) document.

## Development Tools

If you find a need for licensed software to do your Fedora development (e.g. an IDE, Java profiler, other tool), contact DuraSpace (email: sysadmin at duraspace.org) and we can try to obtain a license. Many non-free, licensed development tools offer free "open source" licenses. Some tools that DuraSpace has already obtained committer licenses for include: IntelliJ IDEA and YourKit Java profiler.