Tickets

Where is the OpenMRS issue tracker?

The OpenMRS JIRA site contains all of our issue tracking and project management tools. (We're using a free open source JIRA license from Atlassian.)

View the currently unresolved issues across all projects.

View a list of introductory issues that are marked as lower complexity.

What is in an Issue?

Issues are used to track both bugs and suggested new features. An issue can be against the main/core code base (TRUNK) or against a module (like FRM).

Issue-related Conventions

- Development discussions should not occur within JIRA issues; rather, development discussions should happen on OpenMRSTalk.
- If you see involved development (design) discussions happening within an issue, please reply to them and then add a comment to the issue stating that "This is a good design discussion, I've added some thoughts on OpenMRSTalk" along with a link to your post.
- Once design or other development ideas have been discussed and conclusions made, then those can be summarized in a comment on the issue.

How to Create an Issue

1. Search existing issues to make sure that your bug or enhancement has not already been described.
2. If you believe your bug/enhancement is new, create a new issue describing the bug/enhancement under the project you intend to contribute to. (Note: You must have an OpenMRS ID to create or work on issues. You can sign up for one on the OpenMRS ID web site. If your OpenMRS ID does not have access to JIRA, contact helpdesk, explaining why you want access and wait for a response via your email address. You are able to select the project where to create your ticket under when filling in the create issue form on JIRA)

How to work on an OpenMRS Issue, in 5 minutes

1. Make sure you have an OpenMRS ID--this gives you an account to our issue tracking system.
2. Identify the issue you want to work on. (It should have the status "Ready For Work").
3. Click the "Claim Issue" button.
   - This will change the status to "In Progress" and mark you as the Assignee, so nobody duplicates your work.
   - If you don't see the button, comment on the issue saying so. (This probably means the issue was not "Ready For Work").
   - If you have any questions about the issue, or it is not clear from the description and comments exactly what is being asked for, then make a comment on the issue asking for clarification.
   - If you expect to spend a significant amount of coding time (>= 1 day), then make a comment on the issue describing your planned approach. (In case your plan needs modifications, better to know this early.)
   - If the issue is going to introduce non-trivial user interface changes, add a mockup to the issue via More Actions -> Add/Edit UI Mockup. This will typically save you time, by forcing you to design the complete workflow up front. And it allows the issue's reporter and other watchers to make suggestions.
4. Design. Write code. (Follow Java Conventions.) Test.
5. Make sure you write one or more Unit Tests.
   - If you don't know about the @should annotation and then Generate Test Case Plugin, then please read them.
6. Once your code is ready, push it to a branch in your github fork of the repository (see Using Git), open a Pull Request, and comment on the issue with a link to the PR.
   - If there is a significant user interface component to your code, attach one or more screenshots to the ticket also.
7. Click the "Request Code Review" button.
   - This will put the issue in a Code Review (Initial) status.
   - (You may want to read more about Code Review)
8. Comment on the issue saying what you want reviewed.
   - If there is only one attached patch, or it's truly obvious which file should be reviewed, you can skip this.
9. Eventually, your code will be reviewed. Multiple things can happen at this point:
   - The reviewer may move the issue back to "In Progress" and make comments in the GitHub Pull Request on the changes you need to make before re-submitting.
   - The reviewer may merge your changes into the master branch of the source code repository, and request that you make further post-commit changes, also moving the ticket back to "In Progress".
   - The reviewer may merge your changes into the master branch of the source code repository, and mark the issue as Closed. (Typically this happens for trivial fixes.)
10. During a GitHub Pull Request review for your issue, you will likely receive quite a few email notifications. (You don't necessarily need to reply to each one individually.)
    - If the code review didn't raise any issues (or if they were small enough that the reviewer fixed them directly) the issue will be closed.
    - If any changes need to be made, then the issue will be put back "In Progress" and the head reviewer will make a comment on the issue pointing to the GitHub pull request comments, that you need to address before resubmitting your code for review again.
    - If you have any questions about these comments, make a comment yourself asking for clarification.
11. If code changes are necessary, then make those changes, and push them to your github fork. Then re-open your pull request (which will now include your original commit, plus the new one).
   • If your code was merged to the master branch of the source code repository, then you’ll want to merge that in from upstream, and make your revisions relative to your now-merged code.
   • If your code wasn’t applied to the source code repository, then you’ll need to start working from your prior fork and branch.
   • Comment on the issue summarizing the issues you addressed in the new commits.
12. Click "Request Code Review" again.
   • (Now we’re back to step 9.)

**JIRA FAQ**

**What is the dashboard?**

A dashboard in JIRA is a customizable homepage for you. Each user can have a different dashboard. By default, OpenMRS will show the dashboards that is all about the upcoming releases.

Click the **Tools-->Find Dashboards** link in the upper right from the jira homepage to see what other custom dashboards people have shared.

**Can I work on an issue?**

Yes, click the "Claim Issue" button (after signing in) to assign the issue to yourself. If you can no longer work on an issue, click the "Unassign" button to allow someone else to pick it up and continue.

**What is "TRUNK"?**

This is the shortname we gave to the issues that are related to the main code base. Because the main code base was stored in http://svn.openmrs.org/openmrs/trunk, the issues are in a project named TRUNK. All issue numbers in TRUNK start with that string (like TRUNK-343) to differentiate from other projects in JIRA

**When to open an new Issue?**

Always. Even if it requires a minor change, consider opening an issue as it helps keep a history of what happened and facilitates further discussion on a particular issue.