

# java.net Issue Tracker

4/5/2008

History of changes:

- None

[This handout describes the way java.net will be used in the GSD project. It assumes that you are logged into java.net. This tutorial is not exhaustive.]

In java.net the bug tracking is called *Issue Tracker* and is accessible from the left of the screen.

Developers and developers' coaches will use *Issue Tracker* as developers and will submit bugs, manage bugs' lifecycle and generate bug reports.

Clients, client coaches and auditors will use it as observers and will submit bugs. Auditors will also assess the quality of the software product that will be developed based on the generated bug reports.

***Note: The list of features implementation and testing status has to be posted on the wiki in the Coding tab under Implementation/Testing status.***

## 1. Submitting a bug

- To submit a bug, click on *Issue Tracker*.



Figure 1 - Enter an Issue – Main Page

- Choose the issue type to be submitted.

The issue types are:

- **Defect:** a problem with an existing feature that is not developed to spec or does not work as designed. These are often referred to as "bugs."
  - **Enhancement:** an improvement to an existing feature.
  - **Feature:** an addition to the software to add a piece of functionality that does not yet exist.
  - **Task:** an activity to be done on behalf or in support of a feature or enhancement. Tasks do not typically require direct changes to the code base.
  - **Patch:** a special kind of issue: a section of code to be applied or attached to existing software. Often, a patch fixes a defect.
- The issue description must be entered with as much detail as possible such that the developers can reproduce / pinpoint the bug. In

particular and if appropriate, in the description of the bug, enter the login information, browser (IE or Firefox), URL, and path (sequence of clicks) to access the URL that resulted in the bug.

- The bug has to be assigned to the developer in charge of the implementation of the feature. The developer in charge is posted on the wiki in the Coding tab under Implementation/Testing status.
- When a bug is submitted all developers will get an email (including the developer who is required to fix the bug).

The screenshot shows a web browser window displaying the 'Enter Issue' form. The form is titled 'Enter issue' and has a green header bar. Below the header, there are several fields for entering issue details. The 'Reporter' field is filled with 'naryambis'. The 'Component' field is filled with 'luci', and the 'Subcomponent' field is filled with 'jython'. The 'Found in version' field is filled with '2.0.0'. The 'Platform' field is set to 'All', and the 'OS' field is set to 'All'. The 'Priority' field is set to 'P3', and the 'Issue type' field is set to 'DETECT'. The 'Assigned to' field is set to 'Lookup users'. There are also fields for 'CC', 'URL' (filled with 'http://'), and a 'Summary' field. A large text area for 'Description' is located below these fields. At the bottom of the form, there are three buttons: 'Submit issue', 'Reset', and 'Remember values as bookmarkable template'. The browser's address bar shows the URL 'http://lucy.dev.java.net/issue/AssignableUsers'.

Figure 2 - Enter an Issue Feature

[Help on issue descriptions per java.net]

### Version

This field identifies the release where the issue or defect was found.

### Component/Subcomponent

Identify the area within the project that this issue is associated with. Only one selection is permitted.

### *Platform*

Select the hardware platform used when the defect was discovered. For example, these options include the following:

- All (happens on all platform; cross-platform issue)
- Macintosh
- PC
- Sun
- HP

**Note:** Selecting the option "All" does not select issues assigned against all platforms. It selects issues that *occur* on all platforms.

### *Operating System*

Select the operating system used when the issue was discovered. For example, these options include the following:

- All (happens on all operating systems, making this a cross-platform issue)
- Windows 95
- Mac System 8.0
- Linux

Note that the operating system implies the platform, but not always. For example, Linux can run on PC, Macintosh, and others.

### *Priority*

Assign a level of importance in this field to help determine the priority. This field is used by the programmers/engineers to prioritize their work.

- P1 - Most important
- P2
- P3
- P4
- P5 - Least important

## *Issue Type*

**Defect:** a problem with an existing feature that is not developed to spec or does not work as designed. These are often referred to as "bugs."

**Enhancement:** an improvement to an existing feature.

**Feature:** an addition to the software to add a piece of functionality that does not yet exist.

**Task:** an activity to be done on behalf or in support of a feature or enhancement. Tasks do not typically require direct changes to the code base.

**Patch:** a special kind of issue: a section of code to be applied or attached to existing software. Often, a patch fixes a defect.

## *Initial state*

If you do not have 'Project Issue Tracking - Change' permissions, any issues you enter will have the initial state of "new" or "unconfirmed." Marking an issue [unconfirmed](#) means it is not yet determined whether it is true or valid. Read more about "state" in the [lifecycle of an issue](#).

## *Assigned To*

Enter the username of the individual in charge of resolving the issue. Alternatively, in the Lookup Users dialog, click the name of a user to assign the issue. The name is entered automatically in the Assigned To field. If the Assigned To field is left blank, the issue is assigned to the component/sub component owner by default. Every time this field changes, the status changes to **NEW** to make it visible in the assignee's list of issues.

## *CC:*

Add usernames of other individuals who need notification when this issue changes status, or when there is activity on this issue. Delimit multiple usernames by single spaces only -- no commas or semi-colons are necessary.

Note: Assign cc addresses sparingly. Project participants whose interest or involvement in this issue is peripheral should be encouraged to use the Issue Tracker to check and track issues rather than to rely on automatic email notification.

## *URL*

How to use this field depends on the issue type:

- For defects, the URL should lead to a fairly stable system where the the problem is obvious or can be easily reproduced.
- For enhancements, the URL should provide details pertaining to the improvement, such as mockups.
- For features, the URL should link to web-based material explaining the improvement such as mockups or design specs.
- For tasks, the URL is optional and may include linking to the associated feature or enhancement.

## *Summary*

This statement should consist of a few unique, self-explanatory words identifying the issue easily in reports and short lists. A summary is described as a terse, specific statement crafted to describe the issue. Limiting your entry to the field width renders the best columnar display for query results.

## *Description*

Provide a full description of the issue including any pertinent history or activity around this issue. Because this field is additive, it serves as the knowledge base and means of communicating through this issue's life cycle. Other project participants view and add comments or information using this field.

## 2. Managing Bugs

- Developers will get an email for each bug that is submitted.
- The issues assigned to a particular developer are visible in *My Issues*. Developers only see the issues assigned to them. If they want to see all the issues they can see them by querying the database (See below).



Figure 3 – My Issues – Main Page

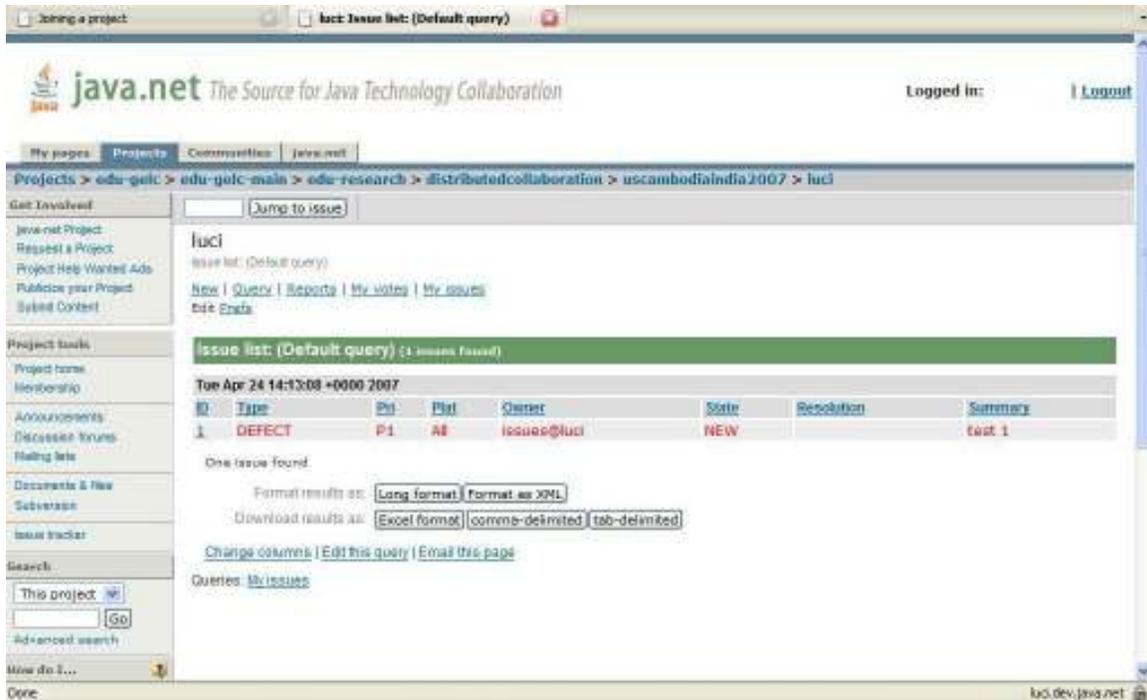


Figure 4 – List of Assigned Issues

- The first thing that a developer needs to do when working on fixing a particular issue is to *ACCEPT* the issue. The resolution statuses of an issue are: fixed, invalid, wontfix, later, remind and worksforme. It is the responsibility of the developer to change the status of the issue. For example, once the issue is fixed then its status will be changed to fixed.

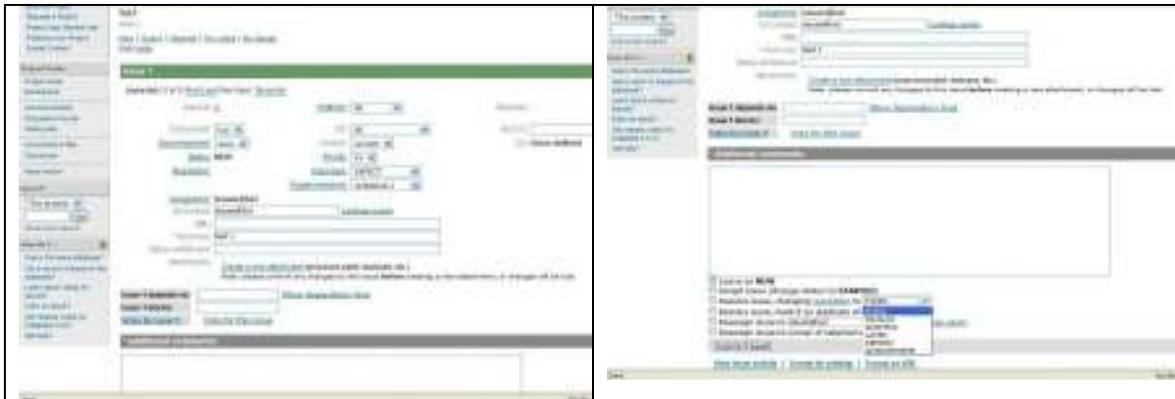


Figure 5 – Details about an Assigned Issues

### 3. Generating Reports

- Reports can be generated by clicking *Reports* or by using *Query Database*.



Figure 6 - Reports – Main Page

- Querying the database is available through the *Issue Tracker* main web page.

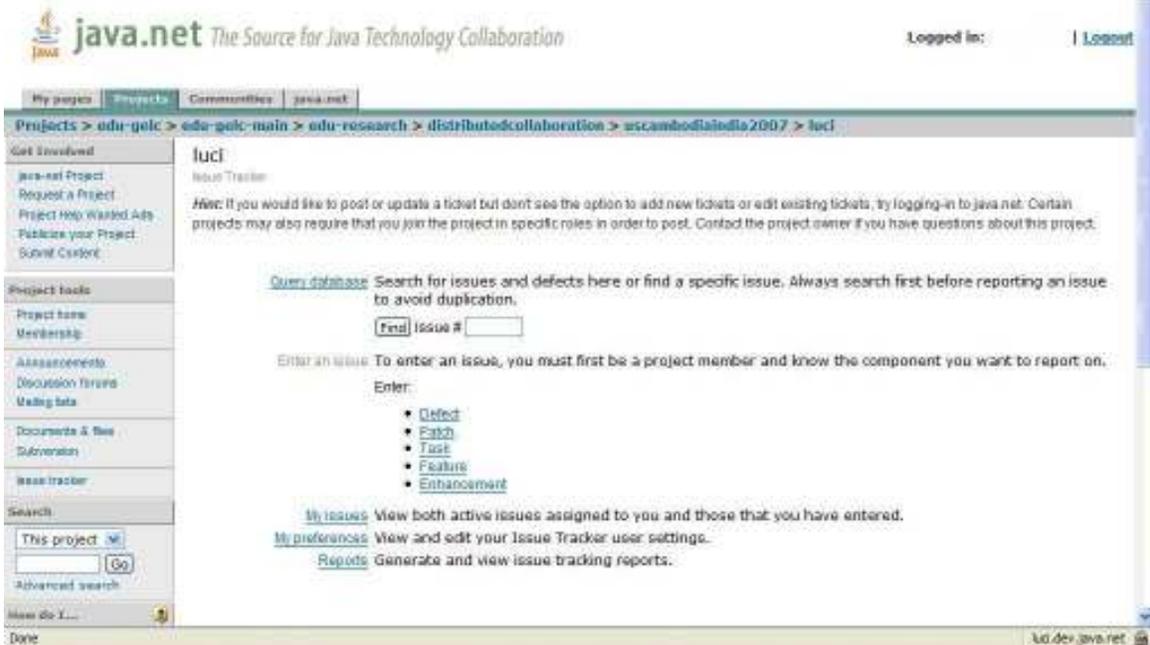


Figure 7 – Query Database Feature

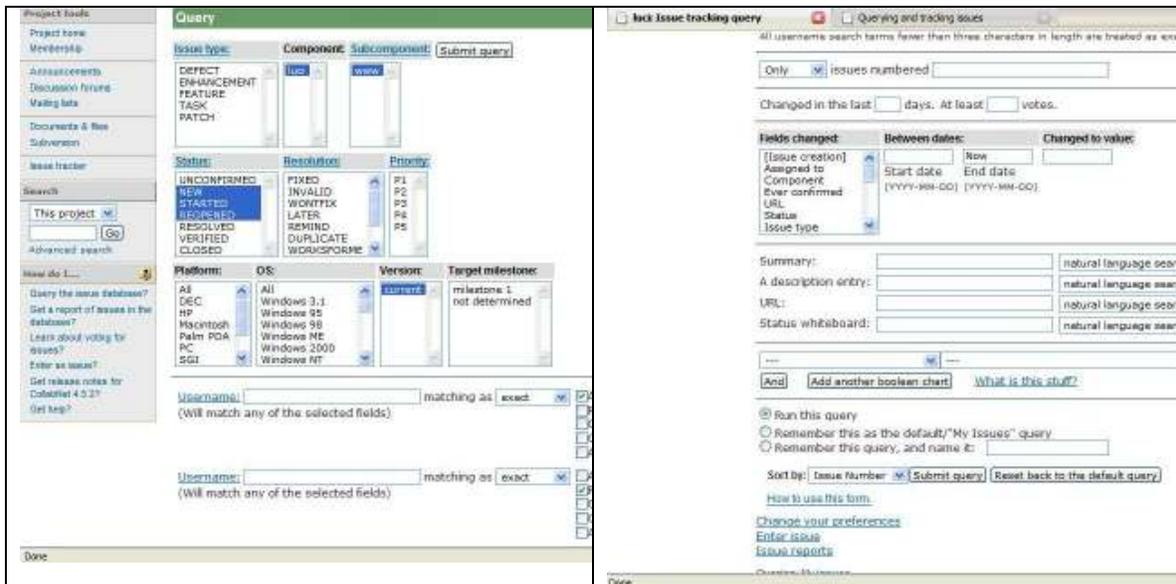


Figure 8 – Elements to Query the Database

## ***4. More help on java.net Issue Tracker***

Accessing Issue Tracker

<https://www.dev.java.net/sdocs/ProjectIssues#accessissues>

Entering and Modifying Issues

[https://www.dev.java.net/nonav/sdocs/ddIssues\\_EnterModify.html](https://www.dev.java.net/nonav/sdocs/ddIssues_EnterModify.html)

Querying and Tracking Issues

[https://www.dev.java.net/nonav/sdocs/ddIssues\\_Query.html](https://www.dev.java.net/nonav/sdocs/ddIssues_Query.html)

Generating status reports about project issues

[https://www.dev.java.net/nonav/sdocs/ddIssues\\_Reports.html](https://www.dev.java.net/nonav/sdocs/ddIssues_Reports.html)