

Bugzilla, ITracker, and Other Bug Trackers

Nicolás Serrano and Ismael Giordia

To err is human, and we learn from our mistakes. For software developers, this means knowing what the error is, resolving it, and learning from it. Bug-tracking systems help us do these tasks. This column looks at two specific open source products and provides useful hints for working with any bug-tracking tool.

I look forward to hearing from you about this Open Source column and the products and tools you want to know more about. If you'd like to write for this column, see the author guidelines at www.computer.org/software/author.htm or contact me.

—Christof Ebert

When you work on a software project, you have to manage the bugs you find. At first, you might list them on a spreadsheet. But when the number of bugs becomes too large and a lot of people must access and input data

tracking system, its developers built their own—Bugzilla, also an open source project. The first public release was written in Perl, and the open source community continues to improve it.

Today, many users—including open source projects (Apache, Linux, Open Office) as well as private and public organizations (NASA, IBM)—are using Bugzilla. So, Bugzilla's focus has changed from being a Mozilla tool to becoming a general bug-tracking system.



Using the tool

Like any information system, Bugzilla's main functions are for inputting and accessing data. When you log into the system, you can input new bugs or search for, track, or edit existing ones. Figure 1 shows part of the advanced-search window, and Figure 2 shows a screen displaying a specific issue (a bug, enhancement, feature, task, or patch).

To track a particular bug, you must be able to locate it. The Product, Component, Version, Status, and Reporter fields are related to tracking, whereas the Summary, Status Whiteboard, Keywords, Severity, Attachments, and Dependencies fields are related to fixing it. These fields contain the data you must input when you're reporting a bug and the data that helps you filter your searches.

Another way to find information is through

on them, you have to give up the spreadsheet and instead use a bug- or issue-tracking system. Many software projects reach this point, especially during testing and deployment when users tend to find an application's bugs.

Nowadays you can choose among dozens of bug-tracking systems. Some are commercial, others are open source projects such as the two we describe here.

Bugzilla

When Mozilla (www.mozilla.org), the large open source Web browser project, needed a bug-

The screenshot shows the Bugzilla search interface with several filter sections:

- Summary:** contains all of the words/strings
- Product:** Bugzilla, Calendar, Camino, CCK, Composer
- Component:** Accessibility, Account Manager, Address Book, Addressbook/LDAP (non-UI), Administration
- Version:** 0.17, 0.3, 0.4, 0.5, 0.6
- Target:** Future, mozilla1.8alpha6, 3.0, Jan
- A Comment:** contains the string
- The URL:** contains all of the words/strings
- Whiteboard:** contains all of the words/strings
- Keywords:** contains all of the keywords
- Status:** UNCONFIRMED, NEW, ASSIGNED, REOPENED, RESOLVED, VERIFIED, CLOSED
- Resolution:** FIXED, INVALID, WON'TFIX, DUPLICATE, WORKFORME, MOVED
- Severity:** blocker, critical, major, normal, minor, trivial, enhancement
- Priority:** -, P1, P2, P3, P4, P5
- Hardware:** All, DEC, HP, Macintosh, PC, SGI, Sun
- OS:** All, Windows 3.1, Windows 95, Windows 98, Windows ME, Windows 2000, Windows NT
- Email and Numbering:** Includes checkboxes for bug owner, reporter, QA contact, CC list member, commenter.
- Bug Changes:** Includes fields for "Only bugs changed between" and "where one or more of the following changed" (Bug creation, Alias, Assignee, URL).
- Sort results by:** Reuse same sort as last time

Figure 1. Part of an advanced bug search in Mozilla using Bugzilla.

The screenshot shows the Bugzilla bug view for Bug 131618:

- Bug#:** 131618 alias:
- Product:** Core
- Component:** MailNews: MIME
- Status:** ASSIGNED
- Resolution:** (empty)
- Assigned To:** Jean-Francois Ducarroz <ducarroz@ducarroz.org>
- QA Contact:** jester@formerly-netscape.com.tld
- URL:** (empty)
- Summary:** Html messages don't show some images
- Whiteboard:** (empty)
- Keywords:** (empty)
- Hardware:** PC
- OS:** Windows XP
- Version:** Trunk
- Priority:** P2
- Severity:** major
- Target Milestone:** mozilla1.1alpha
- Reporter:** Vincent Jaubert <neo-8ba@myamail.com>
- CC:** ddkilzer@kilzer.net, oli@helheim.org.uk, patrick.steiner@gmx.ch, pbarteccki@mozilla.org, shults@cs.wcu.edu
- Flags:** blocking-aviary1.0.1, blocking-aviary1.1, blocking1.7.6, blocking1.8b, blocking1.8b2
- Requestee:** (empty)
- Attachments:**

Attachment	Type	Created	Size	Flags	Actions
The mail that show the bug	message/rfc822	2002-03-17 15:01 PST	42.20 KB	none	Edit
Create a New Attachment (proposed patch, testcase, etc.)					
- Dependencies:** Bug 131618 depends on: (empty) [Show dependency tree](#)
- Blocks:** Bug 131618 blocks: (empty) [Show dependency graph](#)
- Votes:** 3 [Show votes for this bug](#) [Vote for this bug](#)

Figure 2. Bugzilla screen view of a specific issue (a bug, enhancement, feature, task, or patch).

Bugzilla's reports, which describe the current state of the bugs, and charts, which describe an application's state over time.

Architecture

Bugzilla is a Web application, so users interact only with its HTML pages. Although Bugzilla's design principles state that it should support commercial databases, it works only with mySQL, a popular open source database. Detailed information about the installation process is available at www.bugzilla.org/docs.

ITracker

We've also worked with ITracker (<http://sourceforge.net/projects/itracker>), an issue-tracking system designed by Jason Carroll in 2002 to support multiple projects with independent user bases. Its features resemble Bugzilla's. The main difference is that it's platform independent (because it's a J2EE application) and database independent. Table 1 compares the two tools.

Other tools

Applied Testing and Technology's Web site (www.aptest.com/bugtrack.html) offers a list of more than 70 bug-tracking tools—some commercial, some open source.


Software configuration management applications provide several tools and methods to manage the life cycle of software products, so they contain bug-tracking systems as well as process management, configuration control, version control, and release management functions. Their prices and setup times are well above those of a focused bug-tracking tool, so SCM systems are actually applications of a different size. Example SCM systems include Clear DDTs, Rational ClearQuest, CM Synergy, and Razor. CM Synergy provides good support for processes through a task-based process model and automated operations for the change process. ClearQuest and Razor stress workflow management, with the option of customizing standard processes. Commercial issue-tracking systems such as TrackStudio let users define workflows for each type of issue.

URLs for More Information

- **Bugzilla:** www.bugzilla.org
- **ITracker:** <http://sourceforge.net/projects/itracker>
- **ClearDDTS:** www-306.ibm.com/software/awdtools/clearddts/version4-7.html
- **CM Synergy:** www.telelogic.com/products/synergy/cmsynergy/index.cfm
- **Rational ClearQuest:** www-306.ibm.com/software/awdtools/clearquest/index.html
- **Razor:** www.visible.com/Products/Razor/index.htm
- **TrackStudio:** www.trackstudio.com
- **ApTest's list of more than 70 tools:** www.aptest.com/bugtrack.html

Another solution we tried was to develop and apply our own bug-tracking system. This approach isn't as powerful as a standalone application, with all the needed fields built in, but it enables a higher level of integration with the application you're developing.

The "Hints for Bug Tracking" sidebar offers additional advice based on our experience.

To choose an application solution for your particular project, check the features you need and think carefully about using applications that exceed your requirements. ITracker is a good solution for a fast setup, whereas Bugzilla has a long tradition in bug tracking, and some of your programmers might already be familiar with it. 

Nicolás Serrano is a computer science and corporate finance professor at the University of Navarra Engineering School in Navarra, Spain. Contact him at nserrano@tecnun.es.

Ismael Giordia is a software engineering professor and a PhD candidate in industrial engineering at the University of Navarra Engineering School. Contact him at icordia@tecnica.com.

Table I

Comparing Bugzilla and ITracker

Features	Bugzilla	ITracker
Is it platform independent?	Mainly for Linux, Windows with Cygwin	Yes, built in J2EE
Is it database independent?	No, works only with MySQL	Yes
How often do practitioners use it?	Often	Sometimes
How customizable is it?	Low: mainly just for presentation	High: allows user-defined fields and reports
Are the number of users limited?	No	No
What is the lifecycle cost?	Medium cost	Medium-low cost
Does it allow for distributed teams?	Yes, Web interface	Yes, Web interface

Hints for Bug Tracking

- Before deciding to install a bug-tracking program on your system, try it out on the Internet. For example, with Mozilla, try http://bugzilla.mozilla.org/enter_bug.cgi.
- The correct input of each bug is as important or even more important than the bug-tracking system itself. The programmer must be able to reproduce the bug that the user has reported.
- Build your project structure and modules well to make it easier for the user and the programmer to identify bugs.
- It's more positive and realistic to think about issues rather than just bugs. If you talk only about bugs, your external viewers will count enhancements as bugs.
- Don't overuse the "send mail to the programmer" feature to input bugs and issues. The programmer might end up defining the mails as spam.
- Decide whether you want to use your application as a bug-tracking system, a task registry, or both. If both, you might be better off using a project management system.
- Show your lead users how to introduce effective bug descriptions so that they can debug successfully and explain them well to other users.
- Use the system often to check for new issue reports; otherwise, your customers won't use it. Reply quickly to customers so that they continue to use the system instead of making a phone call to report a bug.
- Use the tool to get information, not to keep tabs on your workers' behavior; otherwise, they won't input true data.



Become a reviewer for
IEEE Software

The key to providing you quality information you can trust is *IEEE Software's* peer review process. Each article we publish must meet the technical and editorial standards of industry professionals like you. Volunteer as a reviewer and become part of the process.

Become an IEEE Software reviewer today!
 Find out how at www.computer.org/software.