

# The Bugzilla Guide

**Matthew P. Barnson**

barnboy@trilobyte.net

This is the documentation for Bugzilla, the Mozilla bug-tracking system.

---

# Table of Contents

<b><u>Chapter 1. About This Guide</u></b> .....	<b>1</b>
<u>1.1. Purpose and Scope of this Guide</u> .....	1
<u>1.2. Copyright Information</u> .....	1
<u>1.3. Disclaimer</u> .....	2
<u>1.4. New Versions</u> .....	2
<u>1.5. Credits</u> .....	2
<u>1.6. Contributors</u> .....	3
<u>1.7. Feedback</u> .....	3
<u>1.8. Translations</u> .....	3
<u>1.9. Document Conventions</u> .....	3
<b><u>Chapter 2. Installing Bugzilla</u></b> .....	<b>5</b>
<u>2.1. UNIX Installation</u> .....	5
<u>2.2. Win32 (Win98+/NT/2K) Installation</u> .....	5
<u>2.2.1. Win32 Installation: Step-by-step</u> .....	5
<u>2.2.2. Additional Windows Tips</u> .....	7
<b><u>Chapter 3. Administering Bugzilla</u></b> .....	<b>8</b>
<u>3.1. Post-Installation Checklist</u> .....	8
<u>3.2. User Administration</u> .....	9
<u>3.2.1. Creating the Default User</u> .....	9
<u>3.2.2. Managing Other Users</u> .....	9
<u>3.2.2.1. Logging In</u> .....	9
<u>3.2.2.2. Creating new users</u> .....	10
<u>3.2.2.3. Disabling Users</u> .....	10
<u>3.2.2.4. Modifying Users</u> .....	10
<u>3.3. Product, Component, Milestone, and Version Administration</u> .....	12
<u>3.3.1. Products</u> .....	12
<u>3.3.2. Components</u> .....	13
<u>3.3.3. Versions</u> .....	13
<u>3.3.4. Milestones</u> .....	14
<u>3.3.5. Voting</u> .....	15
<u>3.3.6. Groups and Group Security</u> .....	16
<u>3.4. Bugzilla Security</u> .....	18
<b><u>Chapter 4. Using Bugzilla</u></b> .....	<b>20</b>
<u>4.1. What is Bugzilla?</u> .....	20
<u>4.2. Why Should We Use Bugzilla?</u> .....	20
<u>4.3. How do I use Bugzilla?</u> .....	21
<u>4.3.1. Create a Bugzilla Account</u> .....	21
<u>4.3.2. The Bugzilla Query Page</u> .....	22
<u>4.3.3. Creating and Managing Bug Reports</u> .....	25
<u>4.3.3.1. Writing a Great Bug Report</u> .....	25
<u>4.3.3.2. Managing your Bug Reports</u> .....	26
<u>4.4. What's in it for me?</u> .....	26
<u>4.4.1. Account Settings</u> .....	26
<u>4.4.2. Email Settings</u> .....	27
<u>4.4.2.1. Email Notification</u> .....	27

# Table of Contents

4.4.2.2. <a href="#">New Email Technology</a> .....	27
4.4.2.3. <a href="#">"Watching" Users</a> .....	27
4.4.3. <a href="#">Page Footer</a> .....	28
4.4.4. <a href="#">Permissions</a> .....	28
4.5. <a href="#">Using Bugzilla–Conclusion</a> .....	28
<b><a href="#">Chapter 5. Integrating Bugzilla with Third–Party Tools</a>.....</b>	<b>29</b>
5.1. <a href="#">Bonsai</a> .....	29
5.2. <a href="#">CVS</a> .....	29
5.3. <a href="#">Perforce SCM</a> .....	29
5.4. <a href="#">Tinderbox</a> .....	29
<b><a href="#">Chapter 6. The Future of Bugzilla</a>.....</b>	<b>30</b>
6.1. <a href="#">Reducing Spam</a> .....	30
6.2. <a href="#">Better Searching</a> .....	31
6.3. <a href="#">Description Flags and Tracking Bugs</a> .....	32
6.4. <a href="#">Bug Issues</a> .....	33
6.5. <a href="#">Database Integrity</a> .....	35
6.6. <a href="#">Bugzilla 3.0</a> .....	35
1. <a href="#">General Questions</a> .....	36
2. <a href="#">Red Hat Bugzilla</a> .....	38
3. <a href="#">Loki Bugzilla (AKA Fenris)</a> .....	40
4. <a href="#">Pointy–Haired–Boss Questions</a> .....	43
5. <a href="#">Bugzilla Installation</a> .....	43
6. <a href="#">Bugzilla Security</a> .....	47
7. <a href="#">Bugzilla Email</a> .....	47
8. <a href="#">Bugzilla Database</a> .....	47
9. <a href="#">Bugzilla and Win32</a> .....	48
10. <a href="#">Bugzilla Usage</a> .....	50
11. <a href="#">Bugzilla Hacking</a> .....	52
<b><a href="#">Appendix A. The Bugzilla FAQ</a>.....</b>	<b>53</b>
<b><a href="#">Appendix B. The Bugzilla Database</a>.....</b>	<b>54</b>
B.1. <a href="#">Database Schema Chart</a> .....	54
B.2. <a href="#">MySQL Bugzilla Database Introduction</a> .....	55
B.3. <a href="#">MySQL Permissions &amp; Grant Tables</a> .....	62
B.4. <a href="#">Cleaning up after mucking with Bugzilla</a> .....	66
<b><a href="#">Appendix C. Useful Patches and Utilities for Bugzilla</a>.....</b>	<b>67</b>
C.1. <a href="#">The setperl.pl Utility</a> .....	67
C.2. <a href="#">Command–line Bugzilla Queries</a> .....	67
C.3. <a href="#">The Quicksearch Utility</a> .....	68
<b><a href="#">Appendix D. GNU Free Documentation License</a>.....</b>	<b>69</b>
0. <a href="#">PREAMBLE</a> .....	69
1. <a href="#">APPLICABILITY AND DEFINITIONS</a> .....	69
2. <a href="#">VERBATIM COPYING</a> .....	70

# Table of Contents

<a href="#">3. COPYING IN QUANTITY</a>	70
<a href="#">4. MODIFICATIONS</a>	71
<a href="#">5. COMBINING DOCUMENTS</a>	72
<a href="#">6. COLLECTIONS OF DOCUMENTS</a>	72
<a href="#">7. AGGREGATION WITH INDEPENDENT WORKS</a>	73
<a href="#">8. TRANSLATION</a>	73
<a href="#">9. TERMINATION</a>	73
<a href="#">10. FUTURE REVISIONS OF THIS LICENSE</a>	73
<a href="#">How to use this License for your documents</a>	74
<a href="#">Glossary</a>	75
<a href="#">A</a>	76
<a href="#">B</a>	77
<a href="#">L</a>	78
<a href="#">P</a>	79
<a href="#">Q</a>	80
<a href="#">R</a>	81
<a href="#">Z</a>	82

# Chapter 1. About This Guide

## 1.1. Purpose and Scope of this Guide

This document was started on September 17, 2000 by Matthew P. Barnson after a great deal of procrastination updating the Bugzilla FAQ, which I left untouched for nearly half a year. After numerous complete rewrites and reformatting, it is the document you see today.

Despite the lack of updates, Bugzilla is simply the best piece of bug-tracking software the world has ever seen. This document is intended to be the comprehensive guide to the installation, administration, maintenance, and use of the Bugzilla bug-tracking system.

This release of the Bugzilla Guide is the *2.11* release. It is so named that it may match the current version of Bugzilla. The numbering tradition stems from that used for many free software projects, in which *even-numbered* point releases (1.2, 1.14, etc.) are considered "stable releases", intended for public consumption; on the other hand, *odd-numbered* point releases (1.3, 2.09, etc.) are considered unstable *development* releases intended for advanced users, systems administrators, developers, and those who enjoy a lot of pain.

Newer revisions of the Bugzilla Guide will follow the numbering conventions of the main-tree Bugzilla releases, available at [Mozilla.org](http://Mozilla.org), with the exception that intermediate releases will have a minor revision number following a period. For instance, if the current version of Bugzilla is 4.2, the current "stable" version of the Bugzilla guide, in, say, it's fifth revision, would be numbered "4.2.5". Got it? Good.

I wrote this in response to the enormous demand for decent Bugzilla documentation. I have incorporated instructions from the Bugzilla README, Frequently Asked Questions, Database Schema Document, and various mailing lists to create it. Chances are, there are glaring errors in this documentation; please contact [<barnboy@trilobyte.net>](mailto:barnboy@trilobyte.net) to correct them.

---

## 1.2. Copyright Information

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

—Copyright (c) 2000–2001 Matthew P. Barnson

If you have any questions regarding this document, its' copyright, or publishing this document in non-electronic form, please contact [<barnboy@trilobyte.net>](mailto:barnboy@trilobyte.net)

---

## 1.3. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples, and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies that may damage your system. Use of this document may cause your girlfriend to leave you, your cats to pee on your furniture and clothing, your computer to cease functioning, your boss to fire you, and global thermonuclear war. Proceed with caution.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. In particular, I like to put down Microsoft(tm). Live with it.

Naming of particular products or brands should not be seen as endorsements, with the exception of the term "GNU/Linux". Use GNU/Linux. Love it. Bathe with it. It is life and happiness. I endorse it wholeheartedly and encourage you to do the same.

You are strongly recommended to make a backup of your system before installing Bugzilla and at regular intervals thereafter. Heaven knows it's saved my bacon time after time; if you implement any suggestion in this Guide, implement this one!

Bugzilla has not undergone a complete security review. Security holes probably exist in the code. Great care should be taken both in the installation and usage of this software. Carefully consider the implications of installing other network services with Bugzilla.

---

## 1.4. New Versions

This is the initial release of the Bugzilla Guide.

This document can be found in the following places:

- [TriloBYTE](#)
- [Mozilla.org](#)
- [The Linux Documentation Project](#)

The latest version of this document can be checked out via CVS. Please follow the instructions available at [the Mozilla CVS page](#), and check out the mozilla/webtools/bugzilla/docs/ branch.

---

## 1.5. Credits

The people listed below have made enormous contributions to the creation of this Guide, through their dedicated hacking efforts, numerous e-mail and IRC support sessions, and overall excellent contribution to the Bugzilla community:

[Terry Weissman](#) for initially converting Bugzilla from BugSplat! and writing the README upon which this documentation is largely based.

[Tara Hernandez](#) for keeping Bugzilla development going strong after Terry left Mozilla.org

[Dave Lawrence](#) for providing insight into the key differences between Red Hat's customized Bugzilla, and being largely responsible for the "Red Hat Bugzilla" appendix

[Dawn Endico](#) for being a hacker extraordinaire and putting up with my incessant questions and arguments on irc.mozilla.org in #mozwebtools

Last but not least, all the members of the [netscape.public.mozilla.webtools](#) newsgroup. Without your discussions, insight, suggestions, and patches, this could never have happened.

---

## 1.6. Contributors

Thanks go to these people for significant contributions to this documentation:

Zach Lipton, Andrew Pearson, Spencer Smith, Eric Hansen

---

## 1.7. Feedback

I welcome feedback on this document. Without your submissions and input, this Guide cannot continue to exist. Please mail additions, comments, criticisms, etc. to <[barnboy@trilobyte.net](mailto:barnboy@trilobyte.net)>. Please send flames to <[devnull@localhost](mailto:devnull@localhost)>

---

## 1.8. Translations

The Bugzilla Guide needs translators! Please volunteer your translation into the language of your choice. If you will translate this Guide, please notify the members of the mozilla-webtools mailing list at <[mozilla-webtools@mozilla.org](mailto:mozilla-webtools@mozilla.org)>

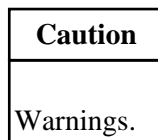
---

## 1.9. Document Conventions

This document uses the following conventions

### Descriptions Appearance

Warnings



Hint

**Tip:**  
Hint.

Notes

**Note:** Note.

Information requiring special attention

Warning
Warning.

File Names `file.extension` Directory Names `directory` Commands to be typed **command** Applications  
Names `application` *Prompt* of users command under bash shell `bash$` *Prompt* of root users command under bash  
shell `bash#` *Prompt* of user command under tcsh shell `tcsh$` Environment Variables `VARIABLE` Emphasized  
`word` `word` Code Example

`<para>Beginning and end of paragraph</para>`

---



# Chapter 2. Installing Bugzilla

## 2.1. UNIX Installation

**Note:** Please consult the README included with the Bugzilla distribution as the current canonical source for UNIX installation instructions. We do, however, have some installation notes for errata from the README.

**Note:** If you are installing Bugzilla on S.u.S.e. Linux, or some other distributions with "paranoid" security options, it is possible that the checksetup.pl script may fail with the error: cannot chdir(/var/spool/mqueue): Permission denied This is because your /var/spool/mqueue directory has a mode of "drwx-----". Type **chmod 755 /var/spool/mqueue** as root to fix this problem.

**Note:**

---

## 2.2. Win32 (Win98+/NT/2K) Installation

These directions have *not* been extensively tested. We need testers! Please try these out and post any changes to the newsgroup.

---

### 2.2.1. Win32 Installation: Step-by-step

**Note:** You should be familiar with, and cross-reference, the UNIX README while performing your Win32 installation. Unfortunately, Win32 directions are not yet as detailed as those for UNIX.

The *most critical* difference for Win32 users is the lack of support for a crypt() function in MySQL for Windows. It does not have it! All ENCRYPT statements must be modified.

1. Install [Apache Web Server](#) for Windows.

**Note:** You may also use Internet Information Server or Personal Web Server for this purpose. However, setup is slightly more difficult. If ActivePerl doesn't seem to handle your file associations correctly (for .cgi and .pl files), please consult the FAQ, in the "Win32" section.

If you are going to use IIS, if on Windows NT you must be updated to at least Service Pack 4.

- Install [ActivePerl](#)

Please also check the following links to fully understand the status of ActivePerl on Win32: [Perl Porting](#), and [Hixie Click Here](#)

- Use ppm from your perl\bin directory to install the following packs: DBI, DBD-Mysql, TimeDate, Chart, Date-Calc, Date-Manip, and GD. You may need to extract them from .zip format using Winzip or other unzip program first. These additional ppm modules can be downloaded from ActiveState.

The syntax for ppm is: `C:> ppm install <module>.ppd`

You can find ActiveState ppm modules at <http://www.activestate.com/PPMPackages/5.6plus>.

- Download and install the Windows GNU tools from [www.cygwin.com](http://www.cygwin.com). Make sure the GNU utilities are in your \$PATH.
- Install MySQL for NT.

**Note:** Your configuration file for MySQL *must* be named C:\MY.CNF.

- Setup MySQL

```
a. C:> C:\mysql\bin\mysql -u root mysql
b. mysql> DELETE FROM user WHERE Host='localhost' AND User='';
c. mysql> UPDATE user SET Password=PASSWORD ('new_password') WHERE
  user='root';
d. mysql> GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER,
  CREATE, DROP, REFERENCES ON bugs.* to bugs@localhost IDENTIFIED
  BY 'bugs_password';
e. mysql> FLUSH PRIVILEGES;
f. mysql> create database bugs;
g. mysql> exit
h. C:> C:\mysql\bin\mysqladmin -u root -p reload
```

- Configure Bugzilla. For Win32, this involves editing "defparams.pl" and "localconfig" to taste. Running "checksetup.pl" should create localconfig for you. Note that getgrnam() doesn't work, and should be deleted. Change this line: "my \$webservergid = getgrnam(\$my\_webservergroup);" to "my \$webservergid = \$my\_webservergroup;"

- 

**Note:** There are several alternatives to Sendmail that will work on Win32. The one mentioned here is a *suggestion*, not a requirement. Some other mail packages that can work include [BLAT](#), [Windmail](#), [Mercury Sendmail](#), and the CPAN Net::SMTP Perl module (available in .ppm). Every option requires some hacking of the Perl scripts for Bugzilla to make it work. The option here simply requires the least.

Download NTsendmail, available from [www.ntsendsmail.com](http://www.ntsendsmail.com). In order for it to work, you must set up some new environment variables (detailed on the ntsendsmail home page). Figuring out where to put those variables is left as an exercise for the reader. You must have a "real" mail server which allows you to relay off it in your \$ENV{"NTsendmail"} (which you should probably place in globals.pl)

Once downloaded and installed, modify all open(SENDMAIL) calls to open "| c:\ntsendsmail\ntsendsmail -t" instead of "|usr/lib/sendmail -t".

**Note:** We need someone to test this and make sure this works as advertised.

- Modify globals.pl and CGI.pl to remove the word "encrypt".

**Note:** I'm not sure this is all that is involved to remove crypt. Any NT Bugzilla hackers want to pipe up?

- Change all references to "processmail" to "processmail.pl" in all files, and rename "processmail" to "processmail.pl"

**Note:** I really think this may be a change we want to make for main-tree Bugzilla. It's painless for the UNIX folks, and will make the Win32 people happier.

- Modify the path to perl on the first line (!) of all files to point to your Perl installation, and add "perl" to the beginning of all Perl system calls that use a perl script as an argument. This may take you a while. There is a "setperl.pl" utility to speed part of this procedure, available in the "Patches and Utilities" section of The Bugzilla Guide.
  - In processmail.pl, add "binmode(HANDLE)" before all read() calls. This may not be necessary, but in some cases the read() under Win32 doesn't count the EOL's without using a binary read().
- 

### 2.2.2. Additional Windows Tips

**Tip:** From Andrew Pearson:

"You can make Bugzilla work with Personal Web Server for Windows 98 and higher, as well as for IIS 4.0. Microsoft has information available at <http://support.microsoft.com/support/kb/articles/Q231/9/98.ASP>

Basically you need to add two String Keys in the registry at the following location:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ScriptMap

The keys should be called ".pl" and ".cgi", and both should have a value something like:  
**c:/perl/bin/perl.exe "%s" "%s"**

The KB article only talks about .pl, but it goes into more detail and provides a perl test script.

---

# Chapter 3. Administering Bugzilla

Or, I just got this cool thing installed. Now what the heck do I do with it?

So you followed the README instructions to the letter, and just logged into bugzilla with your super-duper god account and you are sitting at the query screen. Yet, you have nothing to query. Your first act of business needs to be to setup the operating parameters for bugzilla.

---

## 3.1. Post-Installation Checklist

After installation, follow the checklist below to ensure that you have a successful installation. If you do not see a recommended setting for a parameter, consider leaving it at the default while you perform your initial tests on your Bugzilla setup.

1. Set "maintainer" to *your* email address. This allows Bugzilla's error messages to display your email address and allow people to contact you for help.
2. Set "urlbase" to the URL reference for your Bugzilla installation. If your bugzilla query page is at <http://www.foo.com/bugzilla/query.cgi>, your url base is <http://www.foo.com/bugzilla/>
3. Set "usebuggroups" to "1" *only* if you need to restrict access to products. I suggest leaving this parameter *off* while initially testing your Bugzilla.
4. Set "usebuggroupsentry" to "1" if you want to be able to restrict access to products. Once again, if you are simply testing your installation, I suggest against turning this parameter on; the strict security checking may stop you from being able to modify your new entries.
5. Set "shadowdb" to "bug\_shadowdb" if you will be running a *very* large installation of Bugzilla. The shadow database enables many simultaneous users to read and write to the database without interfering with one another.

**Note:** Enabling "shadowdb" can adversely affect the stability of your installation of Bugzilla. You may frequently need to manually synchronize your databases, or schedule nightly syncs via "cron"

Once again, in testing you should avoid this option — use it if or when you *need* to use it, and have repeatedly run into the problem it was designed to solve — very long wait times while attempting to commit a change to the database.

If you use the "shadowdb" option, it is only natural that you should turn the "queryagainstshadowdb" option "On" as well. Otherwise you are replicating data into a shadow database for no reason!

- If you have custom logos or HTML you must put in place to fit within your site design guidelines, place the code in the "headerhtml", "footerhtml", "errorhtml", "bannerhtml", or "blurbhtml" text boxes.

**Note:** The "headerhtml" text box is the HTML printed out *before* any other code on the page. If you have a special banner, put the code for it in "bannerhtml". You may want to leave these settings at the defaults initially.

- Add any text you wish to the "passwordmail" parameter box. For instance, many people choose to use this box to give a quick training blurb about how to use Bugzilla at your site.
- Set "newemailtech" to "on". Your users will thank you. This is the default in the post-2.12 world.
- Do you want to use the qa contact ("useqacontact") and status whiteboard ("usestatuswhiteboard") fields?

These fields are useful because they allow for more flexibility, particularly when you have an existing Quality Assurance and/or Release Engineering team, but they may not be needed for smaller installations.

- Set "whinedays" to the amount of days you want to let bugs go in the "New" or "Reopened" state before notifying people they have untouched new bugs. If you do not plan to use this feature, simply do not set up the whining cron job described in the README, or set this value to "0".
- Set the "commenton" options according to your site policy. It is a wise idea to require comments when users resolve, reassign, or reopen bugs.

**Note:** It is generally far better to require a developer comment when resolving bugs than not. Few things are more annoying to bug database users than having a developer mark a bug "fixed" without any comment as to what the fix was (or even that it was truly fixed!)

- Set "supportwatchers" to "On". This feature is helpful for team leads to monitor progress in their respective areas, and can offer many other benefits, such as allowing a developer to pick up a former engineer's bugs without requiring her to change all the information in the bug.
- 

## 3.2. User Administration

User administration is one of the easiest parts of Bugzilla. Keeping it from getting out of hand, however, can become a challenge.

---

### 3.2.1. Creating the Default User

When you first run checksetup.pl after installing Bugzilla, it will prompt you for the administrative username (email address) and password for this "super user". If for some reason you were to delete the "super user" account, re-running checksetup.pl will again prompt you for this username and password.

**Tip:** If you wish to add more administrative users, you must use the MySQL interface. Run "mysql" from the command line, and use these commands ("mysql>" denotes the mysql prompt, not something you should type in): **mysql> use bugs; mysql> update profiles set groupset=0x7fffffff where login\_name = "(user's login name)";**

---

### 3.2.2. Managing Other Users

#### 3.2.2.1. Logging In

1. Open the index.html page for your Bugzilla installation in your browser window.
2. Click the "Query Existing Bug Reports" link.
3. Click the "Log In" link at the foot of the page.
4. Type your email address, and the password which was emailed to you when you created your Bugzilla account, into the spaces provided.

Congratulations, you are logged in!

---

### 3.2.2.2. Creating new users

Your users can create their own user accounts by clicking the "New Account" link at the bottom of each page. However, should you desire to create user accounts ahead of time, here is how you do it.

1. After logging in, click the "Users" link at the footer of the query page.
2. To see a specific user, type a portion of their login name in the box provided and click "submit". To see all users, simply click the "submit" button. You must click "submit" here to be able to add a new user.

**Tip:** More functionality is available via the list on the right-hand side of the text entry box. You can match what you type as a case-insensitive substring (the default) of all users on your system, a case-sensitive regular expression (please see the "man regexp" manual page for details on regular expression syntax), or a *reverse* regular expression match, where every user name which does NOT match the regular expression is selected.

- Click the "Add New User" link at the bottom of the user list
- Fill out the form presented. This page is self-explanatory. When done, click "submit".

**Note:** Adding a user this way will *not* send an email informing them of their username and password. In general, it is preferable to log out and use the "New Account" button to create users, as it will pre-populate all the required fields and also notify the user of her account name and password.

---

### 3.2.2.3. Disabling Users

I bet you noticed that big "Disabled Text" entry box available from the "Add New User" screen, when you edit an account? By entering any text in this box and selecting "submit", you have prevented the user from using Bugzilla via the web interface. Your explanation, written in this text box, will be presented to the user the next time she attempts to use the system.

Warning
Don't disable your own administrative account, or you will hate life!

### 3.2.2.4. Modifying Users

Here I will attempt to describe the function of each option on the user edit screen.

- *Login Name:* This is generally the user's email address. However, if you have edited your system parameters, this may just be the user's login name or some other identifier.

**Tip:** For compatibility reasons, you should probably stick with email addresses as user login names. It will make your life easier.

- *Real Name:* Duh!

## The Bugzilla Guide

- *Password*: You will only see asterisks in versions of Bugzilla newer than 2.10 or early 2.11. You can change the user password [here](#).
- *Email Notification*: You may choose from one of three options:
  1. All qualifying bugs except those which I change: The user will be notified of any change to any bug for which she is the reporter, assignee, Q/A contact, CC recipient, or "watcher".
  2. Only those bugs which I am listed on the CC line: The user will not be notified of changes to bugs where she is the assignee, reporter, or Q/A contact, but will receive them if she is on the CC list.

**Note:** She will still receive whining cron emails if you set up the "whinemail" feature.
  3. *All Qualifying Bugs*: This user is a glutton for punishment. If her name is in the reporter, Q/A contact, CC, assignee, or is a "watcher", she will get email updates regarding the bug.

*Disable Text*: If you type anything in this box, including just a space, the user account is disabled from making any changes to bugs via the web interface, and what you type in this box is presented as the reason.

Warning
Don't disable the administrator account!

**Note:** As of this writing, the user can still submit bugs via the e-mail gateway, if you set it up, despite the disabled text field. The e-mail gateway should *not* be enabled for secure installations of Bugzilla.

- *CanConfirm*: This field is only used if you have enabled "unconfirmed" status in your parameters screen. If you enable this for a user, that user can then move bugs from "Unconfirmed" to "Confirmed" status (ergo: "New" status). Be judicious about allowing users to turn this bit on for other users.
- *Creategroups*: This option will allow a user to create and destroy groups in Bugzilla. Unless you are using the Bugzilla GroupSentry security option "usebuggroupentry" in your parameters, this setting has no effect.
- *Editbugs*: Unless a user has this bit set, they can only edit those bugs for which they are the assignee or the reporter.

**Note:** Leaving this option unchecked does not prevent users from adding comments to a bug! They simply cannot change a bug priority, severity, etc. unless they are the assignee or reporter.

- *Editcomponents*: This flag allows a user to create new products and components, as well as modify and destroy those that have no bugs associated with them. If a product or component has bugs associated with it, those bugs must be moved to a different product or component before Bugzilla will allow them to be destroyed. The name of a product or component can be changed without affecting the associated bugs, but it tends to annoy the hell out of your users when these change a lot.
- *Editkeywords*: If you use Bugzilla's keyword functionality, enabling this feature allows a user can create and destroy keywords. As always, the keywords for existing bugs containing the keyword the user wishes to destroy must be changed before Bugzilla will allow it to die. You must be very careful about creating too many new keywords if you run a very large Bugzilla installation; keywords are global variables across products, and you can often run into a phenomenon called "keyword bloat". This confuses users, and then the feature goes unused.
- *Editusers*: This flag allows a user do what you're doing right now: edit other users. This will allow those with the right to do so to remove administrator priveleges from other users or grant them to themselves.

Enable with care.

- **PRODUCT**: PRODUCT bugs access. This allows an administrator, with product-level granularity, to specify in which products a user can edit bugs. The user must still have the "editbugs" privilege to edit bugs in this area; this simply restricts them from even seeing bugs outside these boundaries if the administrator has enabled the group sentry parameter "usebuggroupsentry". Unless you are using bug groups, this option has no effect.
- 

## 3.3. Product, Component, Milestone, and Version Administration

*Dear Lord, we have to get our users to do WHAT?*

---

### 3.3.1. Products

Formerly, and in some spots still, called "Programs"

[Products](#) are the broadest category in Bugzilla, and you should have the least of these. If your company makes computer games, you should have one product per game, and possibly a few special products (website, meetings...)

A Product (formerly called "Program", and still referred to that way in some portions of the source code) controls some very important functions. The number of "votes" available for users to vote for the most important bugs is set per-product, as is the number of votes required to move a bug automatically from the UNCONFIRMED status to the NEW status. One can close a Product for further bug entry and define various Versions available from the Edit Product screen.

To create a new product:

1. Select "components" from the yellow footer

**Tip:** It may seem counterintuitive to click "components" when you want to edit the properties associated with Products. This is one of a long list of things we want in Bugzilla 3.0...

- Select the "Add" link to the right of "Add a new product".
- Enter the name of the product and a description. The Description field is free-form.

**Tip:** Don't worry about the "Closed for bug entry", "Maximum Votes per person", "Maximum votes a person can put on a single bug", "Number of votes a bug in this Product needs to automatically get out of the UNCONFIRMED state", and "Version" options yet. We'll cover those in a few moments.

---



## 3.3.2. Components

Components are subsections of a Product.

### Example 3–1. Creating some Components

The computer game you are designing may have a "UI" component, an "API" component, a "Sound System" component, and a "Plugins" component, each overseen by a different programmer. It often makes sense to divide Components in Bugzilla according to the natural divisions of responsibility within your Product or company.

Each component has an owner and (if you turned it on in the parameters), a QA contact. The owner should be the primary person who fixes bugs in that component. The QA Contact should be the person who will ensure these bugs are completely fixed. The Owner, QA Contact, and Reporter will get email when new bugs are created in this Component and when these bugs change. Default Owner and Default QA Contact fields only dictate the *default assignments*; the Owner and Q/A Contact fields in a bug are otherwise unrelated to the Component.

To create a new Component:

1. Select the "Edit components" link from the "Edit Product" page
2. Select the "Add" link to the right of the "Add a new component" text on the "Select Component" page.
3. Fill out the "Component" field, a short "Description", and the "Initial Owner". The "Component" field should not contain a space. The "Description" field is free-form. The "Initial Owner" field must be that of a valid user already existing in the database. If the initial owner does not exist, Bugzilla will refuse to create the component.

**Tip:** Is your "Default Owner" a user who is not yet in the database? No problem.

- a. Select the "Log out" link on the footer of the page.
  - b. Select the "New Account" link on the footer of the "Relogin" page
  - c. Type in the email address of the default owner you want to create in the "E-mail address" field, and her full name in the "Real name" field, then select the "Submit Query" button.
  - d. Now select "Log in" again, type in your login information, and you can modify the product to use the Default Owner information you require.
4. Either "edit" more components or return to the "query" page on the ensuing "Adding new component" page. To return to the Product you were editing, you must select the "components" link as before.
- 

## 3.3.3. Versions

Versions are the revisions of the product, such as "Flinders 3.1", "Flinders 95", and "Flinders 2000". Using Versions helps you isolate code changes and are an aid in reporting.

### Example 3–2. Common Use of Versions

A user reports a bug against Version "Beta 2.0" of your product. The current Version of your software is "Release Candidate 1", and no longer has the bug. This will help you triage and classify bugs according to their relevance. It is also possible people may report bugs against bleeding–edge beta versions that are not evident in older versions of the software. This can help isolate code changes that caused the bug

### Example 3–3. A Different Use of Versions

This field has been used to good effect by an online service provider in a slightly different way. They had three versions of the product: "Production", "QA", and "Dev". Although it may be the same product, a bug in the development environment is not normally as critical as a Production bug, nor does it need to be reported publicly. When used in conjunction with Target Milestones, one can easily specify the environment where a bug can be reproduced, and the Milestone by which it will be fixed.

To create and edit Versions:

1. From the "Edit Product" screen, select "Edit Versions"
2. You will notice that the product already has the default version "undefined". If your product doesn't use version numbers, you may want to leave this as it is or edit it so that it is "----". You can then go back to the edit versions page and add new versions to your product.

Otherwise, click the "Add" button to the right of the "Add a new version" text.

3. Enter the name of the Version. This can be free–form characters up to the limit of the text box. Then select the "Add" button.
4. At this point you can select "Edit" to edit more Versions, or return to the "Query" page, from which you can navigate back to the product through the "components" link at the foot of the Query page.

---

## 3.3.4. Milestones

Milestones are "targets" that you plan to get a bug fixed by. For example, you have a bug that you plan to fix for your 3.0 release, it would be assigned the milestone of 3.0. Or, you have a bug that you plan to fix for 2.8, this would have a milestone of 2.8.

**Note:** Milestone options will only appear for a Product if you turned the "usetargetmilestone" field in the "Edit Parameters" screen "On".

To create new Milestones, set Default Milestones, and set Milestone URL:

1. Select "edit milestones"
2. Select "Add" to the right of the "Add a new milestone" text
3. Enter the name of the Milestone in the "Milestone" field. You can optionally set the "Sortkey", which is a positive or negative number (–255 to 255) that defines where in the list this particular milestone appears. Select "Add".

### Example 3–4. Using SortKey with Target Milestone

Let's say you create a target milestone called "Release 1.0", with Sortkey set to "0". Later, you realize that you will have a public beta, called "Beta1". You can create a Milestone called "Beta1", with a Sortkey of "-1" in order to ensure people will see the Target Milestone of "Beta1" earlier on the list than "Release 1.0"

4. If you want to add more milestones, select the "Edit" link. If you don't, well shoot, you have to go back to the "query" page and select "components" again, and make your way back to the Product you were editing.

**Note:** This is another in the list of unusual user interface decisions that we'd like to get cleaned up. Shouldn't there be a link to the effect of "edit the Product I was editing when I ended up here"? In any case, clicking "components" in the footer takes you back to the "Select product" screen, from which you can begin editing your product again.

5. From the Edit Product screen again (once you've made your way back), enter the URL for a description of what your milestones are for this product in the "Milestone URL" field. It should be of the format "http://www.foo.com/bugzilla/product\_milestones.html"

Some common uses of this field include product descriptions, product roadmaps, and of course a simple description of the meaning of each milestone.

6. If you're using Target Milestones, the "Default Milestone" field must have some kind of entry. If you really don't care if people set coherent Target Milestones, simply leave this at the default, "----". However, controlling and regularly updating the Default Milestone field is a powerful tool when reporting the status of projects.

Select the "Update" button when you are done.

---

7.

## 3.3.5. Voting

The concept of "voting" is a poorly understood, yet powerful feature for the management of open-source projects. Each user is assigned so many Votes per product, which they can freely reassign (or assign multiple votes to a single bug). This allows developers to gauge user need for a particular enhancement or bugfix. By allowing bugs with a certain number of votes to automatically move from "UNCONFIRMED" to "NEW", users of the bug system can help high-priority bugs garner attention so they don't sit for a long time awaiting triage.

The daunting challenge of Votes is deciding where you draw the line for a "vocal majority". If you only have a user base of 100 users, setting a low threshold for bugs to move from UNCONFIRMED to NEW makes sense. As the Bugzilla user base expands, however, these thresholds must be re-evaluated. You should gauge whether this feature is worth the time and close monitoring involved, and perhaps forego implementation until you have a critical mass of users who demand it.

To modify Voting settings:

1. Navigate to the "Edit Product" screen for the Product you wish to modify

2. Set "Maximum Votes per person" to your calculated value. Setting this field to "0" disables voting.
3. Set "Maximum Votes a person can put on a single bug" to your calculated value. It should probably be some number lower than the "Maximum votes per person". Setting this field to "0" disables voting, but leaves the voting options open to the user. This is confusing.
4. Set "Number of votes a bug in this product needs to automatically get out of the UNCONFIRMED state" to your calculated number. Setting this field to "0" disables the automatic move of bugs from UNCONFIRMED to NEW. Some people advocate leaving this at "0", but of what use are Votes if your Bugzilla user base is unable to affect which bugs appear on Development radar?

**Tip:** You should probably set this number to higher than a small coalition of Bugzilla users can influence it. Most sites use this as a "referendum" mechanism — if users are able to vote a bug out of UNCONFIRMED, it is a *really* bad bug!

- Once you have adjusted the values to your preference, select the "Update" button.
- 

### 3.3.6. Groups and Group Security

Groups can be very useful in bugzilla, because they allow users to isolate bugs or products that should only be seen by certain people. Groups can also be a complicated minefield of interdependencies and weirdness if mismanaged.

#### Example 3–5. When to Use Group Security

Many Bugzilla sites isolate "Security–related" bugs from all other bugs. This way, they can have a fix ready before the security vulnerability is announced to the world. You can create a "Security" product which, by default, has no members, and only add members to the group (in their individual User page, as described under User Administration) who should have privileged access to "Security" bugs. Alternately, you may create a Group independently of any Product, and change the Group mask on individual bugs to restrict access to members only of certain Groups.

Groups only work if you enable the "usebuggroups" parameter. In addition, if the "usebuggroupentry" parameter is "On", one can restrict access to products by groups, so that only members of a product group are able to view bugs within that product. Group security in Bugzilla can be divided into two categories: Generic and Product–Based.

**Note:** Groups in Bugzilla are a complicated beast that evolved out of very simple user permission bitmasks, apparently itself derived from common concepts in UNIX access controls. A "bitmask" is a fixed–length number whose value can describe one, and only one, set of states. For instance, UNIX file permissions are assigned bitmask values: "execute" has a value of 1, "write" has a value of 2, and "read" has a value of 4. Add them together, and a file can be read, written to, and executed if it has a bitmask of "7". (This is a simplified example — anybody who knows UNIX security knows there is much more to it than this. Please bear with me for the purpose of this note.) The only way a bitmask scheme can work is by doubling the bit count for each value. Thus if UNIX wanted to offer another file permission, the next would have to be a value of 8, then the next 16, the next 32, etc.

Similarly, Bugzilla offers a bitmask to define group permissions, with an internal limit of 64. Several are already occupied by built-in permissions. The way around this limitation is to avoid assigning groups to products if you have many products, avoid bloating of group lists, and religiously prune irrelevant groups. In reality, most installations of Bugzilla support far fewer than 64 groups, so this limitation has not hit for most sites, but it is on the table to be revised for Bugzilla 3.0 because it interferes with the security schemes of some administrators.

To enable Generic Group Security ("usebuggroups"):

1. Turn "On" "usebuggroups" in the "Edit Parameters" screen.
2. You will generally have no groups set up. Select the "groups" link in the footer.
3. Take a moment to understand the instructions on the "Edit Groups" screen. Once you feel confident you understand what is expected of you, select the "Add Group" link.
4. Fill out the "New Name" (remember, no spaces!), "New Description", and "New User RegExp" fields. "New User RegExp" allows you to automatically place all users who fulfill the Regular Expression into the new group.

### Example 3-6. Creating a New Group

I created a group called "DefaultGroup" with a description of "This is simply a group to play with", and a "New User RegExp" of "\*@velio.com". This new group automatically includes all Bugzilla users with "@velio.com" at the end of their user id. When I finished, my new group was assigned bit #128.

When you have finished, select the "Add" button.

To enable Product-Based Group Security ("usebuggroupsenry"):

Warning
Don't forget that you only have 64 groups masks available, total, for your installation of Bugzilla! If you plan on having more than 50 products in your individual Bugzilla installation, and require group security for your products, you should consider either running multiple Bugzillas or using Generic Group Security instead of Product-Based ("usebuggroupsenry") Group Security.

1. Turn "On" "usebuggroups" and "usebuggroupsenry" in the "Edit Parameters" screen.

Warning
"usebuggroupsenry" has the capacity to prevent the administrative user from directly altering bugs because of conflicting group permissions. If you plan on using "usebuggroupsenry", you should plan on restricting administrative account usage to administrative duties only. In other words, manage bugs with an unprivileged user account, and manage users, groups, Products, etc. with the administrative account.

2. You will generally have no Groups set up, unless you enabled "usebuggroupsenry" prior to creating any Products. To create "Generic Group Security" groups, follow the instructions given above. To create Product-Based Group security, simply follow the instructions for creating a new Product. If

you need to add users to these new groups as you create them, you will find the option to add them to the group available under the "Edit User" screens.

---

## 3.4. Bugzilla Security

*Putting your money in a wall safe is better protection than depending on the fact that no one knows that you hide your money in a mayonnaise jar in your fridge.*

**Note:** Poorly-configured MySQL, Bugzilla, and FTP installations have given attackers full access to systems in the past. Please take these guidelines seriously, even for Bugzilla machines hidden away behind your firewall. 80% of all computer trespassers are insiders, not anonymous crackers.

First thing's first: Secure your installation.

**Note:** These instructions must, of necessity, be somewhat vague since Bugzilla runs on so many different platforms. If you have refinements of these directions for specific platforms, please submit them to [mozilla-webtools@mozilla.org](mailto:mozilla-webtools@mozilla.org)

1. Ensure you are running at least MySQL version 3.22.32 or newer. Earlier versions had notable security holes and poorly secured default configuration choices.
2. *There is no substitute for understanding the tools on your system!* Read [The MySQL Privelege System](#) until you can recite it from memory!

At the very least, ensure you password the "mysql -u root" account and the "bugs" account, establish grant table rights (consult the Keystone guide in Appendix C: The Bugzilla Database for some easy-to-use details) that do not allow CREATE, DROP, RELOAD, SHUTDOWN, and PROCESS for user "bugs". I wrote up the Keystone advice back when I knew far less about security than I do now : )

3. Lock down /etc/inetd.conf. Heck, disable inet entirely on this box. It should only listen to port 25 for Sendmail and port 80 for Apache.
4. Do not run Apache as "nobody". This will require very lax permissions in your Bugzilla directories. Run it, instead, as a user with a name, set via your httpd.conf file.
5. Ensure you have adequate access controls for \$BUGZILLA\_HOME/data/ and \$BUGZILLA\_HOME/localconfig. The localconfig file stores your "bugs" user password, which would be terrible to have in the hands of a criminal. Also some files under \$BUGZILLA\_HOME/data store sensitive information.

On Apache, you can use .htaccess files to protect access to these directories, as outlined in [Bug 57161](#) for the localconfig file, and [Bug 65572](#) for adequate protection in your data/ and shadow/ directories.

Note the instructions which follow are Apache-specific. If you use IIS, Netscape, or other non-Apache web servers, please consult your system documentation for how to secure these files from being transmitted to curious users.

## The Bugzilla Guide

Place the following text into a file named ".htaccess", readable by your web server, in your \$BUGZILLA\_HOME/data directory.

```
<Files comments>  
allow from all  
</Files>  
deny from all
```

Place the following text into a file named ".htaccess", readable by your web server, in your \$BUGZILLA\_HOME/ directory.

```
<Files localconfig>  
deny from all  
</Files>  
allow from all
```

Place the following text into a file named ".htaccess", readable by your web server, in your \$BUGZILLA\_HOME/shadow directory.

```
deny from all
```

---

6.

# Chapter 4. Using Bugzilla

*What, Why, How, & What's in it for me?*

---

## 4.1. What is Bugzilla?

Bugzilla is one example of a class of programs called "Defect Tracking Systems", or, more commonly, "Bug-Tracking Systems". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively. At the time Bugzilla was originally written, as a port from Netscape Communications' "Bugsplat!" program to Perl from TCL, there were very few competitors in the market for bug-tracking software. Most commercial defect-tracking software vendors at the time charged enormous licensing fees. Bugzilla quickly became a favorite of the open-source crowd (with its genesis in the open-source browser project, Mozilla) and is now the de-facto standard defect-tracking system against which all others are measured.

Bugzilla has matured immensely, and now boasts many advanced features. These include:

- integrated, product-based granular security schema
- inter-bug dependencies and dependency graphing
- advanced reporting capabilities
- a robust, stable RDBMS back-end
- extensive configurability
- a very well-understood and well-thought-out natural bug resolution protocol
- email, XML, and HTTP APIs
- integration with several automated software configuration management systems
- too many more features to list

Despite its current robustness and popularity, however, Bugzilla faces some near-term challenges, such as reliance on a single database, a lack of abstraction of the user interface and program logic, verbose email bug notifications, a powerful but daunting query interface, little reporting configurability, problems with extremely large queries, some unsupportable bug resolution options, no internationalization, and dependence on some nonstandard libraries.

Despite these small problems, Bugzilla is very hard to beat. It is under *very* active development to address the current issues, and a long-awaited overhaul in the form of Bugzilla 3.0 is expected sometime later this year.

---

## 4.2. Why Should We Use Bugzilla?

*No, Who's on first...*

For many years, defect-tracking software has remained principally the domain of large software development houses. Even then, most shops never bothered with bug-tracking software, and instead simply relied on shared lists and email to monitor the status of defects. This procedure is error-prone and tends to cause those bugs judged least significant by developers to be dropped or ignored



These days, many companies are finding that integrated defect-tracking systems reduce downtime, increase productivity, and raise customer satisfaction with their systems. Along with full disclosure, an open bug-tracker allows manufacturers to keep in touch with their clients and resellers, to communicate about problems effectively throughout the data management chain. Many corporations have also discovered that defect-tracking helps reduce costs by providing IT support accountability, telephone support knowledge bases, and a common, well-understood system for accounting for unusual system or software issues.

But why should *you* use Bugzilla?

Bugzilla is very adaptable to various situations. Known uses currently include IT support queues, Systems Administration deployment management, chip design and development problem tracking (both pre-and-post fabrication), and software bug tracking for luminaries such as Redhat, Loki software, Linux-Mandrake, and VA Systems. Combined with systems such as CVS, Bonsai, or Perforce SCM, Bugzilla provides a powerful, easy-to-use solution to configuration management and replication problems

Bugzilla can dramatically increase the productivity and accountability of individual employees by providing a documented workflow and positive feedback for good performance. How many times do you wake up in the morning, remembering that you were supposed to do *\*something\** today, but you just can't quite remember? Put it in Bugzilla, and you have a record of it from which you can extrapolate milestones, predict product versions for integration, and by using Bugzilla's e-mail integration features be able to follow the discussion trail that led to critical decisions.

Ultimately, Bugzilla puts the power in your hands to improve your value to your employer or business while providing a usable framework for your natural attention to detail and knowledge store to flourish.

---

### 4.3. How do I use Bugzilla?

*Hey! I'm Woody! Howdy, Howdy, Howdy!*

Bugzilla is a large and complex system. Describing how to use it requires some time. If you are only interested in installing or administering a Bugzilla installation, please consult the Installing and Administering Bugzilla portions of this Guide. This section is principally aimed towards developing end-user mastery of Bugzilla, so you may fully enjoy the benefits afforded by using this reliable open-source bug-tracking software.

Throughout this portion of the Guide, we will refer to user account options available at the Bugzilla test installation, [landfill.tequilarista.org](http://landfill.tequilarista.org). Although Landfill serves as a great introduction to Bugzilla, it does not offer all the options you would have as a user on your own installation of Bugzilla, nor can it do more than serve as a general introduction to Bugzilla. However, please use it if you want to follow this tutorial.

---

#### 4.3.1. Create a Bugzilla Account

First thing's first! If you want to use Bugzilla, first you need to create an account. Consult with the administrator responsible for your installation of Bugzilla for the URL you should use to access it. If you're test-driving the end-user Bugzilla experience, use this URL: <http://landfill.tequilarista.org/mozilla/bugzilla/>

1. Click the "Open a new Bugzilla account" link.
2. Enter your "E-mail address" and "Real Name" (or whatever name you want to call yourself) in the spaces provided, then select the "Create Account" button.
3. Within 5–10 minutes, you should receive an email to the address you provided above, which contains your login name (generally the same as the email address), and a password you can use to access your account. This password is randomly generated, and should be changed at your nearest opportunity (we'll go into how to do it later).
4. Click the "Log In" link in the yellow area at the bottom of the page in your browser, then enter your "E-mail address" and "Password" you just received into the spaces provided, and select "Login".

**Note:** If you ever forget your password, you can come back to this page, enter your "E-mail address", then select the "E-mail me a password" button to have your password mailed to you again so that you can login.

### Caution

Many modern browsers include an "Auto-Complete" or "Form Fill" feature to remember the user names and passwords you type in at many sites. Unfortunately, sometimes they attempt to "guess" what you will put in as your password, and guess wrong. If you notice a text box is already filled out, please overwrite the contents of the text box so you can be sure to input the correct information.

Congratulations! If you followed these directions, you now are the proud owner of a user account on [landfill.tequilarista.org](http://landfill.tequilarista.org) (Landfill) or your local Bugzilla install. You should now see in your browser a page called the "Bugzilla Query Page". It may look daunting, but with this Guide to walk you through it, you will master it in no time.

---

## 4.3.2. The Bugzilla Query Page

The Bugzilla Query Page is the heart and soul of Bugzilla. It is the master interface where you can find any bug report, comment, or patch currently in the Bugzilla system. We'll go into how to create your own bug report later on.

There are efforts underway to simplify query usage. If you have a local installation of Bugzilla 2.12 or higher, you should have "quicksearch.html" available to use and simplify your searches. There is also, or shortly will be, a helper for the query interface, called "queryhelp.cgi". Landfill tends to run the latest code, so these two utilities should be available there for your perusal.

At this point, please visit the main Bugzilla site, [bugzilla.mozilla.org](http://bugzilla.mozilla.org), to see a more fleshed-out query page.

The first thing you need to notice about the Bugzilla Query Page is that nearly every box you see on your screen has a hyperlink nearby, explaining what it is or what it does. Near the upper-left-hand corner of your browser window you should see the word "Status" underlined. Select it.

Notice the page that popped up? Every underlined word you see on your screen is a hyperlink that will take you to context-sensitive help. Click around for a while, and learn what everything here does. To return to the query interface after pulling up a help page, use the "Back" button in your browser.

## The Bugzilla Guide

I'm sure that after checking out the online help, you are now an Expert on the Bugzilla Query Page. If, however, you feel you haven't mastered it yet, let me walk you through making a few successful queries to find out what there are in the Bugzilla bug-tracking system itself.

1. Ensure you are back on the "Bugzilla Query Page" Do nothing in the boxes marked "Status", "Resolution", "Platform", "OpSys", "Priority", or "Severity". The default query for "Status" is to find all bugs that are NEW, ASSIGNED, or REOPENED, which is what we want. If you don't select anything in the other 5 scrollboxes there, then you are saying that "any of these are OK"; we're not locking ourselves into only finding bugs on the "DEC" Platform, or "Windows 95" OpSys (Operating System). You're smart, I think you have it figured out.

Basically, selecting *anything* on the query page narrows your search down. Leaving stuff unselected, or text boxes unfilled, broadens your search!

2. You see the box immediately below the top six boxes that contains an "Email" text box, with the words "matching as", a drop-down selection box, then some checkboxes with "Assigned To" checked by default? This allows you to filter your search down based upon email address. Let's put my email address in there, and see what happens.

Type "barnboy@trilobyte.net" in the top Email text box.

3. Let's narrow the search some more. Scroll down until you find the box with the word "Program" over the top of it. This is where we can narrow our search down to only specific products (software programs or product lines) in our Bugzilla database. Please notice the box is a *scrollbox*. Using the down arrow on the scrollbox, scroll down until you can see an entry called "Webtools". Select this entry.
4. Did you notice that some of the boxes to the right changed when you selected "Webtools"? Every Program (or Product) has different Versions, Components, and Target Milestones associated with it. A "Version" is the number of a software program.

### Example 4-1. Some Famous Software Versions

Do you remember the hype in 1995 when Microsoft Windows 95(r) was released? It may have been several years ago, but Microsoft(tm) spent over \$300 Million advertising this new Version of their software. Three years later, they released Microsoft Windows 98(r), another new version, to great fanfare, and then in 2000 quietly released Microsoft Windows ME(Millennium Edition)(r).

Software "Versions" help a manufacturer differentiate their current product from their previous products. Most do not identify their products by the year they were released. Instead, the "original" version of their software will often be numbered "1.0", with small bug-fix releases on subsequent tenths of a digit. In most cases, it's not a decimal number; for instance, often 1.9 is an *older* version of the software than 1.11, but is a *newer* version than 1.1.1.

In general, a "Version" in Bugzilla should refer to *released* products, not products that have not yet been released to the public. Forthcoming products are what the Target Milestone field is for.

A "Component" is a piece of a Product. It may be a standalone program, or some other logical division of a Product or Program. Normally, a Component has a single Owner, who is responsible for overseeing efforts to improve that Component.

### Example 4–2. Mozilla Webtools Components

Mozilla's "Webtools" Product is composed of several pieces (Components):

*Bonsai*, a tool to show recent changes to Mozilla

*Bugzilla*, a defect-tracking tool

*Build*, a tool to automatically compile source code into machine-readable form

*Despot*, a program that controls access to the other Webtools

*LXR*, a utility that automatically marks up text files to make them more readable

*MozBot*, a "robot" that announces changes to Mozilla in Chat

*TestManager*, a tool to help find bugs in Mozilla

*Tinderbox*, which displays reports from Build

A different person is responsible for each of these Components. Tara Hernandez keeps the "Bugzilla" component up-to-date.

A "Milestone", or "Target Milestone" is often a planned future "Version" of a product. In many cases, though, Milestones simply represent significant dates for a developer. Having certain features in your Product is frequently tied to revenue (money) the developer will receive if the features work by the time she reaches the Target Milestone. Target Milestones are a great tool to organize your time. If someone will pay you \$100,000 for incorporating certain features by a certain date, those features by that Milestone date become a very high priority. Milestones tend to be highly malleable creatures, though, that appear to be in reach but are out of reach by the time the important day arrives.

The Bugzilla Project has set up Milestones for future Bugzilla versions 2.14, 2.16, 2.18, 3.0, etc. However, a Target Milestone can just as easily be a specific date, code name, or weird alphanumeric combination, like "M19".

- OK, now let's select the "Bugzilla" component from its scrollbox.
- Skip down the page a bit — do you see the "submit query" button? Select it, and let's run this query!
- Congratulations! You've completed your first Query, and have before you the Bug List of the author of this Guide, Matthew P. Barnson (barnboy@trilobyte.net). If I'm doing well, you'll have a cryptic "Zarro Boogs Found" message on your screen. It is just a happy hacker's way of saying "Zero Bugs Found". However, I am fairly certain I will always have some bugs assigned to me that aren't done yet, so you won't often see that message!

I encourage you to click the bug numbers in the left-hand column and examine my bugs. Also notice that if you click the underlined links near the top of this page, they do not take you to context-sensitive help here, but instead sort the columns of bugs on the screen! When you need to sort your bugs by priority, severity, or the people they are assigned to, this is a tremendous timesaver.

A couple more interesting things about the Bug List page:

*Change Columns*: by selecting this link, you can show all kinds of information in the Bug List

*Change several bugs at once*: If you have sufficient rights to change all the bugs shown in the Bug List, you can mass-modify them. This is a big time-saver.

*Send mail to bug owners*: If you have many related bugs, you can request an update from every person who

owns the bugs in the Bug List asking them the status.

*Edit this query:* If you didn't get exactly the results you were looking for, you can return to the Query page through this link and make small revisions to the query you just made so you get more accurate results.

**Note:** There are many more options to the Bugzilla Query Page and the Bug List than I have shown you. But this should be enough for you to learn to get around. I encourage you to check out the [Bugzilla Home Page](#) to learn about the Anatomy and Life Cycle of a Bug before continuing.

---

### 4.3.3. Creating and Managing Bug Reports

*And all this time, I thought we were taking bugs out...*

---

#### 4.3.3.1. Writing a Great Bug Report

Before we plunge into writing your first bug report, I encourage you to read [Mozilla.org's Bug Writing Guidelines](#). While some of the advice is Mozilla-specific, the basic principles of reporting Reproducible, Specific bugs, isolating the Product you are using, the Version of the Product, the Component which failed, the Hardware Platform, and Operating System you were using at the time of the failure go a long way toward ensuring accurate, responsible fixes for the bug that bit you.

While you are at it, why not learn how to find previously reported bugs? Mozilla.org has published a great tutorial on finding duplicate bugs, available at <http://www.mozilla.org/quality/help/beginning-duplicate-finding.html>.

I realize this was a lot to read. However, understanding the mentality of writing great bug reports will help us on the next part!

1. Go back to <http://landfill.tequilarista.org/mozilla/bugzilla/> in your browser.
2. Select the [Enter a new bug report](#) link.
3. Select a product.
4. Now you should be at the "Enter Bug" form. The "reporter" should have been automatically filled out for you (or else Bugzilla prompted you to Log In again — you did keep the email with your username and password, didn't you?).
5. Select a Component in the scrollbox.
6. Bugzilla should have made reasonable guesses, based upon your browser, for the "Platform" and "OS" drop-down boxes. If those are wrong, change them — if you're on an SGI box running IRIX, we want to know!
7. Fill in the "Assigned To" box with the email address you provided earlier. This way you don't end up sending copies of your bug to lots of other people, since it's just a test bug.
8. Leave the "CC" text box blank. Fill in the "URL" box with "http://www.mozilla.org".
9. Enter "The Bugzilla Guide" in the Summary text box, and place any comments you have on this tutorial, or the Guide in general, into the Description box.

Voila! Select "Commit" and send in your bug report! Next we'll look at resolving bugs.

---

### 4.3.3.2. Managing your Bug Reports

OK, you should have a link to the bug you just created near the top of your page. It should say "Bug XXXX posted", with a link to the right saying "Back to BUG# XXXX". Select this link.

1. Scroll down a bit on the subsequent page, until you see the "Resolve bug, changing resolution to (dropdown box). Normally, you would "Accept bug (change status to ASSIGNED)", fix it, and then resolve. But in this case, we're going to short-circuit the process because this wasn't a real bug. Change the dropdown next to "Resolve Bug" to "INVALID", make sure the radio button is marked next to "Resolve Bug", then click "Commit".
2. Hey! It said it couldn't take the change in a big red box! That's right, you must specify a Comment in order to make this change. Select the "Back" button in your browser, add a Comment, then try Resolving the bug with INVALID status again. This time it should work.

You have now learned the basics of Bugzilla navigation, entering a bug, and bug maintenance. I encourage you to explore these features, and see what you can do with them! We'll spend no more time on individual Bugs or Queries from this point on, so you are on your own there.

But I'll give a few last hints!

There is a [CLUE](#) on the Query page that will teach you more how to use the form.

If you click the hyperlink on the [Component](#) box of the Query page, you will be presented a form that will describe what all the components are.

Possibly the most powerful feature of the Query page is the [Boolean Chart](#) section. It's a bit confusing to use the first time, but can provide unparalleled flexibility in your queries, allowing you to build extremely powerful requests.

Finally, you can build some nifty [Reports](#) using the "Bug Reports" link near the bottom of the query page, and also available via the "Reports" link at the footer of each page.

---

## 4.4. What's in it for me?

*Indiana, it feels like we walking on fortune cookies!*

*These ain't fortune cookies, kid...*

Customized User Preferences offer tremendous versatility to your individual Bugzilla experience. Let's plunge into what you can do! The first step is to click the "Edit prefs" link at the footer of each page once you have logged in to [Landfill](#).

---

### 4.4.1. Account Settings

On this page, you can change your basic Account Settings, including your password and full name. For security reasons, in order to change anything on this page you must type your *current* password into the "Old

Password" field. If you wish to change your password, type the new password you want into the "New Password" field and again into the "Re-enter new password" field to ensure you typed your new password correctly. Select the "Submit" button and you're done!

---

## 4.4.2. Email Settings

### 4.4.2.1. Email Notification

Ahh, here you can reduce or increase the amount of email sent you from Bugzilla! In the drop-down "Notify me of changes to", select one of

*All qualifying bugs:* sends you every change to every bug where your name is somewhere on it, regardless of who changed it.

*Only those bugs which I am listed in the CC line:* prevents you from receiving mail for which you are the reporter, owner, or QA contact. If you are on the CC list, presumably someone had a *good* reason for you to get the email.

*All qualifying bugs except those which I change:* This is the default, and a sensible setting. If someone else changes your bugs, you will get emailed, but if you change bugs yourself you will receive no notification of the change.

---

### 4.4.2.2. New Email Technology

**Note:** This option may not be available in all Bugzilla installations, depending upon the preferences of the systems administrator responsible for the setup of your Bugzilla. However, if you really want this functionality, ask her to "enable newemailtech in Params" and "make it the default for all new users", referring her to the Administration section of this Guide.

Disregard the warnings about "experimental and bleeding edge"; the code to handle email in a cleaner manner than that historically used for Bugzilla is quite robust and well-tested now.

I recommend you enable the option, "Click here to sign up (and risk any bugs)". Your email-box will thank you for it. The fundamental shift in "newemailtech" is away from standard UNIX "diff" output, which is quite ugly, to a prettier, better laid-out email.

---

### 4.4.2.3. "Watching" Users

**Note:** This option may not be available in all Bugzilla installations, depending upon the preferences of the systems administrator responsible for the setup of your Bugzilla. However, if you really want this functionality, ask her to "enable watchers in Params".

By entering user email names into the "Users to watch" text entry box, delineated by commas, you can watch bugs of other users. This powerful functionality enables seamless transitions as developers change projects, managers wish to get in touch with the issues faced by their direct reports, or users go on vacation. If any of these three situations apply to you, you will undoubtedly find this feature quite convenient.

---

### 4.4.3. Page Footer

**Note:** By default, this page is quite barren. However, go explore the Query Page some more; you will find that you can store numerous queries on the server, so if you regularly run a particular query it is just a drop-down menu away. On this page of Preferences, if you have many stored queries you can elect to have them always one-click away!

If you have many stored queries on the server, here you will find individual drop-downs for each stored query. Each drop-down gives you the option of that query appearing on the footer of every page in Bugzilla! This gives you powerful one-click access to any complex searches you may set up, and is an excellent way to impress your boss...

**Tip:** By default, the "My Bugs" link appears at the bottom of each page. However, this query gives you both the bugs you have reported, as well as those you are assigned. One of the most common uses for this page is to remove the "My Bugs" link, replacing it with two other queries, commonly called "My Bug Reports" and "My Bugs" (but only referencing bugs assigned to you). This allows you to distinguish those bugs you have reported from those you are assigned. I commonly set up complex Boolean queries in the Query page and link them to my footer in this page. When they are significantly complex, a one-click reference can save hours of work.

---

### 4.4.4. Permissions

This is a purely informative page which outlines your current permissions on this installation of Bugzilla. If you have permissions to grant certain permissions to other users, the "other users" link appears on this page as well as the footer. For more information regarding user administration, please consult the Administration section of this Guide.

---

## 4.5. Using Bugzilla–Conclusion

Thank you for reading through this portion of the Bugzilla Guide. I anticipate it may not yet meet the needs of all readers. If you have additional comments or corrections to make, please submit your contributions to the [mozilla-webtools](mailto:mozilla-webtools@mozilla.org) mailing list/newsgroup. The mailing list is mirrored to the netscape.public.mozilla.webtools newsgroup, and the newsgroup is mirrored to mozilla-webtools@mozilla.org

---



# Chapter 5. Integrating Bugzilla with Third-Party Tools

## 5.1. Bonsai

We need Bonsai integration information.

---

## 5.2. CVS

We need CVS integration information

---

## 5.3. Perforce SCM

Richard Brooksby and his team have an integration tool in public beta. You can find it at <http://www.ravenbrook.com/project/p4dti>.

---

## 5.4. Tinderbox

We need Tinderbox integration information

---

# Chapter 6. The Future of Bugzilla

This section largely contributed by Matthew Tuck

---

## 6.1. Reducing Spam

Those who use Bugzilla frequently are probably used to notification spam – unwanted or unnecessary notifications. A number of proposals have been put forward to attempt to reduce this.

### 1. Reduce CC Spam

Some of you probably know me as that guy who CCs on heaps and heaps of bugs. Just as you get a lot of CC changes from me, so do I get a lot from others. Why should CC changes send out email notifications?

It's not necessarily the best idea to just remove the CC spam, there are other issues too, like the difficulty of adding to large CC fields.

For these reasons and more, an RFE for a per user "BCC" facility exists that people could use to silently and privately track bugs, in a similar way to voting today, but applying to an unlimited number of bugs. See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=7345](http://bugzilla.mozilla.org/show_bug.cgi?id=7345)".

### 2. Bulk Changes

You know the drill – a large milestone change, a component movement, whatever, and lots of notifications are generated. If there's enough maybe you'll just go delete, delete, delete, whoops, there goes another notification that wasn't from the bulk change you missed.

Shouldn't bulk changes send out one notification? A proposal for this is at "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=26943](http://bugzilla.mozilla.org/show_bug.cgi?id=26943)".

### 3. Configurable Notification Criteria

It would be good if you could choose what you want to receive. There are two parts to this.

(a) Choose a selection of bugs you're interested in. This would be similar to CC except you let the set be computed from selection criteria rather than limited to the bugs your name is on. There is currently a limited version of this in the bugzilla preferences, ie "all qualifying bugs"/"all qualifying bugs except the ones I change"/"only those bugs which I am listed on the cc line".

(b) Choose what changes will trigger a notification for the bugs you are watching. With this, you could choose whether you want to receive cc, dependency and keyword changes, for example.

Both of these proposals live at  
"[http://bugzilla.mozilla.org/show\\_bug.cgi?id=14137](http://bugzilla.mozilla.org/show_bug.cgi?id=14137)".

---

## 6.2. Better Searching

Current searching tools in Bugzilla include the querying mechanism, special summary reports and dependency trees. This message is about new facilities.

### 1. General Summary Reports

For some time now it has been apparent to me that the query bug list leaves a little to be desired in its linear nature. There is a need to have categorised subsets, and counts of each category. If you don't believe me, how about these facilities already in place or which people have asked for:

Most Doomed Reports – Categorised On Assignee, Shows and Counts Number of Bugs For Each Assignee

Bug #15806 (Most Voted For Bugs) – Categorised On Product, Shows Bugs Voters Most Want Fixed

Bug #9789 (BugAthon Tracking Page) – Categorised On Developer (Subset), Counts Number of Bugs

Bug #9409 and #9411 – The desire to be able to report on more subsets.

Hopefully you can see the gist of what is desired here. It's a general reporting mechanism.

This mechanism lets you choose the subset of bugs to operate on (like query), let's you categorise them, possibly along with subcategories and counts the number of bugs within each category. It might or might not show the actual bugs themselves, and it might limit the number of bugs within a category, or categories to report on.

I'm further sure that many applications of this mechanism would only be recognised once it was implemented.

The general summary reports bug is at  
"[http://bugzilla.mozilla.org/show\\_bug.cgi?id=12282](http://bugzilla.mozilla.org/show_bug.cgi?id=12282)".

### 2. Related Bugs

It would be nice to have a field where you could enter other bugs related to the current bug – it would be handy for navigation and possibly even finding duplicates. See  
"[http://bugzilla.mozilla.org/show\\_bug.cgi?id=12286](http://bugzilla.mozilla.org/show_bug.cgi?id=12286)".

### 3. Column Specification Support

## 6.2. Better Searching

Currently query seems to get what columns to report on from whatever the user last used. This doesn't work well for "prepackaged queries", where you followed a link. You can probably add a column by specifying a sort column, but this is difficult and suboptimal.

Furthermore, I find that when I want to add a column to a query, it's usually a one off and I would prefer it to go away for the next query. Hence, it would be nice to specify the columns that appear on the query (and general summary report) pages. The default query mechanism should be able to let you specify your default columns.

This proposal lives at  
["http://bugzilla.mozilla.org/show\\_bug.cgi?id=12284"](http://bugzilla.mozilla.org/show_bug.cgi?id=12284).

---

### 6.3. Description Flags and Tracking Bugs

Since I last posted on this issue, we now have "keywords" that solve many of the issues of description and status whiteboard keywords. We have seen a migration towards keywords, but there is still further to go.

#### Description ( + Status Whiteboard ) Keywords

---

Some description keywords remain. I'd like to hear what reasons, other than time, there are for these staying as they are. I'm suspecting many are not really being used. Hopefully we can totally remove these eventually.

#### Tracking Bugs

---

When I suggested keywords, I did so to get rid of tracking bugs too, though we've had less success on that front.

There are many disadvantages to tracking bugs.

- They can pollute bugs counts, and you must make sure you exclude them. I believe the meta keyword might be used for this purpose.
- They have an assignee but there is nothing to fix, and that person can get whined at by Bugzilla.
- It would be better to craft your own "dependency tree" rather than rely on a fixed hierarchy in the bug system.
- In creating a nice little hierarchy, many bugs duplicate information that should be available in other ways, eg ["http://bugzilla.mozilla.org/show\\_bug.cgi?id=12833"](http://bugzilla.mozilla.org/show_bug.cgi?id=12833) which is about beta 1 networking issues. These could fall behind the actual data. What tracking bugs are good for, ad hoc lists, is what keywords

are better for.

– An automatically generated dependency structure between one "tracking bug" and another would be better than a manual one, since it gives exact rather than manually set up classifications.

Probably the only feature preventing tracking bugs being replaced is the dependency tree. The quintessential tracking bug seems to be bug #7229 "chofmann's watch list", which probably has about a couple of hundred bugs at various levels, which allows a nice visualisation.

Before keywords can replace tracking bugs better visualisation is going to be required. General summary reports and dependency forests of a bug list ("[http://bugzilla.mozilla.org/show\\_bug.cgi?id=12992](http://bugzilla.mozilla.org/show_bug.cgi?id=12992)") could both help, but neither solves the problem totally. Perhaps keywords within keywords would help here. In any case, I'm still thinking about this one.

Some tracking bugs could definitely be turned into keywords immediately though, and I'll point the finger at "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=7954](http://bugzilla.mozilla.org/show_bug.cgi?id=7954)" here since that's what came to mind first.

---

## 6.4. Bug Issues

### 1. Inline Bug Changes

Why do I see so many "moving to M5" and "reassigning to blahblah" messages, and in other circumstances none are entered? Why aren't these automatically generated? A comment should be only necessary when there is something to add, and if I'm not interested in this sort of information, I should be able to hide it.

At the moment we're in a hybrid world where we don't get everything, but we can't get rid of the bug change "messages" either. Furthermore, "View Bug Activity" requires me to manually cross reference events on another page, rather than being able to visually see the chronological order. Shouldn't I be able to see all the information on one page?

A proposal to allow bugs to be shown either way is at "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=11368](http://bugzilla.mozilla.org/show_bug.cgi?id=11368)".

### 2. Hard Wrapping Comments

One thing that annoys me is the fact that comments are "hard wrapped" to a certain column width. This is a mistake Internet Mail and News has made, unlike every word processor in existence, and as a consequence, Usenet suffers to this day from bad software. Why has Bugzilla repeated the problem?

Hard wrapping to a certain column width is open to abuse (see old Mozilla browsers that didn't wrap properly, resulting in many ugly bug reports we have to read to this day), and furthermore doesn't expand to fill greater screen sizes. I'm also under the impression the current hard wrap uses a non-standard HTML facility. See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=11901](http://bugzilla.mozilla.org/show_bug.cgi?id=11901)".

### 3. REMIND and LATER Are Evil

I really hate REMIND and LATER. Not because they mean something won't be implemented, but because they aren't the best solutions.

Why are they bad? Well, basically because they are not resolved, yet they are marked as such. Hence queries have to be well crafted to include them.

LATER, according to Bugzilla, means it won't be done this release. There is a better mechanism of doing this, that is assigning to `nobody@mozilla.org` and making the milestone blank. It's more likely to appear in a casual query, and it doesn't resolve the bug.

REMIND, according to Bugzilla, means it might still be implemented this release. Well, why not just move it to a later milestone then? You're a lot less likely to forget it. If it's really needed, a keyword would be better.

Some people can't use blank milestones to mean an untargetted milestone, since they use this to assess new bugs that have no target. Hence, it would be nice to distinguish between bugs that have not yet been considered, and those that really are not assigned to any milestone in the future (assumedly beyond).

All this is covered at "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=13534](http://bugzilla.mozilla.org/show_bug.cgi?id=13534)".

### 4. Create An Enhancement Field

Currently enhancement is an option in severity. This means that important enhancements (like for example, POP3 support) are not properly distinguished as such, because they need a proper severity. This dilutes the meaning of enhancement.

If enhancement was separated, we could properly see what was an enhancement. See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=9412](http://bugzilla.mozilla.org/show_bug.cgi?id=9412)". I see keywords like [RFE] and [FEATURE] that seem to be compensating for this problem.

---

## 6.5. Database Integrity

Bugzilla could be more proactive in detecting suboptimal situations and prevent them or whine about them.

### 1. Bugzilla Crime #1: Marking A Bug Fixed With Unresolved Dependencies

It can't be marked fixed with unresolved dependencies. Either mark it INVALID (tracking bugs), fix the dependencies at the same time, or resolve the blockers.

See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=24496](http://bugzilla.mozilla.org/show_bug.cgi?id=24496)".

### 2. Keyword Restrictions

Some keywords should only apply in certain circumstances, eg beta1 => Milestone <

M14, css1 => Component = Style System are possibilities. See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=26940](http://bugzilla.mozilla.org/show_bug.cgi?id=26940)".

### 3. Whine About Old Votes

Old votes can just sit on resolved bugs. This is problematic with duplicates especially. Automatic transferral/removal is not appropriate since bugs can be reopened, but a whining solution might work. See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=27553](http://bugzilla.mozilla.org/show_bug.cgi?id=27553)".

### 4. Whine And Warn About Milestone Mismatches

Here's a fun one. Bug X (M17) depends on Bug Y (M15). Bug Y gets moved out to M19. The notification to the assignee of Bug X gets ignored (of course) and Bug X is now due to be fixed before one of its blockers.

Warnings about this when it is detected as well as whining about it in email would help bring these issues to the attention of people sooner.

Note that this would be less of a problem if we didn't have so many tracking bugs since they aren't updated that often and often have this problem.

See "[http://bugzilla.mozilla.org/show\\_bug.cgi?id=16743](http://bugzilla.mozilla.org/show_bug.cgi?id=16743)".

---

## 6.6. Bugzilla 3.0

One day, Bugzilla 3.0 will have lots of cool stuff.

---

# Appendix A. The Bugzilla FAQ

## 1. General Questions

- A.1.1. [Where can I find information about Bugzilla?](#)
- A.1.2. [What license is Bugzilla distributed under?](#)
- A.1.3. [How do I get commercial support for Bugzilla?](#)
- A.1.4. [What major companies or projects are currently using Bugzilla for bug-tracking?](#)
- A.1.5. [Who maintains Bugzilla?](#)
- A.1.6. [How does Bugzilla stack up against other bug-tracking databases?](#)
- A.1.7. [How do I change my user name in Bugzilla?](#)
- A.1.8. [Why doesn't Bugzilla offer this or that feature or compatibility with this other tracking software?](#)
- A.1.9. [Why MySQL? I'm interested in seeing Bugzilla run on Oracle/Sybase/Msql/PostgreSQL/MSSQL?](#)
- A.1.10. [Why do the scripts say "/usr/bonsaitools/bin/perl" instead of "/usr/bin/perl" or something else?](#)

## 2. Red Hat Bugzilla

- A.2.1. [What about Red Hat Bugzilla?](#)
- A.2.2. [What are the primary benefits of Red Hat Bugzilla?](#)
- A.2.3. [What's the current status of Red Hat Bugzilla?](#)

## 3. Loki Bugzilla (AKA Fenris)

- A.3.1. [What about Loki Bugzilla?](#)
- A.3.2. [Who maintains Fenris \(Loki Bugzilla\) now?](#)
- A.3.3.

## 4. Pointy-Haired-Boss Questions

- A.4.1. [Is Bugzilla web-based or do you have to have specific software or specific operating system on your machine?](#)
- A.4.2. [Has anyone you know of already done any Bugzilla integration with Perforce \(SCM software\)?](#)
- A.4.3. [Does Bugzilla allow the user to track multiple projects?](#)
- A.4.4. [If I am on many projects, and search for all bugs assigned to me, will Bugzilla list them for me and allow me to sort by project, severity etc?](#)
- A.4.5. [Does Bugzilla allow attachments \(text, screenshots, urls etc\)? If yes, are there any that are NOT allowed?](#)
- A.4.6. [Does Bugzilla allow us to define our own priorities and levels? Do we have complete freedom to change the labels of fields and format of them, and the choice of acceptable values?](#)
- A.4.7. [Does Bugzilla provide any reporting features, metrics, graphs, etc? You know, the type of stuff that management likes to see. :\)](#)
- A.4.8. [Is there email notification and if so, what do you see when you get an email? Do you see bug number and title or is it only the number?](#)
- A.4.9. [Can email notification be set up to send to multiple people, some on the To List, CC List, BCC List etc?](#)
- A.4.10. [If there is email notification, do users have to have any particular type of email application?](#)
- A.4.11. [If I just wanted to track certain bugs, as they go through life, can I set it up to alert me via email whenever that bug changes, whether it be owner, status or description etc.?](#)
- A.4.12. [Does Bugzilla allow data to be imported and exported? If I had outsiders write up a bug report using a MS Word bug template, could that template be imported into "matching" fields? If I wanted to take the results of a query and export that data to MS Excel, could I do that?](#)
- A.4.13. [Does Bugzilla allow fields to be added, changed or deleted? If I want to customize the bug submission form to meet our needs, can I do that using our terminology?](#)



- A.4.14. [Has anyone converted Bugzilla to another language to be used in other countries? Is it localizable?](#)
- A.4.15. [Can a user create and save reports? Can they do this in Word format? Excel format?](#)
- A.4.16. [Can a user re-run a report with a new project, same query?](#)
- A.4.17. [Can a user modify an existing report and then save it into another name?](#)
- A.4.18. [Does Bugzilla have the ability to search by word, phrase, compound search?](#)
- A.4.19. [Can the admin person establish separate group and individual user privileges?](#)
- A.4.20. [Does Bugzilla provide record locking when there is simultaneous access to the same bug? Does the second person get a notice that the bug is in use or how are they notified?](#)
- A.4.21. [Are there any backup features provided?](#)
- A.4.22. [Can users be on the system while a backup is in progress?](#)
- A.4.23. [What type of human resources are needed to be on staff to install and maintain Bugzilla? Specifically, what type of skills does the person need to have? I need to find out if we were to go with Bugzilla, what types of individuals would we need to hire and how much would that cost vs buying an "Out-of-the-Box" solution.](#)
- A.4.24. [What time frame are we looking at if we decide to hire people to install and maintain the Bugzilla? Is this something that takes hours or weeks to install and a couple of hours per week to maintain and customize or is this a multi-week install process, plus a full time job for 1 person, 2 people, etc?](#)
- A.4.25. [Is there any licensing fee or other fees for using Bugzilla? Any out-of-pocket cost other than the bodies needed as identified above?](#)
- 5. [Bugzilla Installation](#)
  - A.5.1. [How do I download and install Bugzilla?](#)
  - A.5.2. [How do I install Bugzilla on Windows NT?](#)
  - A.5.3. [Is there an easy way to change the Bugzilla cookie name?](#)
- 6. [Bugzilla Security](#)
  - A.6.1. [How do I completely disable MySQL security if it's giving me problems \(I've followed the instructions in the README!\)?](#)
  - A.6.2. [Are there any security problems with Bugzilla?](#)
  - A.6.3. [I've implemented the security fixes mentioned in Chris Yeh's security advisory of 5/10/2000 advising not to run MySQL as root, and am running into problems with MySQL no longer working correctly.](#)
- 7. [Bugzilla Email](#)
  - A.7.1. [I have a user who doesn't want to receive any more email from Bugzilla. How do I stop it entirely for this user?](#)
  - A.7.2. [I'm evaluating/testing Bugzilla, and don't want it to send email to anyone but me. How do I do it?](#)
  - A.7.3. [I want whineatnews.pl to whine at something more, or other than, only new bugs. How do I do it?](#)
  - A.7.4. [I don't like/want to use Procmail to hand mail off to bug\\_email.pl. What alternatives do I have?](#)
  - A.7.5. [How do I set up the email interface to submit/change bugs via email?](#)
  - A.7.6. [Email takes FOREVER to reach me from bugzilla -- it's extremely slow. What gives?](#)
  - A.7.7. [How come email never reaches me from bugzilla changes?](#)
- 8. [Bugzilla Database](#)
  - A.8.1. [I've heard Bugzilla can be used with Oracle?](#)
  - A.8.2. [Bugs are missing from queries, but exist in the database \(and I can pull them up by specifying the bug ID\). What's wrong?](#)
  - A.8.3. [I think my database might be corrupted, or contain invalid entries. What do I do?](#)
  - A.8.4. [I want to manually edit some entries in my database. How?](#)
  - A.8.5. [I try to add myself as a user, but Bugzilla always tells me my password is wrong.](#)

- A.8.6. [I think I've set up MySQL permissions correctly, but bugzilla still can't connect.](#)
- A.8.7. [How do I synchronize bug information among multiple different Bugzilla databases?](#)
- A.8.8. [Why do I get bizarre errors when trying to submit data, particularly problems with "groupset"?](#)
- A.8.9. [How come even after I delete bugs, the long descriptions show up?](#)
- 9. [Bugzilla and Win32](#)
  - A.9.1. [What is the easiest way to run Bugzilla on Win32 \(Win98+/NT/2K\)?](#)
  - A.9.2. [Is there a "Bundle::Bugzilla" equivalent for Win32?](#)
  - A.9.3. [CGI's are failing with a "something.cgi is not a valid Windows NT application" error. Why?](#)
  - A.9.4. [Can I have some general instructions on how to make Bugzilla on Win32 work?](#)
  - A.9.5. [I'm having trouble with the perl modules for NT not being able to talk to the database.](#)
- 10. [Bugzilla Usage](#)
  - A.10.1. [The query page is very confusing. Isn't there a simpler way to query?](#)
  - A.10.2. [I'm confused by the behavior of the "accept" button in the Show Bug form. Why doesn't it assign the bug to me when I accept it?](#)
  - A.10.3. [I can't upload anything into the database via the "Create Attachment" link. What am I doing wrong?](#)
  - A.10.4. [Email submissions to Bugzilla that have attachments end up asking me to save it as a "cgi" file.](#)
  - A.10.5. [How do I change a keyword in Bugzilla, once some bugs are using it?](#)
- 11. [Bugzilla Hacking](#)
  - A.11.1. [What bugs are in Bugzilla right now?](#)
  - A.11.2. [What's the best way to submit patches? What guidelines should I follow?](#)

## 1. General Questions

**A.1.1.** Where can I find information about Bugzilla?

You can stay up-to-date with the latest Bugzilla information at <http://www.mozilla.org/projects/bugzilla/>

**A.1.2.** What license is Bugzilla distributed under?

Bugzilla is covered by the Mozilla Public License. See details at <http://www.mozilla.org/MPL/>

**A.1.3.** How do I get commercial support for Bugzilla?

[www.collab.net](http://www.collab.net) offers Bugzilla as part of their standard offering to large projects. They do have some minimum fees that are pretty hefty, and generally aren't interested in small projects.

There are several experienced Bugzilla hackers on the mailing list/newsgroup who are willing to whore themselves out for generous compensation. Try sending a message to the mailing list asking for a volunteer.

**A.1.4.** What major companies or projects are currently using Bugzilla for bug-tracking?

There are *dozens* of major companies with public Bugzilla sites to track bugs in their products. A few include:

Netscape/AOL

Mozilla.org

AtHome Corporation

Red Hat Software  
Loki Entertainment Software  
SuSe Corp  
The Horde Project  
The Eazel Project  
AbiSource  
Real Time Enterprises, Inc  
Eggheads.org  
Strata Software  
RockLinux  
Creative Labs (makers of SoundBlaster)  
The Apache Foundation  
The Gnome Foundation  
Linux-Mandrake

Suffice to say, there are more than enough huge projects using Bugzilla that we can safely say it's extremely popular.

### **A.1.5. Who maintains Bugzilla?**

There are many, many contributors from around the world maintaining Bugzilla. The designated "Maintainer" is Tara Hernandez, with QA support by Matthew Tuck. Dan Mosedale and Dawn Endico are employees of Mozilla.org responsible for the installation of Bugzilla there, and are very frequent code contributors. Terry Weissman originally ported Bugzilla, but "these days, Terry just hangs around and heckles." The rest of us are mostly transient developers; Bugzilla suits our needs, and we contribute code as we have needs for updates.

### **A.1.6. How does Bugzilla stack up against other bug-tracking databases?**

A year has gone by, and I *still* can't find any head-to-head comparisons of Bugzilla against other defect-tracking software. However, from my personal experience with other bug-trackers, Bugzilla offers superior performance on commodity hardware, better price (free!), more developer-friendly features (such as stored queries, email integration, and platform independence), improved scalability, open source code, greater flexibility, and superior ease-of-use.

If you happen to be a commercial Bugzilla vendor, please step forward with a rebuttal so I can include it in the FAQ. We're not in pursuit of Bugzilla ueber alles; we simply love having a powerful, open-source tool to get our jobs done.

### **A.1.7. How do I change my user name in Bugzilla?**

You can't. However, the administrative account can, by simply opening your user account in `editusers.cgi` and changing the login name.

### **A.1.8. Why doesn't Bugzilla offer this or that feature or compatability with this other tracking software?**

It may be that the support has not been built yet, or that you have not yet found it. Bugzilla is making tremendous strides in usability, customizability, scalability, and user interface. It is widely considered the

most complete and popular open–source bug–tracking software in existence.

That doesn't mean it can't use improvement! You can help the project along by either hacking a patch yourself that supports the functionality you require, or else submitting a "Request for Enhancement" (RFE) using the bug submission interface at [bugzilla.mozilla.org](http://bugzilla.mozilla.org).

**A.1.9.** Why MySQL? I'm interested in seeing Bugzilla run on Oracle/Sybase/Msql/PostgreSQL/MSSQL?

Terry Weissman answers,

You're not the only one. But *I* am not very interested. I'm not a real SQL or database person. I just wanted to make a useful tool, and build it on top of free software. So, I picked MySQL, and learned SQL by staring at the MySQL manual and some code lying around here, and wrote Bugzilla. I didn't know that Enum's were non–standard SQL. I'm not sure if I would have cared, but I didn't even know. So, to me, things are "portable" because it uses MySQL, and MySQL is portable enough. I fully understand (now) that people want to be portable to other databases, but that's never been a real concern of mine.

Things aren't quite that grim these days, however. Terry pretty much sums up much of the thinking many of us have for Bugzilla, but there is light on the horizon for database–independence! Here are some options:

[Red Hat Bugzilla](#): Runs a modified Bugzilla 2.8 atop an Oracle database.

[Interzilla](#): A project to run Bugzilla on Interbase. No code released yet, however.

*Bugzilla 3.0*: One of the primary stated goals is multiple database support.

**A.1.10.** Why do the scripts say `"/usr/bonsaitools/bin/perl"` instead of `"/usr/bin/perl"` or something else?

Mozilla.org uses `/usr/bonsaitools/bin/perl`. The prime rule in making submissions is "don't break `bugzilla.mozilla.org`". If it breaks it, your patch will be reverted faster than you can do a diff.

Here's Terry Weissman's comment, for some historical context:

[This was] purely my own convention. I wanted a place to put a version of Perl and other tools that was strictly under my control for the various webtools, and not subject to anyone else. Edit it to point to whatever you like.

**Note:** We always recommend that, if possible, you keep the path as `/usr/bonsaitools/bin/perl`, and simply add a `/usr/bonsaitools` and `/usr/bonsaitools/bin` directory, then symlink your version of perl to `/usr/bonsaitools/bin/perl`. This will make upgrading your Bugzilla much easier in the future.

Obviously, if you do not have root access to your Bugzilla box, our suggestion is irrelevant.

## 2. Red Hat Bugzilla

**A.2.1.** What about Red Hat Bugzilla?

Red Hat Bugzilla is arguably more user–friendly, customizable, and scalable than stock Bugzilla. Check it out at <http://bugzilla.redhat.com> and the sources at <ftp://people.redhat.com/dkl/>. They've set their Bugzilla up to work with Oracle out of the box. Note that Redhat Bugzilla is based upon the 2.8 Bugzilla tree; Bugzilla

has made some tremendous advances since the 2.8 release. Why not download both Bugzillas to check out the differences for yourself?

Dave Lawrence, the original Red Hat Bugzilla maintainer, mentions:

Somebody needs to take the ball and run with it. I'm the only maintainer and am very pressed for time.

If you, or someone you know, has the time and expertise to do the integration work so main-tree Bugzilla 2.12 and higher integrates the Red Hat Bugzilla Oracle modifications, please donate your time to supporting the Bugzilla project.

### A.2.2. What are the primary benefits of Red Hat Bugzilla?

*Dave Lawrence:*

For the record, we are not using any template type implementation for the cosmetic changes maded to Bugzilla. It is just alot of html changes in the code itself. I admit I may have gotten a little carried away with it but the corporate types asked for a more standardized interface to match up with other projects relating to Red Hat web sites. A lot of other web based internal tools I am working on also look like Bugzilla.

I do want to land the changes that I have made to Bugzilla but I may have to back out a good deal and make a different version of Red Hat's Bugzilla for checking in to CVS. Especially the cosmetic changes because it seems they may not fit the general public. I will do that as soon as I can. I also still do my regular QA responsibilities along with Bugzilla so time is difficult sometimes to come by.

There are also a good deal of other changes that were requested by management for things like support contracts and different permission groups for making bugs private. Here is a short list of the major changes that have been made:

1. No enum types. All old enum types are now separate smaller tables.
2. No bit wise operations. Not all databases support this so they were changed to a more generic way of doing this task
3. Bug reports can only be altered by the reporter, assignee, or a privileged bugzilla user. The rest of the world can see the bug but in a non-changeable format (unless the bug has been marked private). They can however add comments, add and remove themselves from the CC list
4. Different group scheme. Each group has an id number related to it. There is a user\_group table which contains userid to groupid mappings to determine which groups each user belongs to. Additionally there is a bug\_group table that has bugid to groupid mappings to show which groups can see a particular bug. If there are no entries for a bug in this table then the bug is public.
5. Product groups. product\_table created to only allow certain products to be visible for certain groups in both bug entry and query. This was particulary helpful for support contracts.
6. Of course many (too many) changes to Bugzilla code itself to allow use with Oracle and still allow operation with Mysql if so desired. Currently if you use Mysql it is set to use Mysql's old permission scheme to keep breakage to a minimum. Hopefully one day this will standardize on one style which may of course be something

completely different.

7. Uses Text::Template perl module for rendering of the dynamic HTML pages such as enter\_bug.cgi, query.cgi, bug\_form.pl, and for the header and footer parts of the page. This allows the html to be separate from the perl code for customizing the look and feel of the page to one's preference.
8. There are many other smaller changes. There is also a port to Oracle that I have been working on as time permits but is not completely finished but somewhat usable. I will merge it into our standard code base when it becomes production quality. Unfortunately there will have to be some conditionals in the code to make it work with other than Oracle due to some differences between Oracle and Mysql.

Both the Mysql and Oracle versions of our current code base are available from <ftp://people.redhat.com/dkl>. If Terry/Tara wants I can submit patch files for all of the changes I have made and he can determine what is suitable for addition to the main bugzilla code base. But for me to commit changes to the actual CVS I will need to back out alot of things that are not suitable for the rest of the Bugzilla community. I am open to suggestions.

### A.2.3. What's the current status of Red Hat Bugzilla?

**Note:** This information is somewhat dated; I last updated it 7 June 2000.

*Dave Lawrence:*

I suppose the current thread warrants an update on the status of Oracle and bugzilla ;) We have now been running Bugzilla 2.8 on Oracle for the last two days in our production environment. I tried to do as much testing as possible with it before going live which is some of the reason for the long delay. I did not get enough feedback as I would have liked from internal developers to help weed out any bugs still left so I said "Fine, i will take it live and then I will get the feedback I want :)" So it is now starting to stabilize and it running quite well after working feverishly the last two days fixing problems as soon as they came in from the outside world. The current branch in cvs is up2date if anyone would like to grab it and try it out. The oracle\_setup.pl is broken right now due to some last minute changes but I will update that soon. Therefore you would probably need to create the database tables the old fashioned way using the supplied sql creation scripts located in the ./oracle directory. We have heavy optimizations in the database it self thanks to the in-house DBA here at Red Hat so it is running quite fast. The database itself is located on a dual PII450 with 1GB ram and 14 high voltage differential raided scsi drives. The tables and indexes are partitioned in 4 chunks across the raided drive which is nice because when ever you need to do a full table scan, it is actually starting in 4 different locations on 4 different drives simultaneously. And the indexes of course are on separate drives from the data so that speeds things up tremendously. When I can find the time I will document all that we have done to get this thing going to help others that may need it.

As Matt has mentioned it is still using out-dated code and with a little help I would like to bring everything up to date for eventual incorporation with the main cvs tree. Due to other duties I have with the company any help with this would be appreciated. What we are using now is what I call a best first effort. It definitely can be improved on and may even need complete rewrites in a lot of areas. A lot of changes may have to be made in the way Bugzilla does things currently to make this transition to a more generic database interface. Fortunately when making the Oracle changes I made sure I didn't do anything that I would consider Oracle specific and could not be easily done with other databases. Alot of the sql statements

need to be broken up into smaller utilities that themselves would need to make decisions on what database they are using but the majority of the code can be made database neutral.

### 3. Loki Bugzilla (AKA Fenris)

**Note:** Loki's "Fenris" Bugzilla is no longer actively maintained. It works well enough for Loki. Additionally, the major differences in Fenris have now been integrated into the main source tree of Bugzilla, so there's not much reason to go grab the source. I left this section of the FAQ principally for historical interest.

#### A.3.1. What about Loki Bugzilla?

Loki Games has a customized version of Bugzilla available at <http://fenris.lokigames.com>. From that page,

You may have noticed that Fenris is a fork from Bugzilla—our patches weren't suitable for integration—and a few people have expressed interest in the code. Fenris has one major improvement over Bugzilla, and that is individual comments are not appended onto a string blob, they are stored as a record in a separate table. This allows you to, for instance, separate comments out according to privilege levels in case your bug database could contain sensitive information not for public eyes. We also provide things like email hiding to protect user's privacy, additional fields such as 'user\_affected' in case someone enters someone else's bug, comment editing and deletion, and more conditional system variables than Bugzilla does (turn off attachments, qacontact, etc.).

#### A.3.2. Who maintains Fenris (Loki Bugzilla) now?

Raphael Barrerro <[raistlin@lokigames.com](mailto:raistlin@lokigames.com)>. Michael Vance created the initial fork, but no longer maintains the project.

#### A.3.3.

### 4. Pointy-Haired-Boss Questions

**Note:** The title of this section doesn't mean you're a PHB — it just means you probably HAVE a PHB who wants to know this :)

#### A.4.1. Is Bugzilla web-based or do you have to have specific software or specific operating system on your machine?

It is web and e-mail based. You can edit bugs by sending specially formatted email to a properly configured Bugzilla, or control via the web.

#### A.4.2. Has anyone you know of already done any Bugzilla integration with Perforce (SCM software)?

Yes! You can find more information elsewhere in "The Bugzilla Guide" in the "Integration with Third-Party Products" section. The section on Perforce isn't very large, but as the maintainer of the Guide is charged with Perforce/Bugzilla integration by his company, you can expect this section to grow.

#### A.4.3. Does Bugzilla allow the user to track multiple projects?

## The Bugzilla Guide

Absolutely! You can track up to a "soft-limit" of around 64 individual "Products", that can each be composed of as many "Components" as you want. Check the Administration section of the Bugzilla Guide for more information regarding setting up Products and Components.

**A.4.4.** If I am on many projects, and search for all bugs assigned to me, will Bugzilla list them for me and allow me to sort by project, severity etc?

Yes.

**A.4.5.** Does Bugzilla allow attachments (text, screenshots, urls etc)? If yes, are there any that are NOT allowed?

Yes. There are many specific MIME-types that are pre-defined by Bugzilla, but you may specify any arbitrary MIME-type you need when you upload the file. Since all attachments are stored in the database, however, I recommend storing large binary attachments elsewhere in the web server's file system and providing a hyperlink as a comment, or in the provided "URL" field in the bug report.

**A.4.6.** Does Bugzilla allow us to define our own priorities and levels? Do we have complete freedom to change the labels of fields and format of them, and the choice of acceptable values?

Yes. However, modifying some fields, notably those related to bug progression states, also require adjusting the program logic to compensate for the change.

**A.4.7.** Does Bugzilla provide any reporting features, metrics, graphs, etc? You know, the type of stuff that management likes to see. :)

Yes. Look at <http://bugzilla.mozilla.org/reports.cgi> for basic reporting facilities.

For more advanced reporting, I recommend hooking up a professional reporting package, such as Crystal Reports, and use ODBC to access the MySQL database. You can do a lot through the Query page of Bugzilla as well, but right now Advanced Reporting is much better accomplished through third-party utilities that can interface with the database directly.

Advanced Reporting is a Bugzilla 3.X proposed feature.

**A.4.8.** Is there email notification and if so, what do you see when you get an email? Do you see bug number and title or is it only the number?

Email notification is user-configurable. The bug id and Topic of the bug report accompany each email notification, along with a list of the changes made.

**A.4.9.** Can email notification be set up to send to multiple people, some on the To List, CC List, BCC List etc?

Yes.

**A.4.10.** If there is email notification, do users have to have any particular type of email application?

Bugzilla email is sent in plain text, the most compatible mail format on the planet.



**Note:** If you decide to use the bugzilla\_email integration features to allow Bugzilla to record responses to mail with the associated bug, you may need to caution your users to set their mailer to "respond to messages in the format in which they were sent". For security reasons Bugzilla ignores HTML tags in comments, and if a user sends HTML-based email into Bugzilla the resulting comment looks downright awful.

**A.4.11.** If I just wanted to track certain bugs, as they go through life, can I set it up to alert me via email whenever that bug changes, whether it be owner, status or description etc.?

Yes. Place yourself in the "cc" field of the bug you wish to monitor. Then change your "Notify me of changes to" field in the Email Settings tab of the User Preferences screen in Bugzilla to the "Only those bugs which I am listed on the CC line" option.

**A.4.12.** Does Bugzilla allow data to be imported and exported? If I had outsiders write up a bug report using a MS Word bug template, could that template be imported into "matching" fields? If I wanted to take the results of a query and export that data to MS Excel, could I do that?

Mozilla allows data export through a custom DTD in XML format. It does not, however, export to specific formats other than the XML Mozilla DTD. Importing the data into Excel or any other application is left as an exercise for the reader.

If you create import filters to other applications from Mozilla's XML, please submit your modifications for inclusion in future Bugzilla distributions.

As for data import, any application can send data to Bugzilla through the HTTP protocol, or through Mozilla's XML API. However, it seems kind of silly to put another front-end in front of Bugzilla; it makes more sense to create a simplified bug submission form in HTML. You can find an excellent example at <http://www.mozilla.org/quality/help/bugzilla-helper.html>

**A.4.13.** Does Bugzilla allow fields to be added, changed or deleted? If I want to customize the bug submission form to meet our needs, can I do that using our terminology?

Yes.

**A.4.14.** Has anyone converted Bugzilla to another language to be used in other countries? Is it localizable?

Currently, no. Internationalization support for Perl did not exist in a robust fashion until the recent release of version 5.6.0; Bugzilla is, and likely will remain (until 3.X) completely non-localized.

**A.4.15.** Can a user create and save reports? Can they do this in Word format? Excel format?

Yes. No. No.

**A.4.16.** Can a user re-run a report with a new project, same query?

Yes.

**A.4.17.** Can a user modify an existing report and then save it into another name?

You can save an unlimited number of queries in Bugzilla. You are free to modify them and rename them to your heart's desire.

**A.4.18.** Does Bugzilla have the ability to search by word, phrase, compound search?

You have no idea. Bugzilla's query interface, particularly with the advanced Boolean operators, is incredibly versatile.

**A.4.19.** Can the admin person establish separate group and individual user privileges?

Yes.

**A.4.20.** Does Bugzilla provide record locking when there is simultaneous access to the same bug? Does the second person get a notice that the bug is in use or how are they notified?

Bugzilla does not lock records. It provides mid-air collision detection, and offers the offending user a choice of options to deal with the conflict.

**A.4.21.** Are there any backup features provided?

MySQL, the database back-end for Bugzilla, allows hot-backup of data. You can find strategies for dealing with backup considerations at <http://www.mysql.com/doc/B/a/Backup.html>

**A.4.22.** Can users be on the system while a backup is in progress?

Yes. However, commits to the database must wait until the tables are unlocked. Bugzilla databases are typically very small, and backups routinely take less than a minute.

**A.4.23.** What type of human resources are needed to be on staff to install and maintain Bugzilla? Specifically, what type of skills does the person need to have? I need to find out if we were to go with Bugzilla, what types of individuals would we need to hire and how much would that cost vs buying an "Out-of-the-Box" solution.

If Bugzilla is set up correctly from the start, continuing maintenance needs are minimal and can be completed by unskilled labor. Things like rotate backup tapes and check log files for the word "error".

Commercial Bug-tracking software typically costs somewhere upwards of \$20,000 or more for 5-10 floating licenses. Bugzilla consultation is available from skilled members of the newsgroup.

As an example, as of this writing I typically charge \$115 for the first hour, and \$89 each hour thereafter for consulting work. It takes me three to five hours to make Bugzilla happy on a Development installation of Linux-Mandrake.

**A.4.24.** What time frame are we looking at if we decide to hire people to install and maintain the Bugzilla? Is this something that takes hours or weeks to install and a couple of hours per week to maintain and customize or is this a multi-week install process, plus a full time job for 1 person, 2 people, etc?

It all depends on your level of commitment. Someone with much Bugzilla experience can get you up and running in less than a day, and your Bugzilla install can run untended for years. If your Bugzilla strategy is critical to your business workflow, hire somebody with reasonable UNIX or Perl skills to handle your process management and bug-tracking maintenance & customization.

**A.4.25.** Is there any licensing fee or other fees for using Bugzilla? Any out-of-pocket cost other than the bodies needed as identified above?

No. MySQL asks, if you find their product valuable, that you purchase a support contract from them that suits your needs.

### 5. Bugzilla Installation

#### A.5.1. How do I download and install Bugzilla?

Check <http://www.mozilla.org/projects/bugzilla/> for details. Once you download it, untar it, read the README and the Bugzilla Guide.

#### A.5.2. How do I install Bugzilla on Windows NT?

Installation on Windows NT has its own section in "The Bugzilla Guide".

#### A.5.3. Is there an easy way to change the Bugzilla cookie name?

At present, no.

### 6. Bugzilla Security

#### A.6.1. How do I completely disable MySQL security if it's giving me problems (I've followed the instructions in the README!)?

Run mysql like this: "mysqld --skip-grant-tables". Please remember *this makes mysql as secure as taping a \$100 to the floor of a football stadium bathroom for safekeeping*. Please read the Security section of the Administration chapter of "The Bugzilla Guide" before proceeding.

#### A.6.2. Are there any security problems with Bugzilla?

The Bugzilla code has not undergone a complete security audit. It is recommended that you closely examine permissions on your Bugzilla installation, and follow the recommended security guidelines found in the README and in The Bugzilla Guide.

#### A.6.3. I've implemented the security fixes mentioned in Chris Yeh's security advisory of 5/10/2000 advising not to run MySQL as root, and am running into problems with MySQL no longer working correctly.

This is a common problem, related to running out of file descriptors. Simply add "ulimit -n unlimited" to the script which starts mysqld.

### 7. Bugzilla Email

#### A.7.1. I have a user who doesn't want to receive any more email from Bugzilla. How do I stop it entirely for this user?

With the email changes to 2.12, the user should be able to set this in user email preferences.

#### A.7.2. I'm evaluating/testing Bugzilla, and don't want it to send email to anyone but me. How do I do it?

Edit the param for the mail text. Replace "To:" with "X-Real-To:", replace "Cc:" with "X-Real-CC:", and add a "To: (myemailaddress)".

**A.7.3.** I want whineatnews.pl to whine at something more, or other than, only new bugs. How do I do it?

Try Klaas Freitag's excellent patch for "whineatassigned" functionality. You can find it at [http://bugzilla.mozilla.org/show\\_bug.cgi?id=6679](http://bugzilla.mozilla.org/show_bug.cgi?id=6679). This patch is against an older version of Bugzilla, so you must apply the diffs manually.

**A.7.4.** I don't like/want to use Procmail to hand mail off to bug\_email.pl. What alternatives do I have?

You can call bug\_email.pl directly from your aliases file, with an entry like this:

```
bugzilla-daemon: "|usr/local/bin/bugzilla/contrib/bug_email.pl"
```

However, this is fairly nasty and subject to problems; you also need to set up your smrsh (sendmail restricted shell) to allow it. In a pinch, though, it can work.

**A.7.5.** How do I set up the email interface to submit/change bugs via email?

You can find an updated README.mailif file in the contrib/ directory of your Bugzilla distribution that walks you through the setup.

**A.7.6.** Email takes FOREVER to reach me from bugzilla — it's extremely slow. What gives?

If you are using an alternate Mail Transport Agent (MTA other than sendmail), make sure the options given in the "processmail" script for all instances of "sendmail" are correct for your MTA. If you are using Sendmail, you may wish to delete the "--ODeliveryMode=deferred" option in the "processmail" script for every invocation of "sendmail". (Be sure and leave the "-t" option, though!)

A better alternative is to change the "-O" option to "--ODeliveryMode=background". This prevents Sendmail from hanging your Bugzilla Perl processes if the domain to which it must send mail is unavailable.

This is now a configurable parameter called "sendmailparm", available from editparams.cgi.

**A.7.7.** How come email never reaches me from bugzilla changes?

Double-check that you have not turned off email in your user preferences. Confirm that Bugzilla is able to send email by visiting the "Log In" link of your Bugzilla installation and clicking the "Email me a password" button after entering your email address.

If you never receive mail from Bugzilla, chances you do not have sendmail in "/usr/lib/sendmail". Ensure sendmail lives in, or is symlinked to, "/usr/lib/sendmail".

## 8. Bugzilla Database

**A.8.1.** I've heard Bugzilla can be used with Oracle?

Red Hat Bugzilla, mentioned above, works with Oracle. The current version from Mozilla.org does not have this capability. Unfortunately, though you will sacrifice a lot of the really great features available in Bugzilla 2.10 and 2.12 if you go with the 2.8-based Redhat version.

**A.8.2.** Bugs are missing from queries, but exist in the database (and I can pull them up by specifying the bug ID). What's wrong?

You've almost certainly enabled the "shadow database", but for some reason it hasn't been updated for all your bugs. This is the database against which queries are run, so that really complex or slow queries won't lock up portions of the database for other users. You can turn off the shadow database in `editparams.cgi`. If you wish to continue using the shadow database, then as your "bugs" user run `./syncshadowdb -syncall` from the command line in the bugzilla installation directory to recreate your shadow database. After it finishes, be sure to check the params and make sure that "queryagainstshadowdb" is still turned on. The `syncshadowdb` program turns it off if it was on, and is supposed to turn it back on when completed; that way, if it crashes in the middle of recreating the database, it will stay off forever until someone turns it back on by hand. Apparently, it doesn't always do that yet.

**A.8.3.** I think my database might be corrupted, or contain invalid entries. What do I do?

Run the "sanity check" utility (`./sanitycheck.cgi` in the `bugzilla_home` directory) to see! If it all comes back, you're OK. If it doesn't come back OK (i.e. any red letters), there are certain things Bugzilla can recover from and certain things it can't. If it can't auto-recover, I hope you're familiar with `mysqladmin` commands or have installed another way to manage your database...

**A.8.4.** I want to manually edit some entries in my database. How?

There is no facility in Bugzilla itself to do this. It's also generally not a smart thing to do if you don't know exactly what you're doing. However, if you understand SQL you can use the `mysqladmin` utility to manually insert, delete, and modify table information. Personally, I use "phpMyAdmin". You have to compile a PHP module with MySQL support to make it work, but it's very clean and easy to use.

**A.8.5.** I try to add myself as a user, but Bugzilla always tells me my password is wrong.

Certain version of MySQL (notably, 3.23.29 and 3.23.30) accidentally disabled the "crypt()" function. This prevented MySQL from storing encrypted passwords. Upgrade to the "3.23 stable" version of MySQL and you should be good to go.

**A.8.6.** I think I've set up MySQL permissions correctly, but bugzilla still can't connect.

Try running MySQL from its binary: `"mysqld --skip-grant-tables"`. This will allow you to completely rule out grant tables as the cause of your frustration. However, I do not recommend you run it this way on a regular basis, unless you really want your web site defaced and your machine cracked.

**A.8.7.** How do I synchronize bug information among multiple different Bugzilla databases?

Well, you can synchronize or you can move bugs. Synchronization will only work one way -- you can create a read-only copy of the database at one site, and have it regularly updated at intervals from the main database.

MySQL has some synchronization features builtin to the latest releases. It would be great if someone looked into the possibilities there and provided a report to the newsgroup on how to effectively synchronize two Bugzilla installations.

If you simply need to transfer bugs from one Bugzilla to another, checkout the "move.pl" script in the Bugzilla distribution.

**A.8.8.** Why do I get bizarre errors when trying to submit data, particularly problems with "groupset"?

If you're sure your MySQL parameters are correct, you might want turn "strictvaluechecks" OFF in editparams.cgi. If you have "usebugsentry" set "On", you also cannot submit a bug as readable by more than one group with "strictvaluechecks" ON.

**A.8.9.** How come even after I delete bugs, the long descriptions show up?

Delete everything from \$BUZILLA\_HOME/shadow. Bugzilla creates shadow files there, with each filename corresponding to a bug number. Also be sure to run syncshadowdb to make sure, if you are using a shadow database, that the shadow database is current.

## 9. Bugzilla and Win32

**A.9.1.** What is the easiest way to run Bugzilla on Win32 (Win98+/NT/2K)?

Remove Windows. Install Linux. Install Bugzilla. The boss will never know the difference.

**A.9.2.** Is there a "Bundle::Bugzilla" equivalent for Win32?

Not currently. Bundle::Bugzilla enormously simplifies Bugzilla installation on UNIX systems. If someone can volunteer to create a suitable PPM bundle for Win32, it would be appreciated.

**A.9.3.** CGI's are failing with a "something.cgi is not a valid Windows NT application" error. Why?

Depending on what Web server you are using, you will have to configure the Web server to treat \*.cgi files as CGI scripts. In IIS, you do this by adding \*.cgi to the App Mappings with the <path>\perl.exe %s %s as the executable.

Microsoft has some advice on this matter, as well:

"Set application mappings. In the ISM, map the extension for the script file(s) to the executable for the script interpreter. For example, you might map the extension .py to Python.exe, the executable for the Python script interpreter. Note For the ActiveState Perl script interpreter, the extension .pl is associated with PerlIS.dll by default. If you want to change the association of .pl to perl.exe, you need to change the application mapping. In the mapping, you must add two percent (%) characters to the end of the pathname for perl.exe, as shown in this example: c:\perl\bin\perl.exe %s %s"

**A.9.4.** Can I have some general instructions on how to make Bugzilla on Win32 work?

The following couple entries are deprecated in favor of the Windows installation instructions available in the "Administration" portion of "The Bugzilla Guide". However, they are provided here for historical interest and insight.

1. #!C:/perl/bin/perl had to be added to every perl file.
2. Converted to Net::SMTP to handle mail messages instead of /usr/bin/sendmail.
3. The crypt function isn't available on Windows NT (at least none that I am aware), so I made encrypted passwords = plaintext passwords.
4. The system call to diff had to be changed to the Cygwin diff.
5. This was just to get a demo running under NT, it seems to be working good, and I have inserted almost 100 bugs from another bug tracking

system. Since this work was done just to get an in-house demo, I am NOT planning on making a patch for submission to Bugzilla. If you would like a zip file, let me know.

Q: Hmm, couldn't figure it out from the general instructions above. How about step-by-step?

A: Sure! Here ya go!

1. Install IIS 4.0 from the NT Option Pack #4.
2. Download and install Active Perl.
3. Install the Windows GNU tools from Cygwin. Make sure to add the bin directory to your system path. (Everyone should have these, whether they decide to use Bugzilla or not. :-) )
4. Download relevant packages from ActiveState at <http://www.activestate.com/packages/zips/>. + DBD-Mysql.zip
5. Extract each zip file with WinZip, and install each ppd file using the notation: ppm install <module>.ppd
6. Install Mysql. \*Note: If you move the default install from c:\mysql, you must add the appropriate startup parameters to the NT service. (ex. -b e:\programs\mysql)
7. Download any Mysql client. [http://www.mysql.com/download\\_win.html](http://www.mysql.com/download_win.html)
8. Setup MySQL. (These are the commands that I used.)

I. Cleanup default database settings.

```
C:\mysql\bin\mysql -u root mysql
mysql> DELETE FROM user WHERE Host='localhost' AND User="";
mysql> quit
C:\mysql\bin\mysqladmin reload
```

II. Set password for root.

```
C:\mysql\bin\mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('new_password')
WHERE user='root';
mysql> FLUSH PRIVILEGES;
mysql> quit
C:\mysql\bin\mysqladmin -u root reload
```

III. Create bugs user.

```
C:\mysql\bin\mysql -u root -p
mysql> insert into user (host,user,password)
values('localhost','bugs','");
mysql> quit
C:\mysql\bin\mysqladmin -u root reload
```

IV. Create the bugs database.

```
C:\mysql\bin\mysql -u root -p
mysql> create database bugs;
```

V. Give the bugs user access to the bugs database.

```
mysql> insert into db
(host,db,user,select_priv,insert_priv,update_priv,delete_priv,create_priv,drop_priv)
```

```
values('localhost','bugs','bugs','Y','Y','Y','Y','Y','N')
mysql> quit
C:\mysql\bin\mysqladmin -u root reload
```

9. Run the table scripts to setup the bugs database.
10. Change CGI.pm to use the following regular expression because of differing backslashes in NT versus UNIX.
  - o \$0 =~ m:[^\\]\*\$;
11. Had to make the crypt password = plain text password in the database. (Thanks to Andrew Lahser "andrew\_lahser@merck.com" on this one.) The files that I changed were:
  - o globals.pl
  - o CGI.pl
  - o alternately, you can try commenting all references to 'crypt' string and replace them with similar lines but without encrypt() or crypr() functions insida all files.
12. Replaced sendmail with Windmail. Basically, you have to come up with a sendmail substitute for NT. Someone said that they used a Perl module (Net::SMTP), but I was trying to save time and do as little Perl coding as possible.
13. Added "perl" to the beginning of all Perl system calls that use a perl script as an argument and renamed processmail to processmail.pl.
14. In processmail.pl, I added binmode(HANDLE) before all read() calls. I'm not sure about this one, but the read() under NT wasn't counting the EOLs without the binary read."

**A.9.5.** I'm having trouble with the perl modules for NT not being able to talk to to the database.

Your modules may be outdated or inaccurate. Try:

1. Hitting <http://www.activestate.com/ActivePerl>
2. Download ActivePerl
3. Go to your prompt
4. Type 'ppm'
5. PPM> **install DBI DBD-mysql GD**

I reckon TimeDate and Data::Dumper come with the activeperl. You can check the ActiveState site for packages for installation through PPM. <http://www.activestate.com/Packages/>

## 10. Bugzilla Usage

**A.10.1.** The query page is very confusing. Isn't there a simpler way to query?

We are developing in that direction. You can follow progress on this at [http://bugzilla.mozilla.org/show\\_bug.cgi?id=16775](http://bugzilla.mozilla.org/show_bug.cgi?id=16775). Some functionality is available in Bugzilla 2.12, and is available as "quicksearch.html"

**A.10.2.** I'm confused by the behavior of the "accept" button in the Show Bug form. Why doesn't it assign the bug to me when I accept it?

The current behavior is acceptable to bugzilla.mozilla.org and most users. I personally don't like it. You have your choice of patches to change this behavior, however.



[Add a "and accept bug" radio button](#)  
["Accept" button automatically assigns to you](#)

Note that these patches are somewhat dated. You will need to do the find and replace manually to apply them. They are very small, though. It is easy.

**A.10.3.** I can't upload anything into the database via the "Create Attachment" link. What am I doing wrong?

The most likely cause is a very old browser or a browser that is incompatible with file upload via POST. Download the latest Netscape, Microsoft, or Mozilla browser to handle uploads correctly.

**A.10.4.** Email submissions to Bugzilla that have attachments end up asking me to save it as a "cgi" file.

Yup. Just rename it once you download it, or save it under a different filename. This will not be fixed anytime too soon, because it would cripple some other functionality.

**A.10.5.** How do I change a keyword in Bugzilla, once some bugs are using it?

In the Bugzilla administrator UI, edit the keyword and it will let you replace the old keyword name with a new one. This will cause a problem with the keyword cache. Run `sanitycheck.cgi` to fix it.

## 11. Bugzilla Hacking

**A.11.1.** What bugs are in Bugzilla right now?

Try [this link](#) to view current bugs or requests for enhancement for Bugzilla.

You can view bugs marked for 2.14 release [here](#). This list includes bugs for the 2.14 release that have already been fixed and checked into CVS. Please consult the [Bugzilla Project Page](#) for details on how to check current sources out of CVS so you can have these bug fixes early!

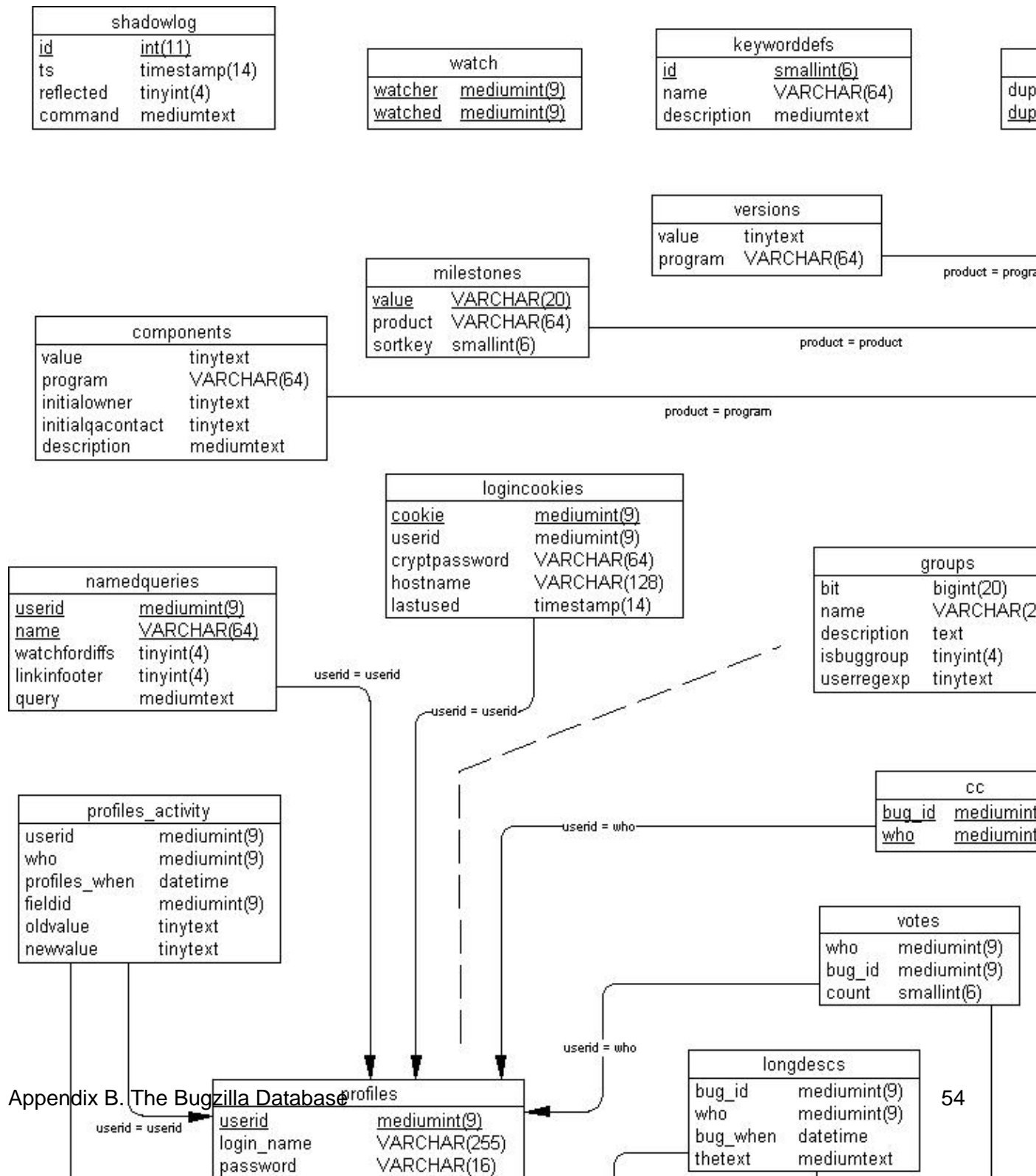
**A.11.2.** What's the best way to submit patches? What guidelines should I follow?

1. Enter a bug into `bugzilla.mozilla.org` for the "Webtools" product, "Bugzilla" component.
  2. Upload your patch as a unified DIFF (having used `diff -u` against the *current sources* checked out of CVS), or new source file by clicking "Create a new attachment" link on the bug page you've just created, and include any descriptions of database changes you may make, into the bug ID you submitted in step #1. Be sure and click the "Patch" radio button to indicate the text you are sending is a patch!
  3. Announce your patch and the associated URL (`http://bugzilla.mozilla.org/show_bug.cgi?id=XXXX`) for discussion in the newsgroup (`netscape.public.mozilla.webtools`). You'll get a really good, fairly immediate reaction to the implications of your patch, which will also give us an idea how well-received the change would be.
  4. If it passes muster with minimal modification, the person to whom the bug is assigned in Bugzilla is responsible for seeing the patch is checked into CVS.
  5. Bask in the glory of the fact that you helped write the most successful open-source bug-tracking software on the planet :)
-

# Appendix B. The Bugzilla Database

**Note:** This document really needs to be updated with more fleshed out information about primary keys, interrelationships, and maybe some nifty tables to document dependencies. Any takers?

## B.1. Database Schema Chart



## B.2. MySQL Bugzilla Database Introduction

Contributor(s): Matthew P. Barnson (mbarnson@excitehome.net)

Last update: May 16, 2000

Changes:

Version 1.0: Initial public release (May 16, 2000)

Maintainer: Matthew P. Barnson (mbarnson@excitehome.net)

===

Table Of Contents

===

FOREWORD  
INTRODUCTION  
THE BASICS  
THE TABLES  
THE DETAILS

===

FOREWORD

===

This information comes straight from my life. I was forced to learn how Bugzilla organizes database because of nitpicky requests from users for tiny changes in wording, rather than having people re-educate themselves or figure out how to work our procedures around the tool. It sucks, but it can and will happen to you, so learn how the schema works and deal with it when it comes.

I'm sorry this version is plain text. I can whip this info out a lot faster if I'm not concerned about complex formatting. I'll get it into sgml for easy portability as time permits.

The Bugzilla Database Schema has a home! In addition to availability via CVS and released versions 2.12 and higher of Bugzilla, you can find the latest & greatest version of the Bugzilla Database Schema at <http://www.trilobyte.net/barnsons/>. This is a living document; please be sure you are up-to-date with the latest version before mirroring.

The Bugzilla Database Schema is designed to provide vital information

## The Bugzilla Guide

regarding the structure of the MySQL database. Where appropriate, this document will refer to URLs rather than including documents in their entirety to ensure completeness even should this paper become out of date.

This document is not maintained by Netscape or Netscape employees, so please do not contact them regarding errors or omissions contained herein. Please direct all questions, comments, updates, flames, etc. to Matthew P. Barnson [mbarnson@excitehome.net](mailto:mbarnson@excitehome.net) (barnboy or barnhome on [#irc.mozilla.org](irc://irc.mozilla.org) in #mozwebtools).

I'm sure I've made some glaring errors or omissions in this paper -- please email me corrections or post corrections to the [netscape.public.mozilla.webtools](mailto:netscape.public.mozilla.webtools@netscape.com) newsgroup.

===

### INTRODUCTION

===

So, here you are with your brand-new installation of Bugzilla. You've got MySQL set up, Apache working right, Perl DBI and DBD talking to the database flawlessly. Maybe you've even entered a few test bugs to make sure email's working; people seem to be notified of new bugs and changes, and you can enter and edit bugs to your heart's content. Perhaps you've gone through the trouble of setting up a gateway for people to submit bugs to your database via email, have had a few people test it, and received rave reviews from your beta testers.

What's the next thing you do? Outline a training strategy for your development team, of course, and bring them up to speed on the new tool you've labored over for hours.

Your first training session starts off very well! You have a captive audience which seems enraptured by the efficiency embodied in this thing called "Bugzilla". You are caught up describing the nifty features, how people can save favorite queries in the database, set them up as headers and footers on their pages, customize their layouts, generate reports, track status with greater efficiency than ever before, leap tall buildings with a single bound and rescue Jane from the clutches of Certain Death!

But Certain Death speaks up -- a tiny voice, from the dark corners of the conference room. "I have a concern," the voice hisses from the darkness, "about the use of the word 'verified'.

The room, previously filled with happy chatter, lapses into reverential silence as Certain Death (better known as the Vice President of Software Engineering) continues. "You see, for two years we've used the word 'verified' to indicate that a developer or quality assurance engineer has confirmed that,

in fact, a bug is valid. I don't want to lose two years of training to a new software product. You need to change the bug status of 'verified' to 'approved' as soon as possible. To avoid confusion, of course."

Oh no! Terror strikes your heart, as you find yourself mumbling "yes, yes, I don't think that would be a problem," You review the changes with Certain Death, and continue to jabber on, "no, it's not too big a change. I mean, we have the source code, right? You know, 'Use the Source, Luke' and all that... no problem," All the while you quiver inside like a beached jellyfish bubbling, burbling, and boiling on a hot Jamaican sand dune...

Thus begins your adventure into the heart of Bugzilla. You've been forced to learn about non-portable enum() fields, varchar columns, and tinyint definitions. The Adventure Awaits You!

===

### The Basics

===

If you were like me, at this point you're totally clueless about the internals of MySQL, and if it weren't for this executive order from the Vice President you couldn't care less about the difference between a "bigint" and a "tinyint" entry in MySQL. I'd refer you first to the MySQL documentation, available at <http://www.mysql.com/doc.html>, but that's mostly a confusing morass of high-level database jargon. Here are the basics you need to know about the database to proceed:

1. To connect to your database, type "mysql -u root" at the command prompt as any user. If this works without asking you for a password, SHAME ON YOU! You should have locked your security down like the README told you to. You can find details on locking down your database in the Bugzilla FAQ in this directory (under "Security"), or more robust security generalities in the MySQL searchable documentation at [http://www.mysql.com/php/manual.php3?section=Privilege\\_system](http://www.mysql.com/php/manual.php3?section=Privilege_system) .

2. You should now be at a prompt that looks like this:

```
mysql>
```

At the prompt, if "bugs" is the name of your Bugzilla database, type:

```
mysql> use bugs;
```

(don't forget the ";" at the end of each line, or you'll be kicking yourself all the way through this documentation)

Young Grasshopper, you are now ready for the unveiling of the Bugzilla database, in the next section...

## === THE TABLES ===

Imagine your MySQL database as a series of spreadsheets, and you won't be too far off. If you use this command:

```
mysql> show tables from bugs;
```

you'll be able to see all the "spreadsheets" (tables) in your database. Cool, huh? It's kinda' like a filesystem, only much faster and more robust. Come on, I'll show you more!

From the command issued above, you should now have some output that looks like this:

```
+-----+
| Tables in bugs |
+-----+
| attachments |
| bugs        |
| bugs_activity |
| cc          |
| components  |
| dependencies |
| fielddefs   |
| groups      |
| keyworddefs |
| keywords    |
| logincookies |
| longdescs   |
| milestones  |
| namedqueries |
| products    |
| profiles    |
| profiles_activity |
| shadowlog   |
| versions    |
| votes       |
| watch       |
+-----+
```

If it doesn't look quite the same, that probably means it's time to update this documentation :)

Here's an overview of what each table does. Most columns in each table have descriptive names that make it fairly trivial to figure out their jobs.

attachments: This table stores all attachments to bugs. It tends to be your

## The Bugzilla Guide

largest table, yet also generally has the fewest entries because file attachments are so (relatively) large.

bugs: This is the core of your system. The bugs table stores most of the current information about a bug, with the exception of the info stored in the other tables.

bugs\_activity: This stores information regarding what changes are made to bugs when -- a history file.

cc: This tiny table simply stores all the CC information for any bug which has any entries in the CC field of the bug. Note that, like most other tables in Bugzilla, it does not refer to users by their user names, but by their unique userid, stored as a primary key in the profiles table.

components: This stores the programs and components (or products and components, in newer Bugzilla parlance) for Bugzilla. Curiously, the "program" (product) field is the full name of the product, rather than some other unique identifier, like bug\_id and user\_id are elsewhere in the database.

dependencies: Stores data about those cool dependency trees.

fielddefs: A nifty table that defines other tables. For instance, when you submit a form that changes the value of "AssignedTo" this table allows translation to the actual field name "assigned\_to" for entry into MySQL.

groups: defines bitmasks for groups. A bitmask is a number that can uniquely identify group memberships. For instance, say the group that is allowed to tweak parameters is assigned a value of "1", the group that is allowed to edit users is assigned a "2", and the group that is allowed to create new groups is assigned the bitmask of "4". By uniquely combining the group bitmasks (much like the chmod command in UNIX,) you can identify a user is allowed to tweak parameters and create groups, but not edit users, by giving him a bitmask of "5", or a user allowed to edit users and create groups, but not tweak parameters, by giving him a bitmask of "6" Simple, huh?

If this makes no sense to you, try this at the mysql prompt:

```
mysql> select * from groups;
```

You'll see the list, it makes much more sense that way.

keyworddefs: Definitions of keywords to be used

keywords: Unlike what you'd think, this table holds which keywords are associated with which bug id's.

logincookies: This stores every login cookie ever assigned to you for every machine you've ever logged into Bugzilla from. Curiously, it never does any housecleaning -- I see cookies in this file I've not used for months. However, since Bugzilla never expires your cookie (for convenience' sake), it makes sense.

longdescs: The meat of bugzilla -- here is where all user comments are stored!

## The Bugzilla Guide

You've only got 2<sup>24</sup> bytes per comment (it's a mediumtext field), so speak sparingly — that's only the amount of space the Old Testament from the Bible would take (uncompressed, 16 megabytes). Each comment is keyed to the bug\_id to which it's attached, so the order is necessarily chronological, for comments are played back in the order in which they are received.

milestones: Interesting that milestones are associated with a specific product in this table, but Bugzilla does not yet support differing milestones by product through the standard configuration interfaces.

namedqueries: This is where everybody stores their "custom queries". Very cool feature; it beats the tar out of having to bookmark each cool query you construct.

products: What products you have, whether new bug entries are allowed for the product, what milestone you're working toward on that product, votes, etc. It will be nice when the components table supports these same features, so you could close a particular component for bug entry without having to close an entire product...

profiles: Ahh, so you were wondering where your precious user information was stored? Here it is! With the passwords in plain text for all to see! (but sshh... don't tell your users!)

profiles\_activity: Need to know who did what when to who's profile? This'll tell you, it's a pretty complete history.

shadowlog: I could be mistaken here, but I believe this table tells you when your shadow database is updated and what commands were used to update it. We don't use a shadow database at our site yet, so it's pretty empty for us.

versions: Version information for every product

votes: Who voted for what when

watch: Who (according to userid) is watching who's bugs (according to their userid).

===

### THE DETAILS

===

Ahh, so you're wondering just what to do with the information above? At the mysql prompt, you can view any information about the columns in a table with this command (where "table" is the name of the table you wish to view):

```
mysql> show columns from table;
```

You can also view all the data in a table with this command:



## The Bugzilla Guide

```
mysql> select * from table;
```

— note: this is a very bad idea to do on, for instance, the "bugs" table if you have 50,000 bugs. You'll be sitting there a while until you ctrl-c or 50,000 bugs play across your screen.

You can limit the display from above a little with the command, where "column" is the name of the column for which you wish to restrict information:

```
mysql> select * from table where (column = "some info");
```

— or the reverse of this

```
mysql> select * from table where (column != "some info");
```

Let's take our example from the introduction, and assume you need to change the word "verified" to "approved" in the resolution field. We know from the above information that the resolution is likely to be stored in the "bugs" table. Note we'll need to change a little perl code as well as this database change, but I won't plunge into that in this document. Let's verify the information is stored in the "bugs" table:

```
mysql> show columns from bugs
```

(exceedingly long output truncated here)

```
| bug_status| enum('UNCONFIRMED','NEW','ASSIGNED','REOPENED','RESOLVED','VERIFIED','CLOSED')||MU
```

Sorry about that long line. We see from this that the "bug status" column is an "enum field", which is a MySQL peculiarity where a string type field can only have certain types of entries. While I think this is very cool, it's not standard SQL. Anyway, we need to add the possible enum field entry 'APPROVED' by altering the "bugs" table.

```
mysql> ALTER table bugs CHANGE bug_status bug_status
-> enum("UNCONFIRMED", "NEW", "ASSIGNED", "REOPENED", "RESOLVED",
-> "VERIFIED", "APPROVED", "CLOSED") not null;
```

(note we can take three lines or more — whatever you put in before the semicolon is evaluated as a single expression)

Now if you do this:

```
mysql> show columns from bugs;
```

you'll see that the bug\_status field has an extra "APPROVED" enum that's available! Cool thing, too, is that this is reflected on your query page as well — you can query by the new status. But how's it fit into the existing scheme of things?

Looks like you need to go back and look for instances of the word "verified" in the perl code for Bugzilla — wherever you find "verified", change it to "approved" and you're in business (make sure that's a case-insensitive search).

Although you can query by the enum field, you can't give something a status of "APPROVED" until you make the perl changes. Note that this change I mentioned can also be done by editing checksetup.pl, which automates a lot of this. But you need to know this stuff anyway, right?

I hope this database tutorial has been useful for you. If you have comments to add, questions, concerns, etc. please direct them to mbarnson@excitehome.net. Please direct flames to /dev/null :) Have a nice day!

===  
LINKS  
===

Great MySQL tutorial site:  
[http://www.devshed.com/Server\\_Side/MySQL/](http://www.devshed.com/Server_Side/MySQL/)

---

### B.3. MySQL Permissions & Grant Tables

**Note:** The following portion of documentation comes from my answer to an old discussion of Keynote, a cool product that does trouble-ticket tracking for IT departments. I wrote this post to the Keynote support group regarding MySQL grant table permissions, and how to use them effectively. It is badly in need of updating, as I believe MySQL has added a field or two to the grant tables since this time, but it serves as a decent introduction and troubleshooting document for grant table issues. I used Keynote to track my troubles until I discovered Bugzilla, which gave me a whole new set of troubles to work on : )

From matt\_barnson@singletrac.com Wed Jul 7 09:00:07 1999  
Date: Mon, 1 Mar 1999 21:37:04 -0700  
From: Matthew Barnson matt\_barnson@singletrac.com  
To: keystone-users@homeport.org  
Subject: [keystone-users] Grant Tables FAQ

[The following text is in the "iso-8859-1" character set]  
[Your display is set for the "US-ASCII" character set]  
[Some characters may be displayed incorrectly]

Maybe we can include this rambling message in the Keystone FAQ? It gets asked a lot, and the only option current listed in the FAQ is "—skip-grant-tables".

Really, you can't go wrong by reading section 6 of the MySQL manual, at <http://www.mysql.com/Manual/manual.html>. I am sure their description is better than mine.

MySQL runs fine without permissions set up correctly if you run the mysql daemon with the "--skip-grant-tables" option. Running this way denies access to nobody. Unfortunately, unless you've got yourself firewalled it also opens the potential for abuse if someone knows you're running it.

Additionally, the default permissions for MySQL allow anyone at localhost access to the database if the database name begins with "test\_" or is named "test" (i.e. "test\_keystone"). You can change the name of your database in the keystone.conf file (\$sys\_dbname). This is the way I am doing it for some of my databases, and it works fine.

The methods described below assume you're running MySQL on the same box as your webserver, and that you don't mind if your \$sys\_dbuser for Keystone has superuser access. See near the bottom of this message for a description of what each field does.

### Method #1:

#### 1. cd /var/lib

#location where you'll want to run /usr/bin/mysql\_install\_db shell script from to get it to work.

#### 2. ln -s mysql data

# soft links the "mysql" directory to "data", which is what mysql\_install\_db expects. Alternately, you can edit mysql\_install\_db and change all the "./data" references to "./mysql".

#### 3. Edit /usr/bin/mysql\_install\_db with your favorite text editor (vi, emacs, jot, pico, etc.)

##### A) Copy the "INSERT INTO db VALUES

('%', 'test\\_%', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');" and paste it immediately after itself. Change the 'test\\_%' value to 'keystone', or the value of \$sys\_dbname in keystone.conf.

##### B) If you are running your keystone database with any user, you'll need to copy the "INSERT INTO user VALUES

('localhost', 'root', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');" line after itself and change 'root' to the name of the keystone database user (\$sys\_dbuser) in keystone.conf.

# adds entries to the script to create grant tables for specific hosts and users. The user you set up has super-user access (\$sys\_dbuser) — you may or may not want this. The layout of mysql\_install\_db is really very uncomplicated.

#### 4. /usr/bin/mysqladmin shutdown

# ya gotta shut it down before you can reinstall the grant tables!

#### 5. rm -i /var/lib/mysql/mysql/\*.\*.IS? and answer 'Y' to the deletion questions.

# nuke your current grant tables. This WILL NOT delete any other databases than your grant tables.

6. `/usr/bin/mysql_install_db`

# run the script you just edited to install your new grant tables.

7. `mysqladmin -u root password (new_password)`

# change the root MySQL password, or else anyone on localhost can login to MySQL as root and make changes. You can skip this step if you want keystone to connect as root with no password.

8. `mysqladmin -u (webservice_user_name) password (new_password)`

# change the password of the `$sys_dbuser`. Note that you will need to change the password in the `keystone.conf` file as well in `$sys_dbpasswd`, and if your permissions are set up incorrectly anybody can type the URL to your `keystone.conf` file and get the password. Note that this will help them much if your permissions are set to `@localhost`.

Method #2: easier, but a pain reproducing if you have to delete your grant tables. This is the "recommended" method for altering grant tables in MySQL. I don't use it because I like the other way :)

```
shell> mysql --user=root keystone
```

```
mysql> GRANT
```

```
SELECT,INSERT,UPDATE,DELETE,INDEX,ALTER,CREATE,DROP,RELOAD,SHUTDOWN,PROCESS,  
FILE,
```

```
    ON keystone.*
```

```
    TO <$sys_dbuser name>@localhost
```

```
    IDENTIFIED BY '(password)'
```

```
    WITH GRANT OPTION;
```

OR

```
mysql> GRANT ALL PRIVILEGES
```

```
ON keystone.*
```

```
TO <$sys_dbuser name>@localhost
```

```
IDENTIFIED BY '(password)'
```

```
WITH GRANT OPTION;
```

# this grants the required permissions to the keystone (`$sys_dbuser`) account defined in `keystone.conf`. However, if you are running many different MySQL-based apps, as we are, it's generally better to edit the `mysql_install_db` script to be able to quickly reproduce your permissions structure again. Note that the `FILE` privilege and `WITH GRANT OPTION` may not be in your best interest to include.

GRANT TABLE FIELDS EXPLANATION:

Quick syntax summary: "%" in MySQL is a wildcard. I.E., if you are defining your DB table and in the 'host' field and enter '%', that means

that any host can access that database. Of course, that host must also have a valid db user in order to do anything useful. 'db'=name of database. In our case, it should be "keystone". "user" should be your "\$sys\_dbuser" defined in keystone.conf. Note that you CANNOT add or change a password by using the "INSERT INTO db (X)" command -- you must change it with the mysql -u command as defined above. Passwords are stored encrypted in the MySQL database, and if you try to enter it directly into the table they will not match.

TABLE: USER. Everything after "password" is a privilege granted (Y/N). This table controls individual user global access rights.

```
'host','user','password','select','insert','update','delete','index','alter',  
, 'create','drop','grant','reload','shutdown','process','file'
```

TABLE: DB. This controls access of USERS to databases.

```
'host','db','user','select','insert','update','delete','index','alter','crea  
te','drop','grant'
```

TABLE: HOST. This controls which HOSTS are allowed what global access rights. Note that the HOST table, USER table, and DB table are very closely connected -- if an authorized USER attempts an SQL request from an unauthorized HOST, she's denied. If a request from an authorized HOST is not an authorized USER, it is denied. If a globally authorized USER does not have rights to a certain DB, she's denied. Get the picture?

```
'host','db','select','insert','update','delete','index','alter','create','dr  
op','grant'
```

You should now have a working knowledge of MySQL grant tables. If there is anything I've left out of this answer that you feel is pertinent, or if my instructions don't work for you, please let me know and I'll re-post this letter again, corrected. I threw it together one night out of exasperation for all the newbies who don't know squat about MySQL yet, so it is almost guaranteed to have errors.

Once again, you can't go wrong by reading section 6 of the MySQL manual. It is more detailed than I!  
<http://www.mysql.com/Manual/manual.html>.

---

10/12/2000

Matthew sent in some mail with updated contact information:  
NEW CONTACT INFORMATION:

```
-----  
Matthew P. Barnson  
Manager, Systems Administration  
Excite@Home Business Applications
```

mbarnson@excitehome.net  
(801)234-8300

---

## B.4. Cleaning up after mucking with Bugzilla

Contributed by Eric Hansen:

There are several things, and one trick. There is a small tiny piece of documentation I saw once that said something very important.

1) After pretty much any manual working of the Mysql db, you must delete a file in the bugzilla directory: data/versioncache

Versioncache basically is a way to speed up bugzilla (from what I understand). It stores a lot of commonly used information. However, this file is refreshed every so often (I can't remember the time interval though). So eventually all changes do propogate out, so you may see stuff suddenly working.

2) Assuming that failed, you will also have to check something with the checksetup.pl file. It actually is run twice. The first time it creates the file: localconfig. You can modify localconfig, (or not if you are doing bug\_status stuff) or you should delete localconfig and rerun your modified checksetup.pl. Since I don't actually see anything in localconfig pertaining to bug\_status, this point is mainly a FYI.

---

# Appendix C. Useful Patches and Utilities for Bugzilla

## C.1. The setperl.pl Utility

You can use the "setperl.pl" utility to quickly and easily change the path to perl on all your Bugzilla files.

1. Download the "setperl.pl" utility to your Bugzilla directory and make it executable.
  - a. bash# `cd /your/path/to/bugzilla`
  - b. bash# `wget -O setperl.pl 'http://bugzilla.mozilla.org/showattachment.cgi?attach_id=10795'`
  - c. bash# `chmod u+x setperl.pl`
2. Prepare (and fix) Bugzilla file permissions.
  - a. bash# `chmod u+w *`
  - b. bash# `chmod u+x duplicates.cgi`
  - c. bash# `chmod a-x bug_status.html`
3. Run the script:

```
bash# ./setperl.pl /your/path/to/perl
```

---

## C.2. Command-line Bugzilla Queries

Users can query Bugzilla from the command line using this suite of utilities.

The query.conf file contains the mapping from options to field names and comparison types. Quoted option names are "grepped" for, so it should be easy to edit this file. Comments (#) have no effect; you must make sure these lines do not contain any quoted "option"

buglist is a shell script which submits a Bugzilla query and writes the resulting HTML page to stdout. It supports both short options, (such as "-Afoo" or "-Rbar") and long options (such as "--assignedto=foo" or "--reporter=bar"). If the first character of an option is not "-", it is treated as if it were prefixed with "--default=".

The columlist is taken from the COLUMNLIST environment variable. This is equivalent to the "Change Columns" option when you list bugs in buglist.cgi. If you have already used Bugzilla, use **grep COLUMNLIST ~/.netscape/cookies** to see your current COLUMNLIST setting.

bugs is a simple shell script which calls buglist and extracts the bug numbers from the output. Adding the prefix "http://bugzilla.mozilla.org/buglist.cgi?bug\_id=" turns the bug list into a working link if any bugs are found. Counting bugs is easy. Pipe the results through `sed -e 's/,/ /g' | wc | awk '{printf $2 "\n"}'`

Akkana says she has good results piping buglist output through `w3m -T text/html -dump`

1. Download three files:
  - a. bash\$ `wget -O query.conf 'http://bugzilla.mozilla.org/showattachment.cgi?attach_id=26157'`

```
b. bash$ wget -O buglist
   'http://bugzilla.mozilla.org/showattachment.cgi?attach_id=26944'
c. bash# wget -O bugs
   'http://bugzilla.mozilla.org/showattachment.cgi?attach_id=26215'
```

2. Make your utilities executable: bash\$ `chmod u+x buglist bugs`

---

## C.3. The Quicksearch Utility

Quicksearch is a new, experimental feature of the 2.12 release. It consist of two Javascript files, "quicksearch.js" and "localconfig.js", and two documentation files, "quicksearch.html" and "quicksearchhack.html"

The index.html page has been updated to include the QuickSearch text box.

To take full advantage of the query power, the Bugzilla maintainer must edit "localconfig.js" according to the value sets used in the local installation.

Currently, keywords must be hard-coded in localconfig.js. If they are not, keywords are not automatically recognized. This means, if localconfig.js is left unconfigured, that searching for a bug with the "foo" keyword will only find bugs with "foo" in the summary, status whiteboard, product or component name, but not those with the keyword "foo".

Workarounds for Bugzilla users:

search for '!foo' (this will find only bugs with the keyword "foo")

search 'foo,!foo' (equivalent to 'foo OR keyword:foo')

When this tool is ported from client-side JavaScript to server-side Perl, the requirement for hard-coding keywords can be fixed. [This bug](#) has details.

---



# Appendix D. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

---

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

---

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

---

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

---

## 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the

Document, free of added material, which the general network—using public has access to download anonymously at no charge using public—standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

---

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
  - I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
  - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
  - K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
  - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
  - M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
  - N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

---

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

---

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

---

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

---

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

---

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

---

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

---

## How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

# Glossary

**A**

*There are no entries for A*



# B

## *Bug*

A "Bug" in Bugzilla refers to an issue entered into the database which has an associated number, assignments, comments, etc. Many also refer to a "Ticket" or "Issue"; in this context, they are synonymous.

## *Bug Number*

Each Bugzilla Bug is assigned a number that uniquely identifies that Bug. The Bug associated with a Bug Number can be pulled up via a query, or easily from the very front page by typing the number in the "Find" box.

## *Bug Life Cycle*

A Bug has stages through which it must pass before becoming a "closed bug", including acceptance, resolution, and verification. The "Bug Life Cycle" is moderately flexible according to the needs of the organization using it, though.

I

## *Infinite Loop*

See: [Recursion](#)

# P

## *Product*

A Product is a broad category of types of bugs. In general, there are several Components to a Product. A Product also defines a default Group (used for Bug Security) for all bugs entered into components beneath it.

### **Example 1. A Sample Product**

A company sells a software product called "X". They also maintain some older software called "Y", and have a secret project "Z". An effective use of Products might be to create Products "X", "Y", and "Z", each with Components "User Interface", "Database", and "Business Logic". They might also change group permissions so that only those people who are members of Group "Z" can see components and bugs under Product "Z".

## Q

### *Q/A*

"Q/A" is short for "Quality Assurance". In most large software development organizations, there is a team devoted to ensuring the product meets minimum standards before shipping. This team will also generally want to track the progress of bugs over their life cycle, thus the need for the "Q/A Contact" field in a Bug.

# R

## *Recursion*

See: [Infinite Loop](#)

# Z

## *Zarro Boogs Found*

This is the cryptic response sent by Bugzilla when a query returned no results. It is just a goofy way of saying "Zero Bugs Found".

---