

Mozilla's Tools for Developers

Mozilla's Firefox and Thunderbird are both open source, where the current source code is freely available. Anyone can modify and build the product, creating a new version of Firefox that might improve performance, add functionality, enhance performance using the compiler's CPU-specific features, and even fix bugs.

You have two choices as to what happens to your work. The first choice is to keep it as your private work. The second choice is to make it part of the official product's code.

Open Source As a Way of Life

Open Source (<http://www.opensource.org/>) is a software license philosophy that strongly encourages the free distribution of usable (unobfuscated) source code. (*Obfuscating* is a process in which source code is intentionally made unreadable by people but is still functional.)

The idea behind open source is that, with freely distributed source code, software will naturally evolve and improve as it is worked on by hundreds or thousands of independent programmers.

Public license is the term used to denote source code that is released for public use, although there might be restrictions on this usage. Several licenses have been created. Some are compatible with each other, and some are not.

The Netscape Public License (NPL) is the license Netscape used to release the source for the Navigator web browser. This license contains special considerations for Netscape, which some developers do not feel are acceptable. Anything derived from NPL is covered by NPL, so if you change a NPL licensed file, the result, in its entirety, is also licensed under NPL.

The Mozilla Public License (MozPL) is the same as NPL, with the deletion of any provisions for preferential treatment of Netscape. If you create code files that are completely new, without any NPL licensed code, you can elect to have your code covered under MozPL in lieu of NPL.



17

IN THIS CHAPTER

- Open Source As a Way of Life
- Developer Tools
- Mozilla Developer Support Tools
- Bugzilla
- Tinderbox
- Bonsai
- LXR
- Working with the Mozilla Development Tools



BSD, developed by the University of California at Berkley, is a license that allows virtually unlimited use of source code, requiring only that the copyright holder be referenced in the source code.

Tip

The Netscape Public License contains the provision that Netscape can use anything licensed using the NPL in its commercial products. This provision is the only exception in the making of profit on open-source software.

Anything you do to Firefox and the other related products (such as Thunderbird) can be used by Netscape. I am unaware, however, of any cases where this has caused any problems.

This provision was necessary because Netscape used the same technology in other products (such as servers) and had commitments to continue selling the Netscape Navigator browser to some commercial customers. Netscape continues to this day to offer its browser, which is now based on Firefox. It can be downloaded free or a CD version can be ordered.

The GNU General Public License (GPL) makes source code freely available but has a very complex set of rules. This license is somewhat incompatible with the other licenses.

The GNU Lesser General Public License (LGPL) is used to protect code so that some derivative works would be not covered by the GPL. (It was originally called the GNU Library General Public License.)

The Artistic License is similar to GPL, in that if software code is distributed outside the organization, it must be freely distributed.

Copyleft is essentially having your work free and requiring that all derivative work also be free—essentially, it is GPL.

With that collection of licenses, anyone would be confused. A reasonable summary of Open Source follows:

- Source is freely distributable to anyone, and anyone can distribute the source. Most licenses allow only for a nominal copy charge (those who do usually sell a CD-R for about \$10.)
- You can add to, modify, or improve the original. If you distribute your product to the public, so must you with the entire source code.
- You can make derivative products using some or the entire source. Your derivative product would be bound by the same rules.

Is Open Source here to stay? Absolutely. Will Microsoft ever adopt (even in a limited way) Open Source? Possibly. Several groups have suggested that Microsoft release the source for Windows 95. Microsoft does offer a free command-line version of the C/C++ compiler (not source, however) at <http://msdn.microsoft.com/visualc/vctoolkit2003/>. This compiler and linker comes with all the necessary standard include and library files and some sample programs you can build.

<http://www.whereswalden.com/mozilla/msvcfree/> contains an excellent article by MIT student Jeff Walden on retrieving additional free Microsoft components and building Mozilla programs.

Developer Tools

There are many tools available to assist in writing themes, extensions, other add-ons and building the Mozilla products.

Official documentation from Mozilla is available. You can find a FAQ, helpful to beginning users, and tips and techniques for more experienced users.

Developers will find a number of tools.

NOTE

Some developer support information found in this section comes from Mozilla. This information can be rather dynamic, changing rapidly as time passes. Always check Mozilla's site at <http://www.mozilla.org/support/>.

Mozilla Developer Support Tools

When many programmers and developers are working on a project and they are not part of a tightly organized team, tools are vitally necessary!

Fortunately, developers have some great avenues of support. Mozilla has an Internet Relay Chat (IRC) server that hosts chat forums for all its products. Firefox doesn't support IRC directly, so for an IRC client, consider ChatZilla—an extension for Firefox.

NOTE

Although ChatZilla is a Firefox extension, it does not run in a Firefox window. Instead, ChatZilla creates its own window that is independent from Firefox.

There are four main developer tools: Bugzilla, Tinderbox, Bonsai, and LXR. Each of these tools is found at <http://www.mozilla.org/developer/>. If you are going to contribute to the mainstream Firefox development effort, you will need to know and understand each of these tools.

Bugzilla

Bugzilla is the tool used at Mozilla to track bugs in the various products. This program is accessible over the Internet using Firefox (or another browser).

The problems of tracking and managing bugs in software can be daunting, especially when many developers are involved. Mozilla's problems are even more difficult because it has no direct control over many of the developers.

Bugzilla resolves many of the problems that bug management present and is used to track and manage bugs in Firefox, Thunderbird, Mozilla, and Camino.

NOTE

Bugzilla is used by several other organizations to track their software bugs. In fact, more than 400 organizations, companies, and projects were using Bugzilla at the time of this writing. If you want to use Bugzilla, check <http://www.bugzilla.org/download/> for download instructions.

You can search for bugs, file bugs, comment on bugs, and be assigned a bug (be the person who fixes the bug). You can also add your email to the list of those who are notified when a bug's status changes.

When you (or any other user) find a bug and want to file a bug report with Bugzilla, you need to follow a simple process:

1. If you do not have a Bugzilla account, you need to create one. Nothing special is needed, just an email address that is used to send you information about the bugs you are interested in.
2. Learn as much about the bug you found. Can it be repeated every time, or was it one-time only? If you have more than one computer, does the bug happen on both? Which steps are needed to reproduce the bug?
3. Next, search the Bugzilla database for your bug. If your experience in searching for bugs is limited, make several searches with different search terms.
4. If you find the bug in Bugzilla, that's great. That makes everything easier for you. Now, read the comments on the bug that matches the one you found. Is there anything you can contribute to the comments (perhaps better or different steps, different platforms, and so on)? If so, add a comment. If you can't add anything significant, do not add anything like "me too"—that type of comment is not productive.
5. If, after searching several times and not finding any bugs in Bugzilla that match your bug, then—and only then—should you consider filing a new bug report.

Let's start with looking up bugs.

Looking Up a Bug

To look up a bug, first go to the Bugzilla page at <https://bugzilla.mozilla.org/> (see Figure 17.1). This is the home page for Bugzilla and the starting point for your working with Bugzilla. Even though you must be a registered user to use Bugzilla, you can search and display bugs without being registered.

Searching for bugs can be done either on the main Bugzilla page (by bug number) or by using an advanced search with more options. The advanced search lets you restrict your search to a single product and find only open, closed, or all bugs.

The most advanced search allows very specific searches (see Figure 17.2).

After you have searched for your bug and have not found it, you can consider filing a new bug report.

Tip

Even experienced users are not successful on their first search sometimes. An experienced user realizes that perhaps he is using the wrong search term.



Log in, and then find bugs.

FIGURE 17.1

The links on the lower left let you search, generate reports, and log in to Bugzilla.

Filing a Bug

All bugs must have certain information. As well, you should do certain things to ensure that the bug is really in Firefox and not a profile, theme, or extension problem.

First, try uninstalling your extensions. You should uninstall one at a time and test after each one is removed. If the bug goes away after removing an extension, that extension is likely the problem and not Firefox.

Next, check to see whether there is something about the website that is unique. Does this bug affect more than one website? Is the website using plug-ins? Is the site trying to use JavaScript and you have JavaScript disabled? Is the site trying to display a pop-up window (for example, a web-based email system might do this to allow you to edit a new message)?

CAUTION

Do not use the Mozilla Bugzilla system to test and practice! Do not play with it—it is not a toy. If you want to practice (and I recommend you do), go to <http://landfill.bugzilla.org>. There you will find a Bugzilla configuration you can play with; file wacky bug reports at; and generally see what happens when a bug is entered, commented on, and other actions (such as closing) are taken.

You need to create a special Bugzilla account at landfill.bugzilla.org to use the test system. You can use the same account information as the Mozilla working Bugzilla system; there will be no conflict.

FIGURE 17.2

Bugzilla's advanced search enables you to find a specific bug by limiting the search.

Finally, you should create a new blank profile. Try to re-create the bug using this profile. If you cannot reproduce the bug with the new blank profile, your profile has likely been corrupted.

Without this information, it is virtually impossible to find and fix the bug. Start at Bugzilla's Enter Bug page.

Information needed in a good bug report includes

- The product name—for example, Firefox or Thunderbird.
- Which component is failing. The Enter Bug page provides a list, and you can read a full description for a component from that page.
- Which hardware platform the problem occurs on. Is it PC, Mac, all of them (you know it fails on all platforms), or some other platform?
- Operating system information. If you're using Windows, which version are you using? If the problem occurs on a Macintosh, which OS does it occur on? If it happens on Linux, which flavor of Linux is involved?
- The build number/version number of the product. This information can be found on the product's About page.
- For Firefox, if the bug is specific to a certain web page (perhaps the page doesn't display properly), the page's URL is needed.
- A one-sentence summary of the bug. Be objective and descriptive. Use keywords and proper terminology wherever possible.
- A detailed description of the bug. Be objective and descriptive. Saying it's a "terrible product" tells nothing. Saying "Opened page, and text on left overwrites images on the right" is an example of a good description.
- Whether the bug can be reproduced reliably. If you can make it fail every time, that is best. Bugs that fail without any indication of what caused the failure are difficult to find and fix.
- The steps to reproduce the problem. These should be detailed and leave nothing out. If possible, start with a clean launch of the product.
- The actual results your steps produced.
- The results you expected these steps to produce.
- Whether the bug presents a security problem. If this is true, there is special handling the bug might receive so the security flaw will not become public knowledge.
- How severe the bug is. Choices range from Blocker (nothing can be done until the bug is fixed) to Normal (a bug that should be fixed) to Enhancement (not really a bug, rather it is something that should be added).

After you have collected all this information about your bug, you can go to the Bugzilla enter a new bug page at https://bugzilla.mozilla.org/enter_bug.cgi.

For a bit of practice, go to <http://landfill.bugzilla.org/bugzilla-2.20-branch/> (see Figure 17.3). Select Enter a New Bug Report from the list on the left side of the page.

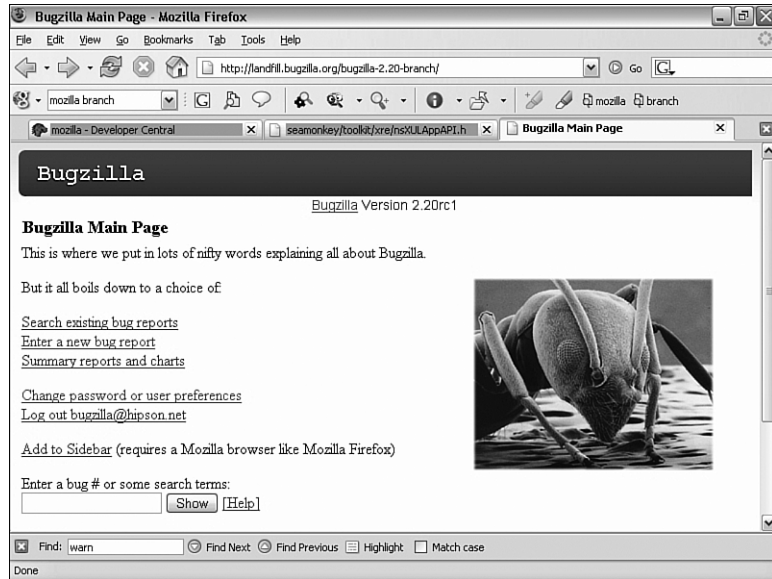


FIGURE 17.3.

After you're logged on to *landfill.bugzilla.org*, you can create a new test bug report.

The steps for creating new bug reports are as follows:

1. Select a product (see Figure 17.4).
2. Fill in the various pieces of information as shown in Figure 17.5.
3. When you're done, click Commit (the Commit button is just off the screen in Figure 17.5).

TIP Consider setting up an email account just for bugs. This can make managing your bugs easier because they will all be together, without unrelated clutter.

Each time you file a bug report with Bugzilla, you receive an email copy of the bug. This copy can serve as a reminder that you filed a bug.

Updating and Modifying a Bug

Any user can update her bug report, adding comments, information, and responses to other people's comments.

On the Bugzilla test system, you can safely practice updating your bug reports. Because you probably won't have the bug number handy, you can do a search. I'll search for a bug I filed on *landfill.bugzilla.org* on spider digestive goo and heartburn.

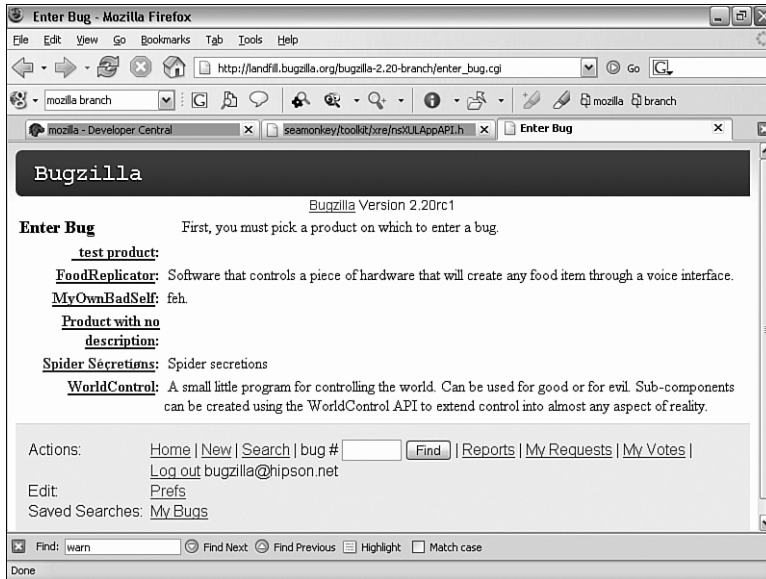


FIGURE 17.4

Each product is a link to the page where you can enter bugs for that product.

The bug is displayed using a form that shows all the bug fields (see Figure 17.6).

To tell Bugzilla that I will be the person who is fixing this bug, I click the Accept bug, which changes the status to Assigned. Then I click Commit. I can also choose to leave a reply to the bug's description (or reply to any other replies) by clicking the Reply link.

You also can leave additional comments on an open bug.

TIP Some fields, such as Reply, can be used by anyone who is logged in. Other fields might be restricted to either the creator or the Bugzilla administrators.

Fixing a Bug

Both creating and commenting on bugs are things anyone can do. Fixing bugs is a bit more complex, however. Virtually all bug repairs require that you be a programmer. Knowledge of C and C++, JavaScript, and the other technologies used in the Mozilla products is a must. But, should you have the skills and want to jump in with both feet, please do so.

Chapter 17

Mozilla's Tools for Developers

Enter Bug: Spider Sécrétions - Mozilla Firefox

File Edit View Go Bookmarks Tab Tools Help

http://landfill.bugzilla.org/bugzilla-2.20-branch/enter_bug.cgi?product=Spide

mozilla branch

mozilla - Developer Central

seamonkey/toolkit/src/nsXMLAppAPI.h

Enter Bug: Spider Sécrétions

Bugzilla

Bugzilla Version 2.20rc1

This page lets you enter a new bug into Bugzilla.

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug.

Reporter: bugzilla@hipson.net

Version: unspecified

Product: Spider Sécrétions

Component: Digestive Goo
Venom
Web

Platform: PC

OS: Windows XP

Priority: P2

Severity: critical

Target Milestone: --

Initial State: NEW

Assign To: gerv@mozilla.org

QA Contact:

Cc:

Estimated Hours: 100.0

Deadline: 2101-07-12 (YYYY-MM-DD)

URL: http://www.peterhipson.com

Summary: Digestive goo eats holes in non-synthetic fiber shirts

Description: Drooled and ended up with holes in my new shirt.

Steps to reproduce:

1. Eat
2. Be sloppy
3. Look at shirt afterwards

Got: had holes

Expected: just dirty

☐ Initial Description is Private

Keywords: (optional)

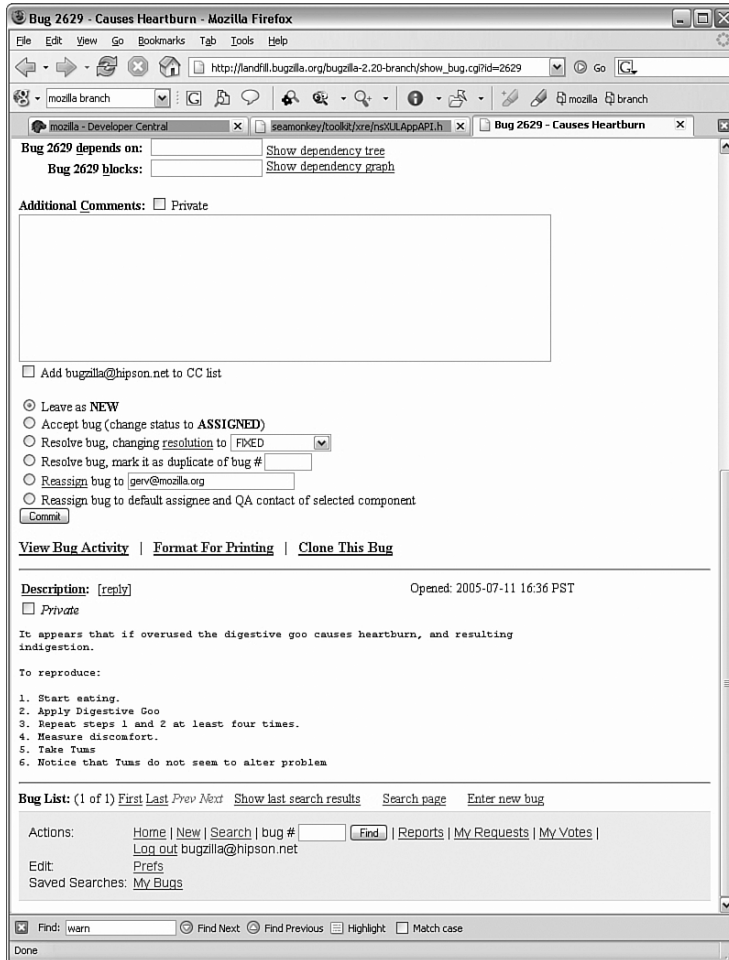
Depends on:

Find: warn Find Next Find Previous Highlight Match case

Done

FIGURE 17.5

Fill in the information requested. This information varies with products, so your bug reports might differ from this example.

**FIGURE 17.6**

I've scrolled down in this bug to the bottom part of the report, where I'll assign the bug to me.

In Chapter, 18, “Browsing the Code,” we will get into the actual work of fixing a bug. In this chapter, we deal with the administrative issues only.

After you've found a bug you think you might be able to fix, I personally recommend that you try this to get some experience. Start with a downloaded set of source files (real bug fixing won't be done with the downloaded source, but for practice this is not a problem). Then do the following:

1. Print the bug's report from Bugzilla. The bug report page has a link at the bottom labeled Format for Printing; it formats the bug report in a way that prints well.
2. Go to the product's source and try to fix the bug. This entails analyzing the code and the bug, determining a fix, coding the fix, and rebuilding the product to test your fix.
3. Did you really manage to fix the bug? If so, the bug might still be opened and not fixed. Regardless, you have gained very valuable experience in fixing and rebuilding the product.

If you follow these steps with a few real bugs and are able to successfully fix them, you can consider fixing bugs for real.

Tinderbox

Tinderbox is a build system tool for Mozilla/Firefox at Mozilla. With Tinderbox, you can see the following:

- **Who has checked out which module**—Many developers or programmers can be working on a product, both adding enhancements and fixing bugs.
- **The status of builds for various platforms**—Not all platforms are built or available at any given time.
- **Which platforms are currently broken**—That is, ones that cannot successfully be built.
- **The state of various component files and who did what to them**—This allows finding out who broke what.

Bonsai

Bonsai is an interface into the CVS system that allows queries using a web browser. You can choose a module, such as Bugzilla, and enter search criteria.

The main benefit of Bonsai is that you can determine who changed what and when the change was made.

A search returns records of updates to source such as those shown in Figure 17.7. When Bonsai lists search results, the columns returned are

- When the file was updated
- Who updated the file
- Which file (the fully qualified file name) was updated
- The file's new revision number (the old revision number incremented by one)
- The branch of the product/version
- +/-, which is the number of lines added and the number of lines removed with this change
- A description of what was changed, why, and any bug reference numbers

This is a Firefox bug.

Date	Author	File Name	Revision	Description
2009-07-08 14:59	darin@meier.net	mozilla/configure.in	1.148	
2009-07-08 14:47	mt@exedo.nl	mozilla/calendar/providers/memory/calMemoryCalendar.js	1.29	Fixes to memory provider's modifyItem. bug 299760, r=shaver
2009-07-08 14:28	mozilla.mano@scent.com	mozilla/browser/base/content/utilityOverlay.js	1.35	Bug 299713 - [Mac] If there's an open about window, activate it on Firefox->About Mozilla Firefox. r=mcconnor, a=bsmedberg
2009-07-08 14:28	mozilla.mano@scent.com	mozilla/browser/base/content/aboutDialog.xul	1.23	
2009-07-08 14:28	dbaron@dbaron.org	mozilla/layout/generic/nsTextFrame.cpp	1.828.2.1	Some work on AddInlineMinWidth.
2009-07-08 14:28	darin@meier.net	mozilla/modules/libmar/tool/MateFile.in	1.6	Remove -NODEFAULTLIB option to fix MSVC build bustage, bug 299557, r=s=chase
2009-07-08 14:28	bsmedberg@covad.net	mozilla/browser/installer/windows/packages-static	1.60	Bug 299573 - packaging error hides reporter overlays - r= darin, self-approving
2009-07-08 14:28	bsmedberg@covad.net	mozilla/browser/installer/unix/packages-static	1.47	
2009-07-08 14:28	dbaron@dbaron.org	mozilla/layout/generic/nsFrame.cpp	2.852.2.1	Rework white-space handling for inline intrinsic width calculation a bit.
2009-07-08 14:28	dbaron@dbaron.org	mozilla/layout/generic/nsFrame.h	2.282.2.55	

FIGURE 17.7

Using Firefox's Find helps you locate the information desired if a large number of records is returned.

Figure 17.7 shows a fix for bug 299713. This fix modified two files in Firefox: `utilityOverlay.js` and `aboutDialog.xul`.

Bonsai is useful when something is broken that worked in the past. Looking at Bonsai's output, you can gather clues to what might have happened.

LXR

The Linux Cross Reference (LXR) is a tool that is used to cross-reference various source modules that make up Firefox. LXR is hosted on a site external to Mozilla.

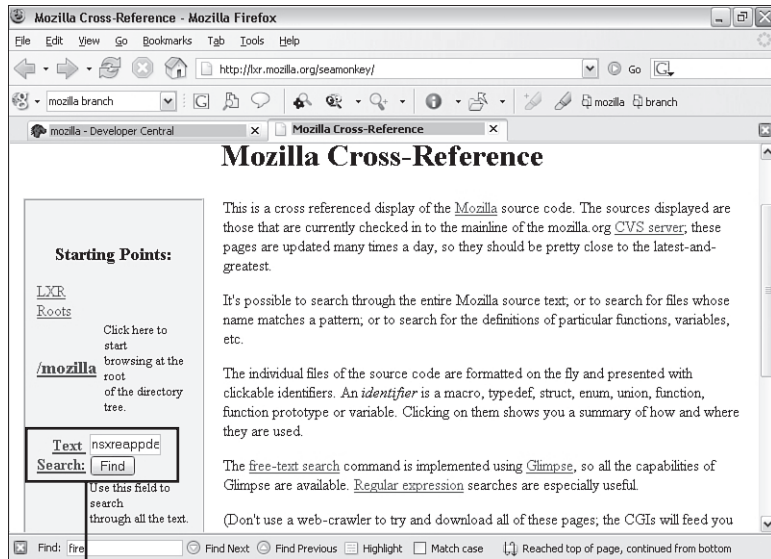
Tip You should visit sourceforge.net to learn more about its work and to see what software is available.

With LXR, and using a browser, you can access indexed and cross-referenced source code.

For more information about LXR, visit <http://sourceforge.net/projects/lxr>. LXR is available for a number of platforms and languages.

Using LXR is relatively painless. Because LXR is indexed, it is much faster than any local disk-based search you might run. You could run Find, a GREP utility, or the Windows Find Files and search for certain text, but the sheer size of the source makes this approach time-consuming.

Instead, go to Mozilla's Developer pages (<http://www.mozilla.org/developer/>) and click the LXR link (see Figure 17.8).



Enter text and click Find.

FIGURE 17.8

Although I'm searching for text, I could also search for specific filenames or identifiers.

You can do a simple search quite easily. First, establish the search's starting point (such as the Mozilla root, which is the default). Then in the Text Search window, type the text to search for—in this example, I searched for `nsxreappdata`. Finally, click Find.

LXR searches the code and displays a list of all the matching files (see Figure 17.9). For the term I searched for, LXR found 24 matches.

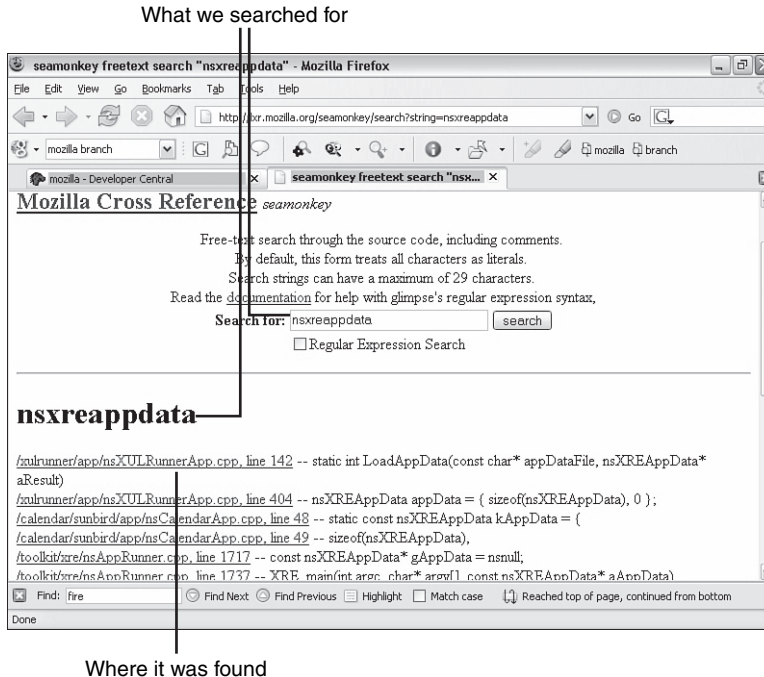


FIGURE 17.9

If you want to look at a file, just click the file's hyperlink in LXR.

Again, click the file you are interested in, and LXR displays the file formatted with hyperlinks on most identifiers (see Figure 17.10).

If you click the term you originally searched for, LXR displays the cross-reference (see Figure 17.11).

Chapter 17

Mozilla's Tools for Developers

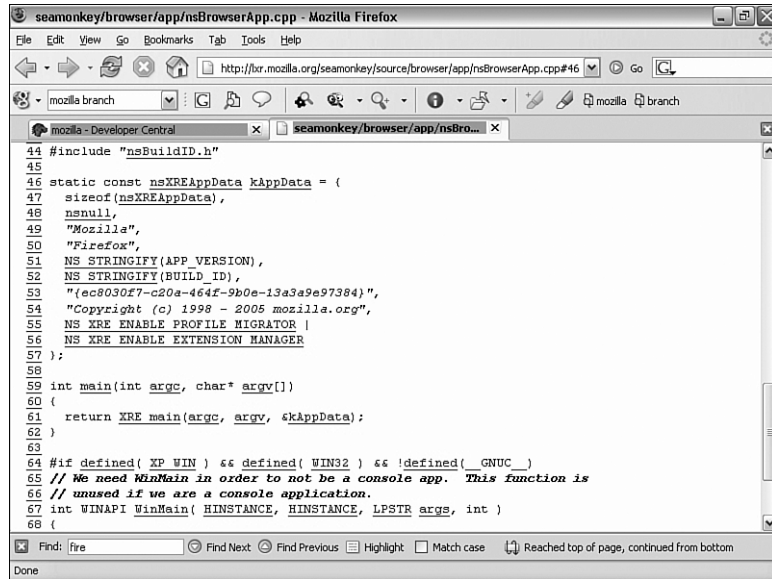


FIGURE 17.10

All the underlined terms in the source can be clicked to see cross-reference information for the item.

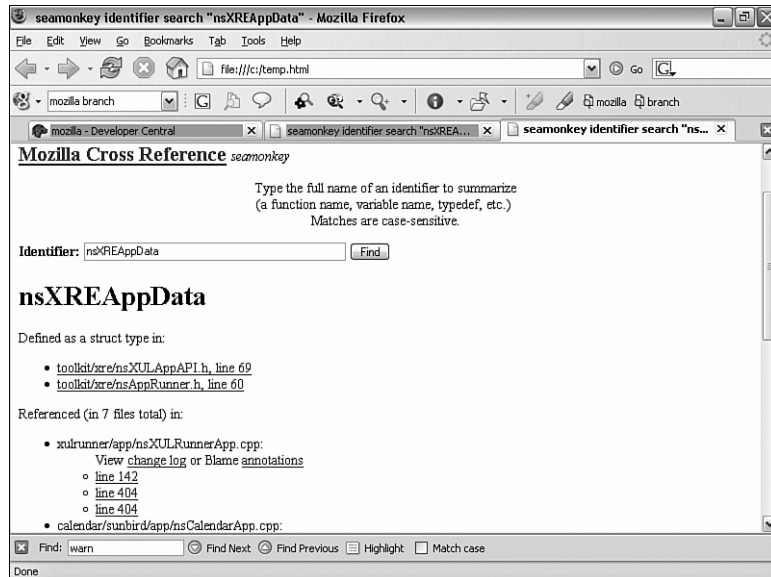


FIGURE 17.11

The cross-reference shows both where the identifier was defined and where it is used.

The capability to search the source quickly is a valuable tool, but the real value in LXR is the capability to cross-reference identifiers and other objects between all the files.

Working with Mozilla Development Tools Secrets for Power Users

Here are a few ideas from the experts:

- Open Source is here to stay. With Open Source, computer users have free access to applications, systems, and source code. Open Source fosters a community spirit between users and developers.
- Mozilla developers—both those adding new content and those who are fixing bugs—have a number of powerful tools to help them do their jobs.
- Bugzilla is a bug management program and tracking system developed at Mozilla. With Bugzilla, developers, testers, and users can search for suspected bugs, update bug reports, and (if they are technically able) make a bug fix.
- Tinderbox is a system that tells you the state of the current builds of the various Mozilla products. Tinderbox codes each build with a color: Green means good; yellow means compile/build in progress; orange means the compile/build was okay but failed a test; and red means the source would not compile or build.
- Bonsai is a tool that establishes accountability for various source changes. When something goes wrong, Bonsai can help tell you who last modified a given source file.
- LXR enables an online user to find references, cross-references, and definitions and do general searches on the various source files.

