FIREFOX HACKS™

Tips & Tools for Next-Generation Web Browsing

O'REILLY®

Nigel McFarlane

Internet Applications

FIREFOX HACKS Tips & Tools for Next-Generation Web Browsing

Firefox—the high-powered, feature-rich, open source web browser—is winning converts at an incredible pace, with tens of millions of loyal users already, and many more downloading it each day. Firefox empowers you to browse faster, more safely, and more efficiently than with any other browser. Imagine a world with no more spyware, vastly reduced spam, seamless pop-up blocking, tabbed browsing, built-in search, RSS integration, hassle-free downloading, and amazing customization possibilities! It's no surprise that this next-generation browser is rapidly winning over Internet Explorer's existing user base.

Firefox Hacks explains how to customize Firefox's deployment, appearance, features, and functionality—delivering techniques, tools, and strategies for making the most out of Firefox's outstanding flexibility. Learn how to:

- Make Firefox look different with skins, extensions, and themes
- Integrate Firefox with other tools, and safely migrate your Internet Explorer bookmarks, history, passwords, and other data
- Make yourself anonymous, turn off caching, stomp on cookies, or flush and clear absolutely everything
- Install, use, and customize extensions, widgets, toolbars and plug-ins—or create your own
- Increase security by stopping all secret network activity, fine-tuning ports and sockets, digitally signing content, dropping security blocks, and more
- Play with preferences, install fonts and character support, govern image and ad display, and otherwise tweak the standard browser window
- Power up your web pages with XML, using Mozilla SOAP, XML-RPC, and WSDL services, as well as MathML, SVG content, XPath, and XSL

Whether you're a crusty hacker, a bored web surfer, a contemporary web designer, or a frustrated IT professional, you'll find all the user-friendly tips, tools, and tricks you need to make a productive switch to Firefox. With *Firefox Hacks*, a superior and safer browsing experience is truly only pages away.

Got a Hack? Go to: *hacks.oreilly.com*

O'REILLY®
www.oreilly.com

US $24.95 **CAN $34.95**
ISBN: 978-0-596-00928-1

Safari BOOKS ONLINE ENABLED Includes FREE 45-Day Online Edition

FIREFOX HACKS™

Other resources from O'Reilly

Related titles

Don't Click on the Blue E!
Test Driving Linux
Cascading Style Sheets: The Definitive Guide
XML Hacks
Flash Out of the Box
Linux Desktop Hacks
Knoppix Hacks
Talk Is Cheap
Windows XP Hacks
Mac OS X Panther Hacks

Hacks Series Home

hacks.oreilly.com is a community site for developers and power users of all stripes. Readers learn from each other as they share their favorite tips and tools for Mac OS X, Linux, Google, Windows XP, and more.

oreilly.com

oreilly.com is more than a complete catalog of O'Reilly books. You'll also find links to news, events, articles, weblogs, sample chapters, and code examples.

oreillynet.com is the essential portal for developers interested in open and emerging technologies, including new platforms, programming languages, and operating systems.

Conferences

O'Reilly brings diverse innovators together to nurture the ideas that spark revolutionary industries. We specialize in documenting the latest tools and systems, translating the innovator's knowledge into useful skills for those in the trenches. Visit *conferences.oreilly.com* for our upcoming events.

Safari Bookshelf (*safari.oreilly.com*) is the premier online reference library for programmers and IT professionals. Search across thousands of electronic books simultaneously and zero in on the information you need in seconds. Read the books on your Bookshelf from cover to cover or simply flip to the page you need. You can even cut and paste code and download chapters for offline viewing. Try it today with a free trial.

FIREFOX HACKS™

Tips & Tools for Next-Generation Web Browsing

Nigel McFarlane

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

Firefox Hacks™
by Nigel McFarlane

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor:	Brian Sawyer	**Production Editor:**	Sanders Kleinfeld
Series Editor:	Rael Dornfest	**Cover Designer:**	Hanna Dyer
Executive Editor:	Dale Dougherty	**Interior Designer:**	David Futato

Printing History:

March 2005: First Edition.

This book uses RepKover,™ a durable and flexible lay-flat binding.

ISBN: 0-596-00928-3
[C] [5/05]

Contents

Credits

About the Author

Nigel McFarlane (*http://www.nigelmcfarlane.com*) is the Mozilla community's regular technical commentator, focused on education, analysis, and a few narrowly scoped bugs. He works full-time on Mozilla matters, except when interrupted by other things. The core Mozilla hackers are slowly getting used to him.

Nigel writes extensively on Mozilla, Firefox, and open source, including the columns *Searching for Substance* at InformIT and *Mozilla Meanderings* for TUX. He contributes to a number of journals, including *Inside Web Development* and *Linux Journal*, numerous online publications, and to the W3C's sXBL specification.

In addition to writing, Nigel messes around with programming and waves his arms about. He sometimes consults to industry and government. He holds science degrees from the University of Melbourne and La Trobe University. He is the author of several JavaScript Web books (WROX Press) and the authoritative *Rapid Application Development with Mozilla* (Prentice Hall)—the Mozilla community's favorite text in print. Other strong points are telecommunications, data systems, and the Web. On the human side, he spends a lot of time studying the forces and processes behind open source, invention, and online communities in general.

When not welded to the computer, Nigel enjoys Melbourne, Australia, his home city. It's often named the world's most livable city, and not without reason. Outdoors there are no computers, so bushwalking, swimming, music, and city rambles are great ways to relax.

Contributors

The following people contributed their hacks, writing, and inspiration to this book:

- John Allsopp and Maxine Sherrin are Western Civilisation Pty Ltd (*http://www.westciv.com*). They are the developers of the leading cross-platform CSS editor *Style Master* and a series of highly influential web development courses. Their web site is one of the longest-standing web developer resources for CSS and web standards. Maxine and John are co-conveners of the annual Web Essentials conference. They are both long-time web standards evangelists.
- Seth Dillingham is president and lead programmer of Macrobyte Resources (*http://www.macrobyte.net*). He and his company specialize in custom development of web applications, such as the popular Conversant (*http://www.free-conversant.com*) groupware and content management system. On those seemingly rare occasions that he steps out of his office, Seth can be found cycling around southern Connecticut or spending time with his lovely wife, Corinne.
- Ben Karel is surprised that you're bothering to read the contributor's bio section. Ben learned the Firefox ropes as a side effect of contributing an improbably large amount of time to the Adblock extension (*http://adblock.mozdev.org/*). He believes that open source software such as Firefox has a wholly unique potential to change the world. Entirely Too Much Information about Ben is available at *http://eschew.org*.
- Brian King has been involved in the Mozilla community since 1999. In that time, he has worked on both free and for-profit software built on top of the Mozilla Application Platform. He is one of the core site admins at *http://www.mozdev.org*, a community site for hosting Mozilla-based projects and for promoting Mozilla technologies. Brian is COO of Mozdev Group, Inc., a Mozilla services and software company.
- Tomáš Marek is a J2EE developer, Mozilla Links Newsletter translator (*http://newsletter.mozdev.org/intro/*), Mozilla evangelist, and an occasional contributor to open source web software. He lives in Beroun in the Czech Republic.
- Roger B. Sidje is mostly known in Mozilla and W3C circles for his work that enabled the display of mathematical documents on the Web through MathML, which is vital for educators, students and researchers. Born in Cameroon, Africa, he moved to France to undertake a PhD on the numerical solution of Markov chains. After that, he headed to

Australia for a position at the University of Queensland in the mathematics department. He works for the Advanced Computational Modelling Centre, where he develops other interests typical of academics.

- Neil Stansbury is the lead engineer for Redback Systems (*http://www.redbacksystems.com*). Living in London, England, he is the principal designer of the cross-platform Mozilla-based client for Novell's GroupWise product.
- Keith M. Swartz is an Enterprise Software Architect at Oracle Corporation, where he has worked for more than 11 years. He has provided support, development, and product management for the E-Business Suite Technology Group and has co-authored and contributed to several books on Oracle products. When not evangelizing the division's future technology directions, he is usually found nagging his senior management to switch to Mozilla and Firefox. He currently resides in Seattle, Washington, with his wonderful wife, Erin, and their cat, Gwen.
- James Vulling (*jrv@rockridge.net.au*), a software engineering graduate, consults in the architecture, development, and maintenance of enterprise scale web applications. The use of XSL in this environment has helped speed up the product development cycle immensely. Away from work, James enjoys home theatre, music and basketball.

Acknowledgments

My thanks to all the contributors to this book, who are a talented bunch and patiently nonjudgmental in the face of ridiculous deadlines. Thank you for your enthusiasm and support. Thanks also to Brian Sawyer, editor extraordinaire, for tolerating me and for deadly precision in email.

Preface

Welcome to *Firefox Hacks*, the book about the browser. Firefox is the web browser that upholds the highest principles of the World Wide Web. It supports and defines the Web as a good place to be—a place where all people can freely and safely participate, without having to tip their hat to anyone. Those are fighting words.

The Firefox product is made by the nonprofit Mozilla Foundation and its legion of helpers. Firefox is *open source* software: software that is free and fully exposed to independent scrutiny. Firefox is a poster-child of the open source trend and the one of the first open source products that average computer users try. If you find yourself liking Firefox, then open source has worked for you.

Firefox is more than just a TV screen for the Web, though. It contains extra technical goodies of all kinds. Whether you're a crusty hacker, a bored web surfer, a contemporary designer, or a frustrated IT professional, Firefox has features to offer you that will make your life easier and more ornamented. Hacking Firefox to meet your needs is what this book is all about.

Why Firefox Hacks?

The term *hacking* has a bad reputation in the press. They use it to refer to someone who breaks into systems or wreaks havoc with computers as their weapon. Among people who write code, though, the term *hack* refers to a "quick-and-dirty" solution to a problem, or a clever way to get something done. And the term *hacker* is taken very much as a compliment, referring to someone as being *creative*, having the technical chops to get things done. The Hacks series is an attempt to reclaim the word, document the good ways people are hacking, and pass the hacker ethic of creative participation on to the uninitiated. Seeing how others approach systems and problems is often the quickest way to learn about a new technology.

The Firefox web browser is built on the extensive and comprehensive engineering effort that is the Mozilla Project. Although the web browser's primary task is to display web pages, it is in fact a remarkably flexible, configurable, and complex environment. It is this broader view of Firefox that leads to many fruitful hacking opportunities. You need not accept Firefox exactly as it is without compromise. There is much you can do to tweak the browser to make working with the Web more pleasant, creative, and time-efficient. Feel free to hack Firefox.

How to Use This Book

You can read this book from cover to cover if you like, but each hack stands on its own, so feel free to browse and jump to the different sections that interest you most. If there's a prerequisite you need to know about, a cross-reference will guide you to the right hack.

Like most books, you can, if you want, start at the front and end at the back. If you know nothing about Firefox at all, the earliest material will ease you into the technology. If you're a hacker *extraordinaire*, then leap to the last few chapters where the hard stuff is. Overall, this book has something for everyone, so start with the chapter that's aimed at your particular needs first.

How This Book Is Organized

Firefox Hacks is divided into nine chapters, organized by subject:

Chapter 1, *Firefox Basics*
: This chapter covers beginner end-user features, without attempting to reproduce the Firefox Help system. It's a brief tour of the browser from a number of different user perspectives.

Chapter 2, *Security*
: This chapter covers the security arrangements that affect how Firefox interacts with the Web. It describes how to raise and lower security in a number of different ways.

Chapter 3, *Installation*
: This chapter explains how to install Firefox your way, rather than the standard way. It provides installation tactics suitable for a number of lifestyles and organizational settings.

Chapter 4, *Web Surfing Enhancements*
: This chapter explains how to make Firefox work harder for you as an information-gathering tool.

Chapter 5, *Power Tools for Web Developers*
: This chapter examines Firefox as a testing and development tool for web page design. Web developers love Firefox.

Chapter 6, *Power XML for Web Pages*
: This chapter is for more advanced web application projects, where programming or medium-weight XML technology is needed. It describes how to connect content to XML standards and code.

Chapter 7, *Hack the Chrome Ugly*
: This chapter looks under the hood of the Firefox installation and finds that with only a drop of programming energy, you can make major changes to the browser.

Chapter 8, *Hack the Chrome Cleanly*
: This chapter explains how to properly prepare fancy Firefox enhancements, such as extensions and themes.

Chapter 9, *Work More Closely with Firefox*
: Firefox is a complex tool and a building block of the Web. If you hope to modify that building block to suit yourself, then you'll need help getting started. This chapter describes how to connect to the core technology and the people already involved in its development.

Conventions Used in This Book

The following is a list of the typographical conventions used in this book:

Italics
: Used to indicate URLs, filenames, filename extensions, and directory/folder names. For example, a path in the filesystem will be written in the form */Developer/Applications*. Italics is also used to introduce new concepts and terms.

`Constant width`
: Used to indicate code examples, the contents of files, and console output, as well as the names of variables, commands, and other code excerpts.

`Constant width bold`
: Used to show user input in code and to highlight portions of code, typically new additions to old code.

`Constant width italic`
: Used in code examples and tables to show sample text to be replaced with your own values.

Color
: The second color is used to indicate a cross-reference within the text.

You should pay special attention to notes set apart from the text with the following icons:

This is a tip, suggestion, or general note. It contains useful supplementary information about the topic at hand.

This is a warning or note of caution, often indicating that your money or your privacy might be at risk.

The thermometer icons, found next to each hack, indicate the relative complexity of the hack:

 beginner  moderate  expert

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Firefox Hacks* by Nigel McFarlane. Copyright 2005 O'Reilly Media, Inc., 0-596-00928-3."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *http://safari.oreilly.com*.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). As a reader of this book, you can help us to improve future editions by sending us your feedback. Please let us know about any errors, inaccuracies, bugs, misleading or confusing statements, and typos that you find anywhere in this book.

Please also let us know what we can do to make this book more useful to you. We take your comments seriously and will try to incorporate reasonable suggestions into future editions. You can write to us at:

O'Reilly Media, Inc.
1005 Gravenstein Hwy N.
Sebastopol, CA 95472
(800) 998-9938 (in the U.S. or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

To ask technical questions or to comment on the book, send email to:

bookquestions@oreilly.com

The web site for *Firefox Hacks* lists examples, errata, and plans for future editions. You can find this page at:

http://www.oreilly.com/catalog/firefoxhks

For more information about this book and others, see the O'Reilly web site:

http://www.oreilly.com

Got a Hack?

To explore Hacks books online or to contribute a hack for future titles, visit:

http://hacks.oreilly.com

CHAPTER ONE

Firefox Basics

Hacks 1–10

Firefox is a software application for interacting with the Web. This chapter gets you up and running with Firefox. You probably know how to surf the Web already. This chapter is a sort of boot camp for the Firefox user interface. You'll learn where all the basic knobs and levers are. You'll also see many interesting nook and crannies. There are no significant programming tasks in this chapter; it's all just learning to fly.

Get Oriented

Before starting our journey, there are a few things you need to know: how to find Firefox files, how to install an extension, and how to set a preference. You don't need this information to read a web page, but it's vital for most of the hacks in this book.

Finding Firefox Files After Installation

After installation, Firefox puts its files in two important directories. The first is the *install area*, also known as the *application area*, which gets the general-purpose bits, such as the Firefox programs themselves. The second is the *profile area*, which gets user-specific information, such as bookmarks. Table 1-1 lists the default locations of these two areas on different operating systems.

Table 1-1. Default locations for Firefox installation and profiles

Operating system	Default install area	Default profile area
Single-user Windows 95/98/Me	*C:\Program Files\Mozilla Firefox*	*C:\Windows\Application Data\Mozilla\Firefox*
Multi-user Windows 95/98/Me	*C:\Program Files\Mozilla Firefox*	*C:\Windows\Profiles\%USERNAME%\Application data\Mozilla\Firefox*

Table 1-1. Default locations for Firefox installation and profiles (continued)

Operating system	Default install area	Default profile area
Windows NT 4.x	*C:\Winnt\Program Files\Mozilla Firefox*	*C:\Winnt\Profiles\%USERNAME%\Application Data\Mozilla\Firefox*
Windows 2000 and XP	*C:\Program Files\Mozilla Firefox*	*C:\Documents and Settings\%USERNAME%\Application Data\Mozilla\Firefox*
UNIX and Linux	*/usr/bin/firefox, /usr/lib/firefox* (or a custom location)	*~/.mozilla/firefox*
Mac OS X	*~/Desktop/Firefox.app/Contents/MacOS/* (if installed on the desktop)	*~/Library/Application Support/Firefox*

Feel free to have a (careful) look around in these directories.

Installing an Extension

Extensions are small enhancements to Firefox that are added optionally and separately to the main browser install. Just as plug-ins, such as Apple's QuickTime, enhance the display of web pages with special content, extensions enhance the display of Firefox's user interface with extra commands and extra windows. You install an extension when you need an extra feature.

You can install extensions from the official *http://update.mozilla.org* web site or from any web site that offers a *.xpi* file. Such files almost always hold one extension each.

If you choose to install an extension, you are acting like a power user rather than a beginner, so you have to use your judgment to decide what extensions will be useful to you. It's no use complaining that Firefox is too complicated if you have 10 or 20 extensions installed. Just remove them and resimplify your life.

Installing from update.mozilla.org. The Advanced Search Button extension (found in the Search Tools category at *http://update.mozilla.org*) is a harmless extension to experiment with. Here's how to get it.

After browsing to it on the update site, click on the matching install link. Sometimes there's a brief pause, and then Firefox presents an acknowledgement window. After another brief pause, Firefox enables the buttons in that window, asking you whether to proceed. In this case, go ahead.

At this point, most extensions are still *unsigned*, just as the dialog box reports. Although the update site has excellent trust credentials, the extension itself has no signature. Extensions have some credentials, though, because they've been through some checks before the update site makes them available. Ultimately, though, it's up to you to believe whether the update site and the extension provider are persons of goodwill. This is true for any download.

Once the user agrees, Firefox downloads and installs the extension. If you choose Tools → Extensions, you'll see it there in the list, but disabled and unusable (click on it and everything is grayed out). Shut down Firefox completely and start it up again. Again, look under Tools → Extensions. You'll see that the extension is now enabled and ready to use.

The sample extension, Advanced Search Button, doesn't change the Firefox user interface much at all. All it does is offer you a new toolbar icon. Choose View → Toolbars → Customize... to see the list of icons available. Scroll down and you'll see there's a new ASB icon there. Drag it to the navigation toolbar with the mouse. Finally, close the icon window. You can now click on the ASB button or the small triangle next to it. If you put a word in the search box and click the ASB button, the Advanced Search Button extension will search for that word in the currently displayed page.

Installing from elsewhere. By default, the Mozilla Update site (*update.mozilla.org*) is the only trusted web site for extensions. If you attempt to install an extension from another web site, Firefox will complain and ask you to provide extra permission. The complaint appears as a yellow information bar at the top of the page. Figure 1-1 shows this complaint for the version of the EditCSS extension that is available at *http://editcss.mozdev.org*.

If you click the Edit Options... button, a dialog box appears (shown in Figure 1-2), where you can add the web site to a whitelist of trusted sites. That has to be done at your discretion.

If the web site is added and the dialog box acknowledged, you're then sent back to the extension installation page. You have to click the extension's link a second time to restart the installation process, with trust now in place. From here, installation proceeds as explained for the Mozilla Update case.

Figure 1-1. Extension warning info bar for untrusted windows

Figure 1-2. Extension web site whitelist dialog box

Setting a Preference with about:config

A *preference* is a user choice that affects how Firefox runs. Firefox has hundreds of preferences. The Firefox Options Dialog box, under Tools → Options (Windows), Edit → Preferences (Unix/Linux), or Firefox → Preferences (Macintosh) is the way to set common preferences. In general, though,

the page shown by the special *about:config* URL can be used to set *any* preference. This process is described in detail in "Play With the Preference System" [Hack #23], but you need that information early. Here's a sneak peek.

All you really need to know is that if you right-click (Command-click on a Mac) in the list of preferences, you can change the preference under the mouse or add a new one. To change an existing preference, choose one of the following items from the context menu:

Modify
: Enter a new preference value.

Toggle
: Change `true` to `false` or `false` to `true`.

Reset
: Remove any change made to the standard value.

To add a new preference, choose the New context menu option and follow these guidelines:

- Choose the type of information the preference must hold. Use Integer for numbers and Boolean for preferences that are `true` or `false`. Otherwise, use String.
- Fill in the preference name and value in the small windows that appear. Type carefully, because there is no syntax checking.

After setting the preference, it appears in bold (unless you reset it). The *about:config* list is sorted alphabetically, so you might have to scroll a bit to see what you've added. Preference changes take effect immediately in most cases.

That's all the warm-up you need. Now, let's hack.

Ten Ways to Display a Web Page

Displaying a web page is like painting and hanging a picture: there are plenty of options.

Web pages that display only one way are called *printouts*. Firefox lets you display and view the content of a web page in a number of different ways. You also get to choose decorations for your windows. This hack shows 10 different ways you can display web content in Firefox.

Normal Browser Window

By default, web pages appear inside a frame with a menu bar and toolbars above and a status bar below. Those bars provide user control and feedback.

The whole browser window is a single XML document written in Mozilla's XUL dialect of XML. The web page appears inside an XUL `<iframe>`. Scripts running inside the web page can't reach out into XUL, so those bars are mostly untouchable. Most bars can be disabled from the View menu. Figure 1-3 shows a normal Firefox window with the sidebar made visible.

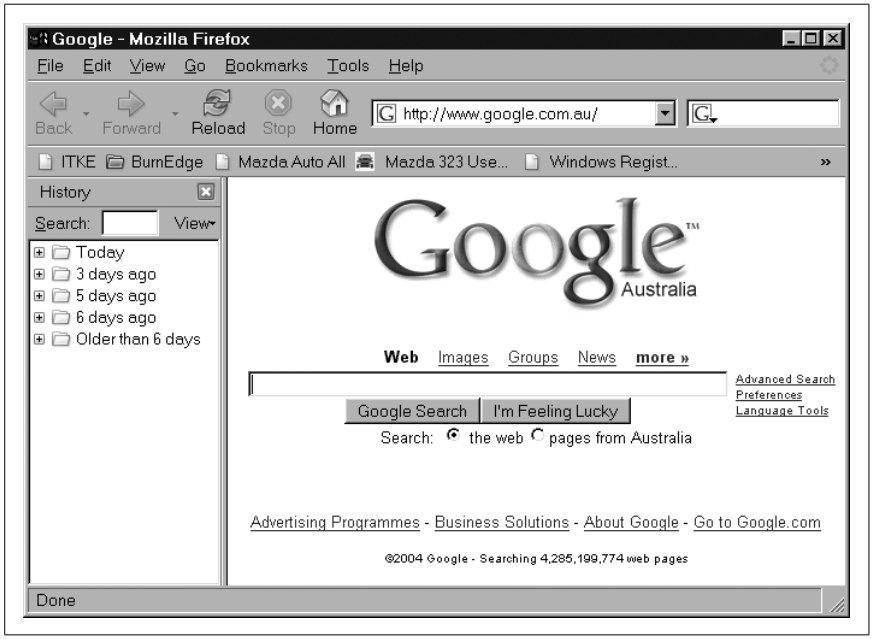

Figure 1-3. Firefox browser window with sidebar

Source Code Window

The source code window is for web developers and hackers only. Choose View → Page Source or press Ctrl-U to display the source code for the current page. If the page is a `<frameset>` page, you see only the frameset definition. You can change line-wrap and syntax-coloring options from the View menu in the new window.

window.open( )

Web pages can contain logic that opens other windows (usually pop ups). Web designers typically do so with the infamous `window.open( )` scripting feature. (Technically, this is a DOM 0 browser feature available from JavaScript only.) These new windows can be opened with specific menu bars and toolbars disabled. Security hobbles prevent some uses of this feature.

If you mitigate security restrictions, then everything is possible. See *Rapid Application Development with Mozilla* by Nigel McFarlane (Prentice Hall PTR) for a more detailed treatment of Mozilla and Firefox's `window.open( )` options. See *JavaScript: The Definitive Guide* by David Flanagan (O'Reilly) or any good scripting web site, such as Internet Related Technologies (*http://www.irt.org*), for the features of `window.open( )` that are supported by all browsers.

Full Screen or Kiosk Mode

You can expand a web page window to fill your entire screen.

Viewing the page in Full Screen mode won't help with readability. For that, you also need to increase the text size using the View → Text Size → Increase menu option.

You can take up all the screen real estate several different ways. The simplest is to click the operating system's Maximize button on the titlebar. This method, however, retains all toolbars and the titlebar. To get more space, press F11 (or choose View → Full Screen) for Full Screen mode, which removes nearly all bars. To get rid of the location toolbar as well, hide it using the View → Toolbars menu before pressing F11. Press F11 again to toggle back to normal. Figure 1-4 shows a web page viewed in Firefox's Full Screen mode.

For a robust kiosk mode (full screen with no other display options) that has most Firefox keyboard and mouse commands disabled, you need to *hack the chrome ugly*, as described in Chapter 7. You can't start kiosk mode from the command line. You can start it with a `window.open( )` call from a secure web page. The following feature string is suitable for a display that's 1024x768 pixels in size:

```
"chrome,modal,resizable=false,alwaysRaised=true,width=1024,height=768"
```

The alternative to `window.open( )` is intermediate scripting of Mozilla's XPCOM components using the chrome system.

Chrome-Free Windows

You can display an HTML or XHTML page free of any browser baggage. To do so, you need access to a command line, such as a DOS Box or `cmd` box under Windows, or a terminal emulator window (like `xterm`) under Unix/Linux.

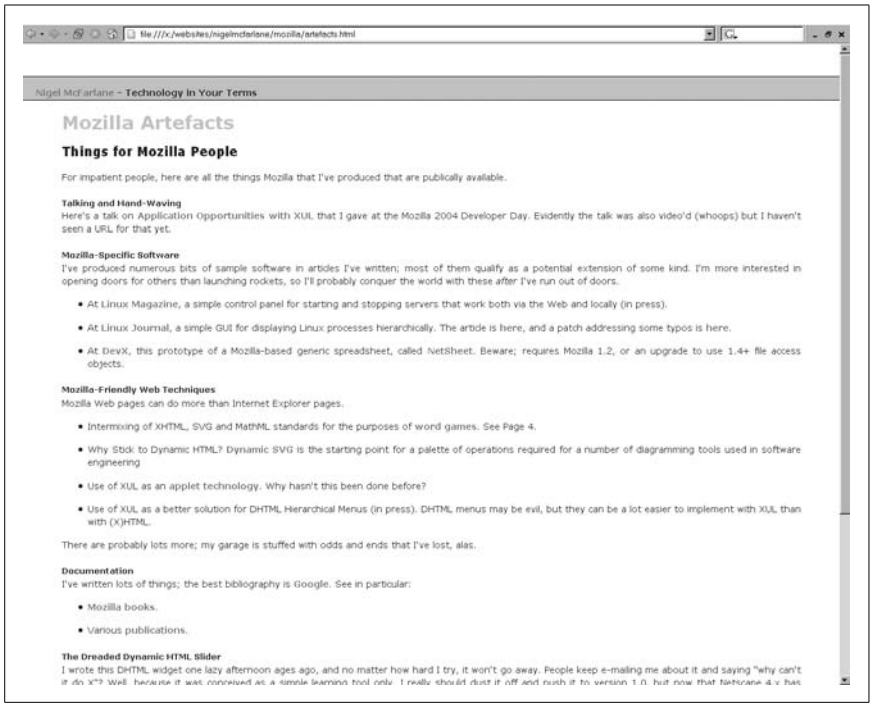

Figure 1-4. A browser window in Full Screen mode

On Windows, you need to follow these preparatory steps. Start a DOS Box from the Start menu, usually via Start → Programs → MS-DOS Prompt. Then move to the Firefox install area with these commands:

```
C:
cd "Program Files"
cd Firefox
```

Next, on all operating systems, just invoke Firefox with the required web page like this:

```
firefox -chrome URL
```

The equivalent `window.open( )` feature string is `"chrome"`.

Print Preview

Before printing your page, you can see what it will look like. Firefox prepares pages for printing using Cascading Style Sheets (CSS) styles. To see a preview, choose File → Print Preview and play with the toolbar. Firefox thinks briefly before displaying the preview.

User-Customized

Firefox support for CSS is designed to give the user control of page appearance. If you know CSS, you can exploit the following two files in Firefox:

userContent.css
: For styles that affect the appearance of CSS-styled pages that appear in a normal browser window. These styles work on HTML, XHTML, MathML, and CSS-styled generic XML documents.

userChrome.css
: For styles that affect the appearance of the Firefox chrome. That includes the toolbars, scrollbars, dialog boxes, menu bars, context menus, and other stuff that Firefox drapes around the displayed page. Since these styles are written using Mozilla's XUL tag set, *userChrome.css* provides styling for all Firefox XUL content.

For Microsoft Windows, there are some sample files called *userChrome-example.css* and *userContent-example.css* included in the Firefox install. They're buried deep inside the profile area. For all platforms, it's best to experiment with harmless styling changes (such as color changes) before proceeding to complex modifications of these files.

Once constructed, these files should be put in the *chrome* directory of your Firefox profile. Restart Firefox to see any and all effects. Don't forget to use the CSS imperative `!important` if you want to guarantee that your styles will override any others at work.

You can make small, generic modifications to suit all of your web browsing, or you can build a custom skin for a web site that you visit frequently. For the latter case, you need Firefox 1.1. You can then write web-site-specific CSS styles using this Firefox-specific syntax:

```
/* one page */
@-moz-document url(http://www.example.com) { color : red; }

/* all sub-pages with this URL as stem */
@-moz-document url-prefix(http://www.irt.org/articles/) { color : red; }

/* all pages at this domain */
@-moz-document domain(mozilla.org) { color : red; }
```

DOM Hierarchy

For HTML and all XML dialects supported by Mozilla, you can deconstruct a given web document into its pieces. These pieces appear as a sideways tree of tags and tag content. The resulting display is much like that of the Windows Explorer or the Macintosh Finder. Technically, the contents of the tree are W3C DOM nodes. That means tags and tag content.

To see this tree, you need to start the DOM Inspector by navigating to Tools → DOM Inspector. The DOM Inspector is an *extension* (an added feature) that comes as a bonus with the standard Firefox install. Initially, the DOM Inspector displays the web page of the window from which you started the Inspector. To view any other page, just type a URL into the top text field and click Inspect at the right. The new page appears below. In all cases, in the main pane on the left, you can drill down through the document, revealing its structure. "Probe HTML with the DOM Inspector" **[Hack #53]** and "Spy on Chrome with the DOM Inspector" **[Hack #76]** describe use of that tool.

Debug-Enabled

A web page can be displayed so that it is managed by a script-debugging tool. This is a feature designed for programmers. To use this display, install the JavaScript Debugger extension.

To add the JavaScript Debugger, go to Tools → Extensions. In the resulting dialog box, click Get More Extensions. That opens up a web page at the central Mozilla Update web site (*http://update.mozilla.org*). In that new web page window, you can shop for free extensions. Click the category Developer Tools, scroll down to JavaScript Debugger and click on the install link. Follow the bouncing prompts, and when that is all done, restart Firefox.

If you want the debugger to appear automatically when a given page loads, just put this tiny bit of JavaScript anywhere in the page content:

```
<script type="application/x-javascript">
 debugger;
</script>
```

Chapter 5 provides many techniques for debugging Firefox, including use of this debugger **[Hack #56]**.

Splash Screens

You can display a web page as a *splash screen*, but this is a harder effect to produce than the others and requires intermediate Web programming. A splash screen appears when a program first starts up and usually contains brand information. For example, the Mozilla Application Suite, which preceded Firefox, has a splash screen. Setup of a splash screen requires some hacking of the chrome files **[Chapter 7]** or of security arrangements **[Chapter 2]**, so explore those subjects first. It also requires scripting. To make such a splash screen appear, use this feature string in a `window.open( )` call from JavaScript:

```
"chrome,centerscreen=true,height=200,width=150,titlebar=false"
```

These options are available only if all security is lifted (e.g., from inside the chrome or from a signed and accepted URL). A similar hack for ordinary browser windows is to calculate window size from the `window.screen.x` and `window.screen.y` properties and then use a feature string like this:

```
"toolbar=no,location=no,personalbar=no,status=no,menubar=no,scrollbars=no,
top=225,left=300,height=200,width=150"
```

This alternative does not remove the titlebar or work around the Popup Manager. It is not guaranteed to remove the status bar either. Once you move into the world of Mozilla programming, many other page display options are possible, but that's a more complex matter than quick hacks. Splash screens provide more than enough complexity for an introductory chapter like this.

Ten Ways to Navigate to a Web Page

There are a million ways to move from the current web page to the next one. This hack describes 10 such ways.

Here's how to surf, Firefox style. Chapter 4 covers many ways to use extensions to increase the pleasure and convenience of surfing the Web. This hack explains what Firefox can do without modification.

Click on a Link

It's not rocket science: left-click on a link, and the current page is replaced with the link's page. If you right-click (Command-click on the Mac), you can open that new page in a new window or a new tab, or you can put the link itself into a bookmark or into the copy-and-paste buffer.

Click on a Bookmark Icon

Click any bookmark or bookmark menu item on the Bookmarks toolbar, and you're off to the link that bookmark represents. Bookmarks are complex little beasties, though. See "Find Stuff" **[Hack #3]** for a way to get through your many bookmarks. See "Use Fancy Bookmarks" **[Hack #33]** if you want to become a bookmark power user.

Put Something in the Location Bar

You can type all kinds of things into the Location bar where the URL of the current page is displayed. If you start typing a URL, Firefox will autocomplete it, providing you with a drop-down list of candidates. Firefox can

even complete an unknown URL for you if you press the right combination of Control, Shift, and Enter keys. Press Enter or click the Location bar icon to start the page fetch.

Pick Something from Your History

There are many history mechanisms beyond the big Forward and Back buttons. See "Find Stuff" [Hack #3] for an introduction to Firefox's history features.

Copy and Paste

The standard Xerox PARC/Apple Macintosh cut, copy, and paste keyboard combinations are available in Firefox. They are Ctrl-X, Ctrl-C, and Ctrl-V, respectively. You can copy a URL from a non-Firefox application and paste it into the Location bar. (You can give that bar's input field the focus by clicking on it.) You can also copy a link's URL by calling up the context menu with the mouse. You can copy and paste that URL within Firefox or into another application.

Under Linux, if you use a multislot copy-and-paste buffer tool, you can stick as many URLs into that set of clipboards as you want. To copy URLs from a terminal emulator window under Linux, such as `xterm` or `gnome-terminal`, just highlight the text. There's no need to explicitly choose Copy in this case. Paste into Firefox as you normally would.

Caret Browsing

For people with poor sight and for those that prefer a word-processing interface, a normal Firefox window can be made *caret-enabled*. The caret symbol is the blinking vertical bar that suggests an insertion point: |. Press F7 to turn it on or off. While it's on, you can navigate around the document using the arrow keys or with the assistance of a screen reader that relies on the caret. Just press Enter when the caret is over a link. You can't edit anything at the caret unless it moves into a text field.

Drag and Drop

Firefox allows you to drag URLs around the main window. You can drag bookmarks from the Bookmarks toolbar to the location bar for opening or drag links in a displayed HTML page to the Bookmark or Location toolbars.

You can also drag plain text to the Location bar. To do so, first display a web page that has a URL stated in plain text, not just embedded in a link. The text might literally say `http://www.yahoo.com`. Highlight the text by left-

clicking and dragging across it until it's all selected. Release the mouse button, and then drag the highlighted text onto the input field in the Location toolbar. The highlighted text appears in that text field when you release the mouse, ready to be retrieved from the Web. You can do this trick with non-URL text as well, but that makes sense only if you have keyword bookmarks set up [Hack #33].

Use the Menu Bar

Lots of web pages and web sites are hidden behind Firefox's menu system. The Go and Bookmarks menus provide immediate access to recent pages. The various File → Open options let you type in something useful. The Help menu options all provide locally stored web pages. The Tools menu leads to a set of dialog boxes and windows that let you interact with Mozilla's support sites for themes, extensions, and plug-ins. Finally, there are a few hidden Easter eggs to discover. Technically, they're not Easter eggs, because they're well documented diagnostics that are important for web developers and other technical people. Try typing these URLs, just for starters:

```
about:buildconfig
about:cache
about:plugins
```

Switch Between Windows and Tabs

When you already have many pages displayed, it's easy to toggle between them. You can cycle between whole windows with standard operating system features, such as Alt-Tab on Windows or panels under X11 on Unix. If you have activated tabbed browsing and several tabs appear in the current window, then just press Ctrl-Tab to cycle between them.

Use Accessibility Devices

Firefox's accessibility to disabled people is improving all the time. Part of accessibility depends on web pages being correctly designed. If the correct design is in place, then accessibility depends on the features of the browser. Firefox provides basic accessibility options via the Alt and Tab keys (on Microsoft Windows). Once Alt is pressed, it's possible to navigate the menu system using a simple and limited range of keystrokes (the arrow keys). Once Tab is pressed, it's possible to navigate all the input widgets on the browser and in the currently displayed page. Both provide such simple access to the menu system and web forms that the disabled surfer doesn't need to use a complex device like a mouse or even a keyboard.

HACK #3 Find Stuff

Search both the Web and the browser environment with Firefox.

This hack explains how to find the stuff you've forgotten, left behind, or yet to discover. It starts with local information in the browser and works its way out to searching the furthest corners of the Web.

Search a Displayed Web Page

You can search the content of a displayed web page. To do so, Firefox provides the Find toolbar. It appears at the bottom of the browser window, above the status bar. Figure 1-5 shows most of the features.

Figure 1-5. The Find toolbar and associated menu items

This toolbar is activated by the Edit → Find or Edit → Find Again menu options or by the Ctrl-F (Command-F on the Mac), Ctrl-G (Command-G on the Mac), or Shift-F3 hotkeys. Here's a rundown of how it works:

1. The Find toolbar starts out disabled but gains the focus and the caret in the toolbar text field when you click on it.

2. Type a string and Firefox automatically begins searching the current page, starting from the page's focused form field, or from the top of the page if no field is selected.
3. Firefox searches all frames but not all tabs. A security feature terminates searching if the page is too big to search in five seconds.
4. If nothing is found, the textbox goes red. "Phrase not found" appears, and a sound is issued.
5. If something is found, you can navigate between items with toolbar arrows or Ctrl-G (Command-G on Mac) and Shift-F3.
6. You can highlight found items with the Highlight toggle button.
7. Press Ctrl-F (Command-F on Mac) again and the current search string is selected for replacement. Dismiss the toolbar by clicking the small cross icon located at its right end.

The Find toolbar contains two searches in one. The default search is the one just described, called *Find As You Type*. It can also be activated with the forward slash (/) key, or if there's nothing focused on the current page and you just start typing. The other kind of search is called *Find Links As You Type*. It can be activated with the single quote key ('). This alternate search scans only hyperlink text on the page. Use it to find specific links on big pages crowded with links.

Search Web Page Source

You can also search the source code of a web page, something that was never possible in old Netscape browsers. View the page's source by choosing View → Page Source or with the Ctrl-U (Command-U) hotkey. A new window appears. That window has an Edit menu with the options Find (Ctrl-F or Command-F), Find Again (Ctrl-G or Command-G) and Go to Line (Ctrl-L or Command-L). Choose any of these and a small dialog box appears to control your search. Figure 1-6 shows most of the available features.

This searching system doesn't use a Find toolbar, but it's similar. It supports full content searching only and doesn't support link searching. It does, however, support case-sensitive searches.

Search Your History

There are several ways to search your surfing history. Figure 1-7 maps out these features.

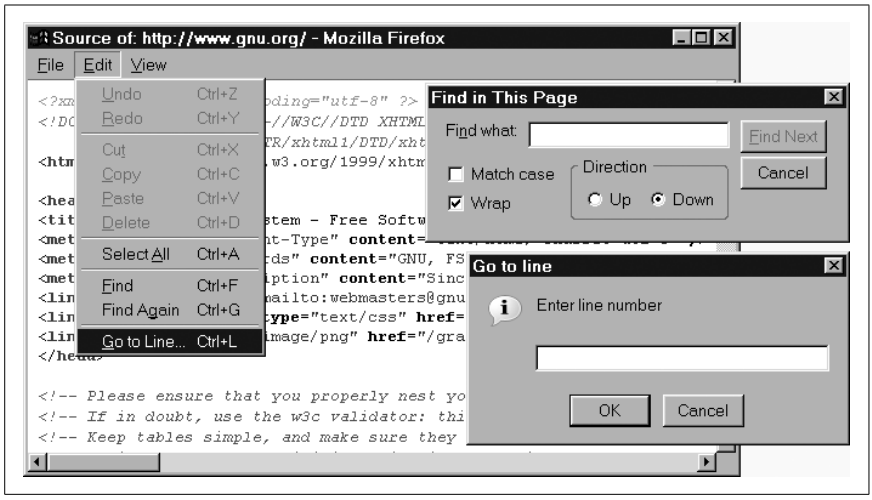

Figure 1-6. Find system for the View Source window

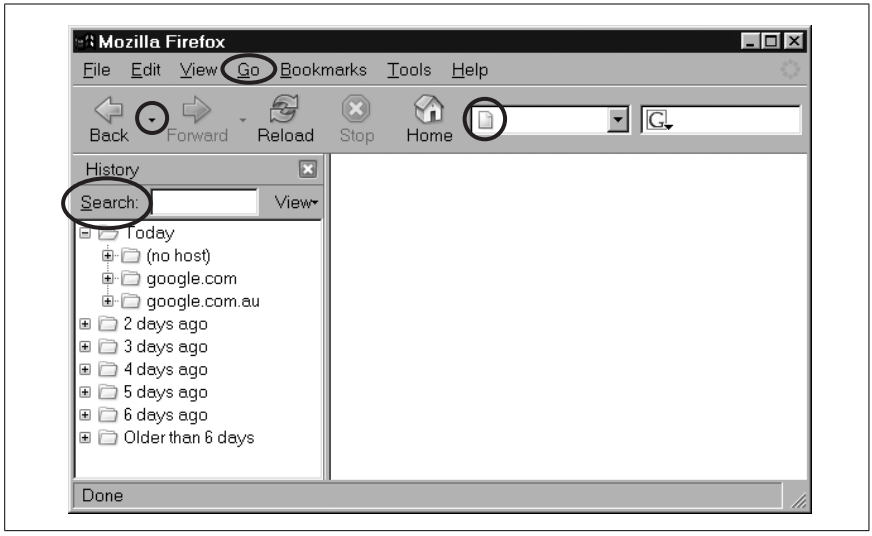

Figure 1-7. History searching features

The most accurate search uses the Back and Forward buttons. Press the small, black down-arrow grippies that sit to the right of these icons. A drop-down list of the *web trail* for the current window appears. If you ever surfed up a blind alley, then reversed, and went forward in a different direction, the set of blind alley pages is lost from the web trail. Full URLs, form data, and scrolled position are all otherwise preserved.

For another way to search, try the Go menu. It shows the most recently viewed URLs across all windows' *session history*. Only remote web pages are shown. Form posts and page scroll positions aren't preserved. As you view more pages, older items are pushed off this menu. To increase the size of the session history, modify this preference **[Hack #23]**:

```
browser.session_history.max_entries /* 50 is default */
```

The History sidebar, displayed with View → Sidebar → History (Ctrl-H or Command-H), provides the broadest search. It records your *global history* (all the web pages examined in the last nine days). It searches the *titles* of those pages only—no title content, no match. Delete the search string to see the day-wise hierarchy of pages again. Hack these preferences to change the defaults:

```
browser.history_expire_days /* 9 is the default */
browser.history.grouping    /* "none", "day" or "site" */
```

Finally, the Location bar can be used to search recently visited web sites in your global history. Type any string into the location bar and suggested *domain name* matches automatically appear in a drop-down list below the text box. This is called *auto-complete*. You can't search for path-specific bits of a URL until you've got the whole domain name (e.g., *http://www.yahoo.com*) right.

Search Your Bookmarks

Are you overflowing with bookmarks and can't remember any of them? Open the Bookmark Manager using Bookmarks → Manage Bookmarks... and type something into the search window. This search examines the full URLs of all bookmarks and their descriptions. The descriptions usually match web page titles, unless you changed the titles when storing the bookmarks. You can also sort the Bookmark Manager's view of your bookmarks from the View menu, which might be helpful. It's almost a *depth-first tree sort*, though, which is a sophisticated sort. It takes a bit of getting used to. Try it.

Search for Lost Logins

Typed your username and password into a web site and then forgot what they were? Firefox can help. Choose Tools → Options to bring up the Options dialog box. On Unix/Linux, it's under Edit → Preferences. On Macintosh it's under Options in the Application menu. Click the Privacy icon. Note the Saved Form Information and Saved Passwords items. Click each one's *expando* (the small plus icon) in turn. Turn on the various options by clicking their corresponding checkboxes, as shown in Figure 1-8.

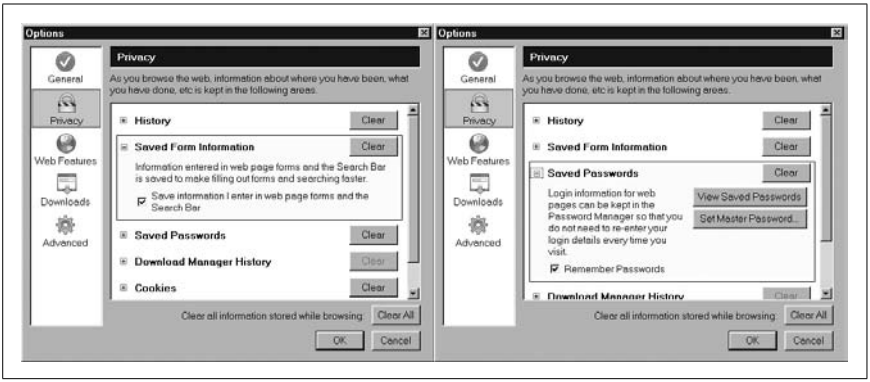

Figure 1-8. Options preferences for form and password tracking

Visit a web page with a form containing at least one text box, fill it in, and submit. Google's home page (*http://www.google.com*) is a good choice. Open a new Firefox window. Visit the page again. Left-click on the textbox twice. A drop-down menu appears, showing you the value you just typed. Pick the one you want.

Alternatively, visit a web page requiring a login. I log in at *http://bugzilla.mozilla.org* a lot (follow the login link). Use Firefox's form completion to enter your username, and Password Manager automatically adds the password. Just click the Login button and you're in.

Later on, you can review what you've saved. Back in the Options dialog box, under Privacy → Saved Passwords, click the View Saved Passwords button. In the resulting dialog box, click Show Passwords and Yes. Your passwords appear in plain text as an additional table column. Found them!

Search the Web

The Navigation toolbar includes a Search field on the extreme right. It uses the Google search engine by default. Type a keyword into the field and press Enter. Google replaces the current page with its search results. Alternatively, click on the small Google icon to see the other search engines installed by default; you can use any of them as well. Empty the Search field again and type a common letter, such as *a*. The Search box drops down a list of keywords starting with *a* that you've typed before.

Search for Firefox Preferences

Firefox has lots of preferences. Type in the URL `about:config` to see many (but not all) of them. There are too many to understand all at once. Type

anything into the textbox labeled `Filter:` to restrict the list to the ones you care about (try typing `go`). The set of preferences is marginally different between Linux and Windows, but 99 percent of Firefox preferences apply to all platforms.

HACK #4 Identify and Use Toolbar Icons

This hack explains the mystery icons that sometimes appear on Firefox toolbars.

It's up to you whether you take advice or not. Firefox delivers a number of passive warnings and cautions to you in the various toolbars of the browser window. Here's a rundown of what those things mean.

If you choose one of the many themes offered by Firefox's Theme Manager, then all the icons will likely appear different. Their locations will be the same, though.

This hack describes only the Firefox-specific icons that are different from standard browser icons:

This is the standard icon for a Mozilla Extension, Plugin, or Theme. Extensions are small add-on pieces of logic (or whole applications) that can be run or used as part of general web activity. If you have installed one or more extensions into Firefox, then something is going to work differently from the default behavior.

This icon sometimes appears at the bottom-right edge of the status bar. It tells you that the web page you're looking at has an RSS feed that complements its normal HTML content. You can hover your cursor over the icon to see the status of the feed. Click on the icon to capture the feed as a set of Firefox Live Bookmarks **[Hack #33]**. See *http://www.mozillazine.org* for an example.

This icon sometimes appears at the bottom-right edge of the status bar. It tells you that the web page you're looking at has alternate stylesheets. You can apply any of the stylesheets provided by clicking on the icon.

This icon sometimes appears at the bottom-right edge of the status bar. It tells you that the web page you're looking at has blocked a pop up. Click on the icon to see details of the page blocked, or to alter pop-up-blocking settings.

One of these three icons sometimes appears in the top-right corner of the menu bar. They indicate that there is an update (a patch or new release) for Firefox or for one of the extensions or themes that you've installed. Beware that issues with color and accessibility might mean these icons have been changed by the time you read this. Each icon indicates a different severity. Red means that a security problem with your browser has been detected at the moment, and you should grab the recommended fix by clicking on the icon. The other severities don't require any action, but it's good practice to keep your browser fully up to date. That makes these icons go away.

A subset of these icons appears in the Mozilla Extension Manager and Theme Manager dialog boxes. From left to right, they stand for "uninstall from the local disk," "update from the Web," and "configure options for a given extension or theme." Hover over the icons with the mouse to remind yourself what they're for.

These icons sometimes appear on the Find toolbar at the bottom of the browser window. The first one indicates that your search has been unsuccessful. The second indicates that the search has just wrapped, which means you've just reached the bottom (or the top) of the web page and the find process will continue again from the other end.

These icons sometimes appear at the right end of the Location bar, where the current page's URL is shown. They can also appear at the right edge of the status bar, at the bottom of the window. The uncrossed icon appears when the web page you are viewing is served up securely. That usually means that Secure HTTP (`https:`) is providing the page. Secure HTTP uses Secure Socket Layer (SSL) technology to ensure that all information between the web server and you is safely encrypted. The other icon appears when you log into a web site using a password, but the password is transmitted unencrypted to the web site. The icon indicates that your login is not as free from prying eyes (network sniffers in particular) as it might otherwise be.

HACK #5 Use Keyboard Shortcuts

Driving Firefox from the keyboard is both the same as and different from other browsers.

This hack shows you which keyboard moves come standard with Firefox. There are many extensions that modify the available set of keystrokes and key chords. You can also hack the keyboard set yourself in a number of ways **[Hack #77]**.