

This is a **low resolution, black and white** version of the article you downloaded. To download the whole of Free Software Magazine in *high* resolution and color, please subscribe!

Subscriptions are free, and every subscriber receives our fantastic weekly newsletters — which are in fact fully edited articles about free software.

Please click here to subscribe:

<http://www.freesoftwaremagazine.com/subscribe>

# Browsers for Mac OS X

## Comparing FOSS browsers for Mac OS X

Martin “MC” Brown

W

hen Apple migrated the Mac operating system platform to Mac OS X, one of the key components was an underpinning based on the FreeBSD operating system.

The use of an open source operating system as the core has in turn led to an increase in the use and availability of free and open source software (FOSS). It is now much easier to develop software for the OS X platform (development software is included, instead of being an expensive addition) and this makes it both easier for people to get involved and more likely to take part in open source community projects.

Despite the ease, there are still some areas of software development where the complexities of the application are too great. Web browsers are a classic example of this—although the principles of web browsing are quite simple, making all of the components of a typical web page (JavaScript, different image formats, plug-ins, CSS etc.) work effectively is quite difficult. Hence the need for community-based projects where many people can pool their experience.

As a knock on effect, web browsing in the free software space is now based around two main camps, those based on the Mozilla codebase (Camino, Firefox, Mozilla) and those based on the KDE sourced KHTML rendering engine (Safari and OmniWeb). Projects based on the former are completely free software, with both the rendering engine and interface being a free software project. The latter are not strictly free software, but the core rendering engine on which they rely is. It is only the interface which is not free software.

For the review, I’m going to be taking a closer look at the features and functionality of the three free software browsers—Mozilla, Firefox and Camino, and what differentiates the functionality of these three browsers from each other. I’ll primarily be comparing speed and functionality, but I’ll also use the opportunity to cover some of the issues that affect the browsers on individual sites. I’ll then take a very brief look at the quasi-free software browsers; Safari and OmniWeb.

Web browsing in the free software space is now based around two main camps, those based on the Mozilla codebase (Camino, Firefox, Mozilla) and those based on the KDE sourced KHTML rendering engine (Safari and OmniWeb)

### Mozilla Suite

The Mozilla project was borne out of the Netscape Navigator project; Mozilla being a combination of the term Mo-saic killer (which is what they hoped Netscape would be) and a reference to Godzilla. Mozilla was also the name of the Netscape mascot—appropriately enough usually represented by a humanoid-lizard, like Godzilla. The Mozilla Organization was created in 1998 to create the next generation of Netscape and was registered as a not-for-profit organization (the Mozilla Foundation) in 2003. Today, Mozilla is the



name used to refer to the range of browsers produced by the Mozilla group, notably the main Mozilla Suite and the forks of Firefox and Camino.

The Mozilla Suite is the name given to the software package that incorporates the functionality of a browser, email, newsgroup reader and HTML editor. The core of the Mozilla suite is the Gecko rendering engine which is also shared by Firefox and Camino. The Gecko engine is based on the HTML/XHTML, DHTML and other standards that make up the web environment and is specifically designed to render these in a consistent fashion across platforms. This means that you browse a page on Mac OS X and Windows within a Mozilla browser and get a consistent view. It also means that, in theory, different browsers that use the engine will have a consistent view. This isn't quite the case for reasons that will become clearer as I look at the other browsers from the Mozilla stable.

For example, let's look at the homepage for MCslp.com (<http://www.mcslp.com>), my website, which is based on the WordPress blogging engine with a theme which is XHTML compliant. You can see how this looks in Figure 1.

As the larger “internet suite”, the Mozilla browser is more geared toward users who want a consistent interface and environment for all of the browsing and internet needs. The interface, as shown in the figure, is based around the original Netscape browser. Shown is the Classic theme but a standard installation also includes the Modern theme which

is a little more aesthetically pleasing. This retains the “toolbar” style of the original—including “handles” for moving the different toolbars around and changing their order and layout. You can also click on these tabs to “collapse” the tool bar so that it takes up less space. What you don't get in the standard format is a convenient search bar for Google or other sources; instead you get a button that takes you to a predetermined Search button and the location field doubles up as the source for search criteria. What you cannot do is customize the toolbar you can enable and disable buttons, but not actually modify or add to the suite of buttons.

Like Firefox, Mozilla supports both themes—which change the appearance of buttons and other aspects of the interface—and extensions, which provide additional functionality. For example there are extensions that provide Ad filtering, download management or additional information during browsing. For example, the LiveHTTPHeader extensions can display the header information sent during an HTTP transaction—invaluable during development. Unlike Firefox there is no convenient extensions manager, and in fact Mozilla almost hides the existence of extension support from the application.

In terms of simple internet browsing Mozilla is more than capable. It does sometimes feel slower than the alternatives and there is often a discernable pause just before the page is displayed that can become a little frustrating. For simple text sites it isn't a problem, but those with heavy graphics, or complex tabular pages do demonstrate the problem.

In terms of the quality of the display and parsing, I experience very few problems with Mozilla when laying out different pages and sites. The only times you experience any specific difficulty is either when the site is not standards compliant, in which case some interesting “default” choices are made about how to layout components, or where the site has deliberately used Internet Explorer capabilities to provide functionality. For example, some menus and items that rely on JavaScript or ActiveX simply don't work. I've also experienced some occasional issues when logging into sites through a simple password system. For example, my blog at Computerworld never loads properly after login. Again, I suspect this is more of an issue of the site code, rather than Mozilla, but it can be a frustrating experience.

Being a suite of internet applications, rather than just a bare browser, Mozilla has the benefit of including most of the tools you will need. I also love the way that you

can read/reply to email directly from within Mozilla. This comes in particularly handy if you are frequently using sites that use email links; there is none of the delay sometimes experience while the OS works out what to do with the link. The embedded IRC client is also very handy. However, once you start using all of the components at the same time, you can sometimes experience some performance problems. Also, as a single monolithic application the memory usage can be prohibitive. Although memory management in OS X is efficient enough that you don't often experience problems, the memory issue is almost certainly related to some of the performance issues I experienced, especially while monitoring the load and memory usage of Mozilla while browsing various pages with all of the options enabled.

I liked Mozilla, but I have to admit to preferring other applications for my Email and chat requirements, and this is where Mozilla, as the "do all" application, starts to look like overkill if all you want to do is use the browser. A good example here of the limitations of the built in system is the address book. Within Mozilla you have your own special address book section, but this doesn't integrate with the OS X address book and that means you can't share instant messaging details with iChat and there is no provision for SMS messaging, an option built into the standard OS X Address Book application.

Where Mozilla is most likely to be used is within an organization where they want to standardize on a single platform for browsing and email access, and are probably supporting an LDAP server for address book functionality. In fact, I deployed Netscape Communicator, which provides similar functionality, in this way several years ago.

## Firefox

If you removed the web browsing component from Mozilla and turned it into a separate application, you would essentially get the Firefox application. It is a bare-knuckle, browser only, application that combines the Gecko display engine on top of the Mozilla application base to produce a light-weight internet browser. Because it's built on the Mozilla core you can add extensions (just as with Mozilla, although with a convenient extensions manager) and adjust the theme and display of the browser.

Because Firefox is just a browser it is a direct competitor to the standard Safari browser supplied with OS X and as

Mozilla Suite	
Name	Mozilla Suite
Maintainer(s)	Mozilla
License	Mozilla Public License (OSI Approved)
Platforms	Linux, Unix, Mac OS X, Windows
<b>MARKS (out of 10)</b>	
Installation	10
Vitality	10
Stability	9
Usability	9
Features	10

a cross platform solution has obviously gained some popularity as a more secure alternative to Internet Explorer on Windows. Since Microsoft have dropped support for Internet Explorer support under OS X, it is often the main choice if the user is unhappy with Safari.

Fortunately, Firefox has a lot of additional benefits that make it a popular alternative. The support for extensions enables the use of ad blocking and enhanced security functionality that Safari does not support. Many also find Firefox faster than Safari, although I have to admit to finding so little difference between the two that this wouldn't be my reason for choosing Firefox. The ad blocking extensions though are invaluable, and a combination of Adblock and the filter updater (which automatically reloads new filter sets for you) can eliminate nearly all ads to the point where only interstitials remain. As with Mozilla, the rendering of Firefox pages is generally excellent, all because of the Gecko engine. The same, occasional, site-specific problems occur. Unlike Mozilla, Firefox is more flexible in its interface. Toolbars are customizable, and, as can be seen in Figure 2, you get a convenient search field that can be used to search a variety of locations simply by selecting an alternate engine from the embedded popup. Tabs are also slightly better supported with switching between tabs noticeably faster (for me at least), and with convenient hot keys (Command-1, Command-2, etc.) for flipping between multiple tabs in a window.

Firefox is undoubtedly a very fast browser and is very quick

MCslp.com in Firefox



Firefox

Name	Firefox
Maintainer(s)	Mozilla
License	Mozilla Public License (OSI Approved)
Platforms	Linux, Unix, Mac OS X, Windows
<b>MARKS (out of 10)</b>	
Installation	10
Vitality	10
Stability	9
Usability	10
Features	10

and easy to use. With the extensions, particularly ad blocking, Firefox becomes an even faster alternative to the full Mozilla Suite product. You may not get the embedded mail/news reader, address book and IRC, but you get the flexibility to use other clients that may contain richer features (I find Apple's Mail best for Email), and the reduced memory footprint is certainly an improvement if you have 512MB of RAM or less. If free software mail/news tools are important then Thunderbird, also from Mozilla, is their standalone email application.

## Camino

The Camino project is based on the same rendering engine as both Mozilla and Firefox, but unlike these two packages, the Camino interface is designed to be more Mac OS X like, rather than the architecture neutral interface offered by its brethren. The effect provides all of the power and speed of the Firefox browser—Camino is just a browser—but with the OS X look and feel.

The architecture neutral format of both Mozilla and Firefox is obviously a benefit in a cross platform environment, but some prefer the look and feel of OS X. Most of the basic look and feel remains the same; key differences when browsing are the use of the OS X style popups, fields and buttons in the display, as you can see here in Figure 3. The most significant differences are in the dialog boxes. For ex-

ample, the preference panel looks and works almost identically to the main OS X System Preferences application.

The other main aim of the Camino project is to produce a very fast browser. Based on the same Gecko engine the Camino project includes some improvements and uses the functionality of the OS X display engine to help improve the overall speed of displaying web pages. The result is quick—pages display very quickly, even the complex ones. There are some problems, well documented on the site, that can cause some interesting effects, but as a product still considered young and in pre-release form (the latest at time of writing was 0.8.4), Camino is still impressive.

Although Camino includes a lot of integration with OS X—for example the bookmarks integrate with Rendezvous and the standard Address Book to include sites described in these repositories—Camino is not a typical Mozilla project. The extensions and themes supported by both Mozilla and Firefox are not supported. Although the themes limitation makes sense (since Camino is essentially permanently in the “OS X” theme) the lack of extension support means that we cannot block ads. Ironically, this often has the effect of severely slowing down the browsing experience. Most of the extensions are not supported because they rely on the XUL interface for configuration, which is not included in the Camino project.

Although Camino is very quick and is the most OS X-friendly of the Mozilla applications, the occasional display problems can be a pain. Also, the lack of extension support,

MCslp.com in Camino



Camino

Name	Camino
Maintainer(s)	Mozilla
License	Mozilla Public License (OSI Approved)
Platforms	Linux, Unix, Mac OS X, Windows
<b>MARKS (out of 10)</b>	
Installation	10
Vitality	10
Stability	9
Usability	10
Features	8

which eliminates the ability to filter adverts, is a feature that Camino could certainly take advantage of. If you browse an intranet or other controlled sites where adverts are not a problem then Camino is probably the best of the bunch, but if you need ad blocking you will need to consider Firefox.

## Safari and OmniWeb

When Apple launched Safari the OS X community was rightly amazed. It included a number of features that in today's browsers we take for granted. Easy access to Google, for example, was functionality we didn't have in Internet Explorer, and Firefox was not available at the time. Snap-back functionality (which takes you back to a Google, or indeed any other, page after clicking through other choices) makes it easy to browse sites without losing your root. Finally the integration with the operating system is absolute—in the latest versions, you can even synchronize your bookmarks between Macs using the Apple .Mac service.

Most interesting of all, Safari uses the KHTML rendering engine. KHTML is part of the Konqueror browser and the KDE Project, a desktop environment commonly used under Linux. The KHTML engine is incorporated into what Apple call the WebKit core, which provides HTML rendering for a variety of systems within OS X, including Safari, Sherlock and the built-in help system. The KHTML engine is a mature product and the rendering of HTML pages is based on the same open standards for HTML that drive the

Mozilla project. There are some circumstances, just as with Mozilla, where you get odd rendering issues but they are few and far between and again very site specific. The use of free software is perhaps not that surprising—Mac OS X is itself built on top of the free software Darwin project which is in turn based on the FreeBSD operating system.

Unfortunately, although KHTML is a free software project, Safari is not and therefore cannot strictly be included in this roundup review. In terms of functionality, Safari is probably closest to the Camino project, it supports browsing tabs and integrates well with the rest of OS X, but it does not do ad blocking as standard. There are however some products that support this (PithHelmet for example).

Also based on the KHTML engine is the OmniWeb browser from OmniGroup Software. OmniWeb is a commercial product and is one of the richest of the browsers available for the platform. It supports a number of features designed to make it easier for people who do a lot of browsing to organize their environment. For example, as well as supporting text tabs, OmniWeb also supports icon tabs, which show a live thumbnail of the tabbed page. You can also create multiple workspaces—these are groups of windows (including all the tabs in each window) and the information in a workspace is retained, even across application restarts. This means that you can create individual workspaces for different projects or tasks; I have separate workspaces for shopping, business work, recreation and a “transient” workspace that I use to browse the pages that I'll probably only ever



MCslp.com in Safari



MCslp sites in OmniWeb



view once.

OmniWeb also includes a built-in ad filter and as an added security measure you can configure security settings on a site by site basis. For example, you can enable JavaScript on your financial websites, while having a default setting where JavaScript is disabled. All of this additional security and functionality is built right into the browser, making it more convenient than Firefox or Mozilla which support this functionality only through additional extensions. As a commercial product though, these features do, quite literally, come with a price. Still, OmniWeb remains my favourite browser.

## Conclusion

If what you want is a plain and simple web browser that is free software based then really your choice is limited to Firefox or Camino. Mozilla, while a very capable browser, is technically a complete internet suite and you will only get the best out of Mozilla if you are also happy to use the included email, news and IRC clients.

Camino offers the best all round environment, from an OS X perspective, simply because it offers the best integration with the system and retains the look and feel of the operating system. However, Camino does not support the entire range of Mozilla extensions and this means that if you want to avoid adverts or improve the security of your browser you may want to consider the Firefox project instead.

Firefox is fast, configurable and supports all of the extensions and themes you would want to use. It lacks integration with the operating system, storing authentication information in its own database instead of the keychain for example, and there is no syncing of bookmarks through .Mac or iSync. However, it remains the strongest choice in the free software market.

## Copyright information

© 2005 Martin C Brown

This article is made available under the “Attribution-NonCommercial-NoDerivs” Creative Commons License 2.0 available from <http://creativecommons.org/licenses/by-nc-nd/2.0/>.

## About the author

Martin “MC” Brown is a freelance writer and consultant. He works with Microsoft as an SME, is a featured blogger for ComputerWorld, a founding member of AnswerSquad.com, Technical Director of Foodware.net and, and has written books on topics as diverse as Microsoft Certification, iMacs, and free software programming.