Firefox 3
REVEALED

What's New, What's Hot & What's Not

BY FABIO CEVASCO

Firefox 3 Revealed: Whats New, Whats Hot, & Whats Cool

by Fabio Cevasco

About the Author

Fabio Cevasco works as a technical writer for Siemens Italia.[1] He's also very fond of PHP programming and enjoys writing and blogging about it on his personal web site, H3RALD.com.[2]

About the Technical Editor

Before joining the SitePoint team as a technical editor, Matthew Magain worked as a software developer for IBM and also spent several years teaching English in Japan. He is the organizer for Melbourne's Web Standards Group,[3] and enjoys candlelit dinners and long walks on the beach. He also enjoys writing bios that sound like they belong in the personals column. Matthew lives with his wife Kimberley and daughter Sophia.

About the Technical Director

As Technical Director for SitePoint, Kevin Yank oversees all of its technical publications—books, articles, newsletters, and blogs. He has written over 50 articles for SitePoint, but is best known for his book, *Build Your Own Database Driven Website Using PHP & MySQL*. Kevin lives in Melbourne, Australia, and enjoys performing improvised comedy theatre and flying light aircraft.

About SitePoint

SitePoint specializes in publishing fun, practical, and easy-to-understand content for web professionals. Visit http://www.sitepoint.com/ to access our books, newsletters, articles, and community forums.

---

[1] http://www.siemens.com/index.jsp?sdc_p=dpofc84l4smnu
[2] http://www.h3rald.com/
[3] http://webstandardsgroup.org/

Firefox 3 Revealed

What's New, What's Hot, & What's Not

Welcome To Firefox 3

The third major release of the much-acclaimed Firefox web browser is out. Although slightly belated, this release actually is another important chapter in open source history: Firefox 3 is truly much better than its predecessors (yes, all of them!).

If you weren't too happy with Firefox 2—many quickly tired of its slowness, unattractive appearance, and bare-bones interface (without extensions installed, that is)—you'll be glad to find out that Firefox 3 is much faster, more modern looking, and packed with new, very interesting features. A large number of users (including me) ended up switching to another browser such as Opera because they couldn't endure the severe memory problems of Firefox 2. Different browsers have different problems, though, and while I learned to enjoy Opera's extremely quick response and handy built-in features, I undoubtedly missed some of Firefox extensions. Common pleas from users of Firefox 2 included:

- If only Firefox could be faster …
- If only Firefox didn't eat up so much RAM …
- If only this or that could be improved …

Well, it looks like Mozilla developers really did listen to their users, and the final product truly deserves praise. Let's find out why.

A Streamlined User Interface

Even if a lot of new features are actually "under the hood," Firefox 3 definitely *looks* different in many different ways. There's a new default theme (sorry, make that *four* default themes), and plenty of GUI improvements.

Visual Refresh

If you've already tried out Firefox 3—and you should have, by now—the first thing you would have noticed is its new default theme. No matter which operating system you're running, it is indeed different.

Alex Faaborg, in writing about the Firefox 3 visual refresh,[1] wrote:

"One reason we want to focus to this level of detail on visual integration with various platforms is because the web browser is an incredibly central piece of the user's operating system, and we don't want the user's initial reaction to be that they have modified their computer to add some type of strange, foreign application."

While this comment may have made some users rejoice, I wasn't too happy with the default Windows XP theme, shown in Figure 1—it just doesn't feel as if it's integrated with the operating system at all! If you compare the Windows XP theme with the Windows Vista theme, the only real difference is the color of the arrows: blue for Vista, and green for XP. Apparently the icons were designed by Iconfactory (as were the Windows XP and Vista icons).

---

[1] http://blog.mozilla.com/faaborg/2007/10/10/the-firefox-3-visual-refresh-system-integration/
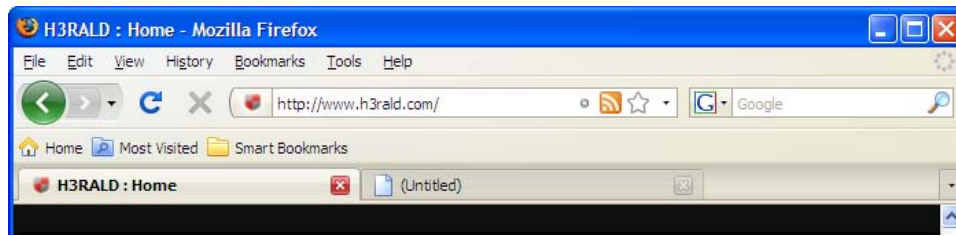
Figure 1. The new Firefox 3 theme for Windows XP

On the other hand, they did a truly awesome job with the default Mac OS X theme: it looks more or less like a native Mac application, even if the small-icon mode is a bit too similar to Safari's.

What about Linux? It hasn't been left out—the Tango theme looks perfect on Ubuntu, although I haven't tested it on the KDE windows manager.

Although some of the themes are really well done, I don't fully understand the necessity of this move. Nobody else is bothering with this: Safari even used the same Mac skin on Windows! Sure, some people prefer a more OS-integrated look and feel, but even Opera came up with the new, shiny, and *cross-platform* Sharp theme[2]. But, as they say, you can't argue about taste.

The AwesomeBar

Remember when you first used the search bar in Firefox? You just had to type in the words you wanted to search for in Google, Yahoo, or any other search engine, press Enter, and the corresponding search result page was loaded automatically for you. This feature is a simple, yet very powerful concept that's now an essential part of every mainstream browser.

Can you imagine using a browser without using the search bar nowadays? Unless it offers something equivalent and possibly more useful like K-Meleon's Ctrl-G to fire up a Google search from the address bar, no, of course you can't.

I experienced a sort of déjà-vu when I tried out the new navigation bar in Firefox 3, pictured in Figure 2, which was immediately dubbed **AwesomeBar** due to its unique and remarkable versatility.

The idea seems simple (to the user's eyes, at least): add search-as-you-type capability to the Firefox bar, looking through all the entries in your Firefox history and bookmarks. What's not so simple is coming up with results that are actually *meaningful*, as Edward Lee described last year in his post, SmartBar to AwesomeBar:[3]

"Firefox will remember the text you typed and the page you selected, so next time when you give a similar input, Firefox will give a higher rank to those pages. It even matches partial inputs, so I can even type just "p" from *p*lanet, and Firefox knows to put that above other pages that just happen to also match "p" …"

---

[2] http://my.opera.com/desktopteam/blog/2008/06/05/looking-sharp/
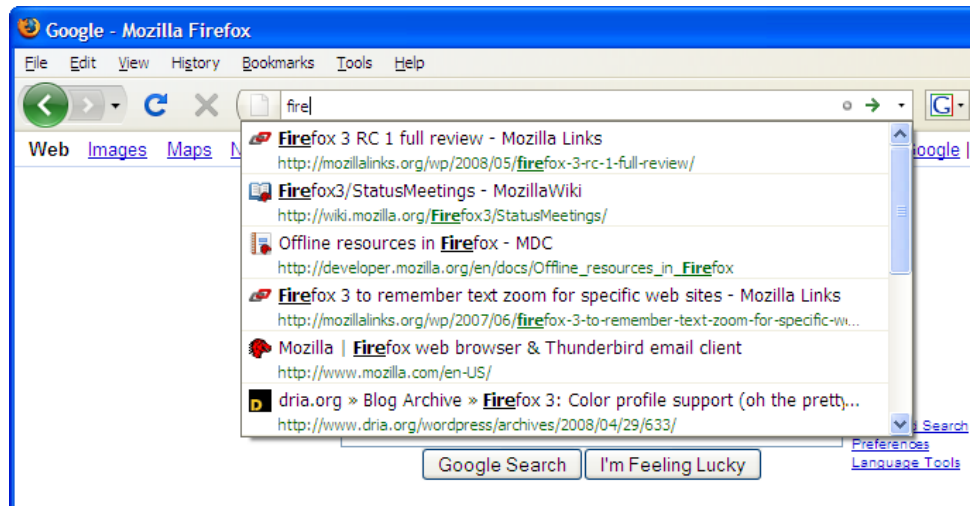[3] http://ed.agadak.net/2007/11/smartbar-to-awesomebar/

Figure 2. The Firefox 3 AwesomeBar in action

Now, this feature is truly useful. In a nutshell, here's what the new AwesomeBar lets you do:

- Search your history and bookmarks for any combination of words while you're typing.

- Restrict your searches to bookmarks, history or tagged pages only by prepending your search terms with "*****", "**^**" or "**+**" respectively.

- Match whole phrases by enclosing words within double quotes.

- Provides search results ordered by frequency and recency.

Compared to the alternatives offered by competitors, the smart searching capabilities offered by Firefox seem far more advanced and useful. Opera 9.5 offers integrated history search within the address bar, but I must admit that the AwesomeBar puts it to shame.

New Places

When I discovered del.icio.us,[4] I slowly started to move my seemingly endless list of Firefox bookmarks to The Cloud. When Yahoo released its official del.icio.us extension for Firefox, I completely forgot about Firefox bookmarks, and I'm still very happy with my choice.

Firefox 3 takes browser-based bookmarking to a whole new level by introducing Places,[5] a feature that's demonstrated in Figure 3.

---

[4] http://del.icio.us/
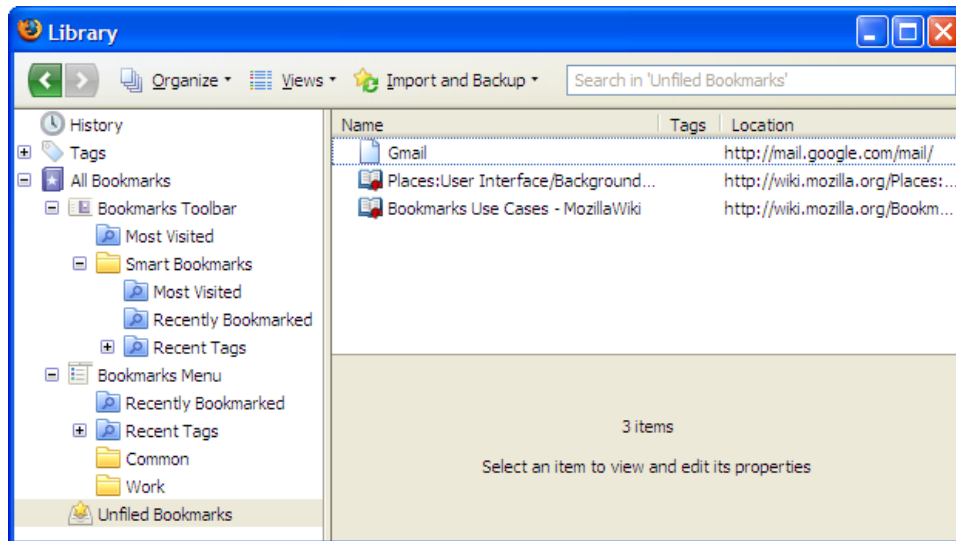
[5] http://wiki.mozilla.org/Places

Figure 3. Storing bookmarks with Firefox 3's Places

The Places feature offers a single interface to organize your History and your Bookmarks at the same time. This interface (accessible via **Bookmarks** > **Organize Bookmarks**) comes packed with a load of new features:

- You can now move pages you visited directly into your bookmark folders by dragging and dropping them, and promoting them to bookmarks (of course you can't do the opposite, though).

- You can search as you type for bookmarks and history items, and perform advanced search queries by defining matching rules on the page title, date visited, and location.

- You can save search queries as persistent search folders.

- You can tag your bookmarks by specifying a list of comma-separated tags.

- You can access the most recent bookmarks and tags as well as the most visited bookmarks via three "smart folders."

- You can bookmark a page with just one click, simply by clicking on the star in the address bar. To specify a folder and tags, and change the title, just click the star again.

- You can back up and restore your bookmarks via a convenient drop-down menu; additionally, it's not necessary to back up your bookmarks anymore, as they're automatically backed up every day.

- You can customize the way you view bookmarks and history items by controlling the fields to display as well as applying different sort criteria.

Personally I can't think of anything else that would improve your bookmarking experience in Firefox. Maybe better del.icio.us integration? There's no need: the new official del.icio.us extension is perfectly integrated with Firefox Places and allows you—among other things—to import your local bookmarks to del.icio.us with a single click.

Furthermore, there are rumors that the AwesomeBar is likely to replace the search bar in future releases of Firefox.

Download Manager

Did you ever try to download a big file with Firefox 2? I did, and I regretted it half an hour later when I discovered that the file could not be downloaded because some error occurred. Of course, because Firefox 2's Download Manager didn't support resuming *even if the FTP server did*, I had no choice but to restart the download with a third-party download manager.

Fortunately, this issue is now a thing of the past: the new Download Manager lives up to its name, finally! It's shown in Figure 4.

First of all, it allows you to pause downloads and resume them later, *even if you exited Firefox in the meantime.* Cross-session resume has been perhaps the most long-awaited improvement for Firefox Download Manager, and now it simply works as everyone expected it to.
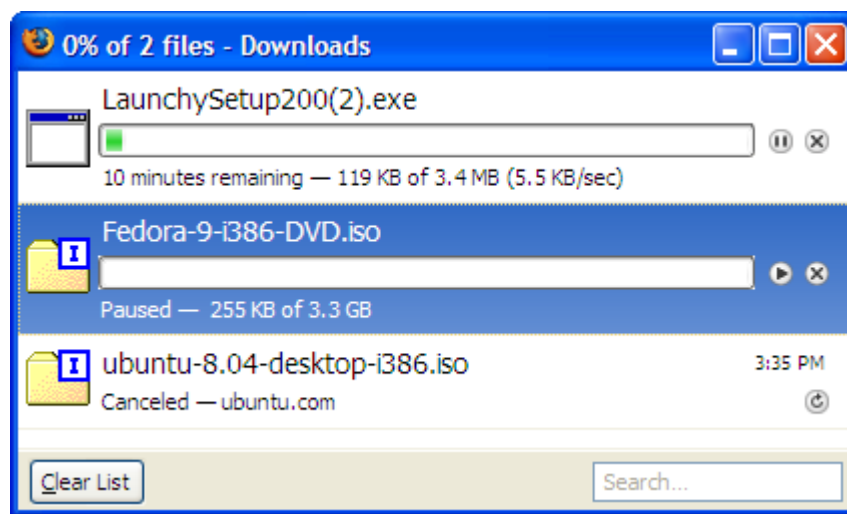
Figure 4. The new and improved Firefox 3 Download Manager

But there's more:

- A summary of all the files currently being downloaded and the estimated download time is now displayed in the status bar. If you click this text, you'll open the Download Manager without having to navigate through the **Tools** menu.

- It is possible to perform a search-as-you-type of the filenames you've recently downloaded simply by entering text inside the search box in the bottom right-hand corner of the Download Manager.

- A few buttons are displayed automatically to pause, resume, cancel, and restart downloads.

- The Download Manager is now integrated with the third-party anti-virus program installed on your computer, so that every file you download will be scanned automatically, for extra security.

Add-on Manager

Similarly, the Add-on Manager, shown in Figure 5, has been improved. It now features four buttons at the top, to load four different views:

Get Add-ons   A list of recommended add-ons is now automatically displayed (and updated) in this view. This means that you can install your favorite add-ons without having to go to the Mozilla Add-ons web site. It even lets you directly search add-ons!

Extensions   Nothing too new here—it's just a list of the extensions that are currently installed. It's possible to set each extension's options, disable it, and uninstall it, through this interface.

Themes   The themes view displays the currently installed theme; it lets you choose a theme and uninstall previously installed themes.

Plugins   This one is new—finally a list of all your plugins (Flash, Acrobat, Java, etc.) can easily be accessed by end users, without having to type **about:plugins** in the address bar. It's possible to selectively disable specific plugins from this view, too.
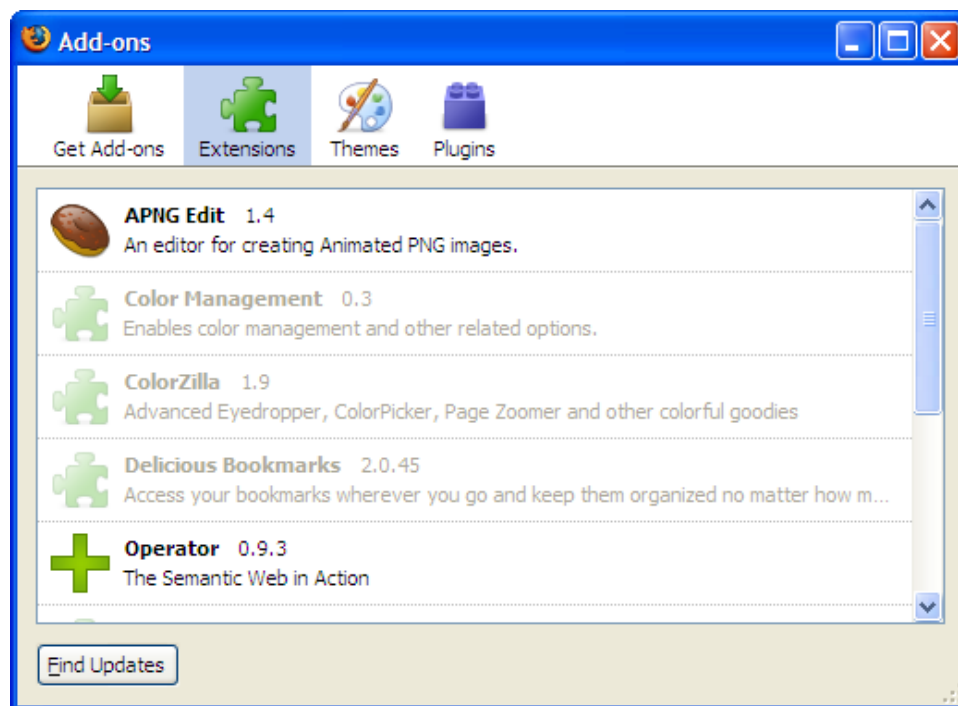
Figure 5. The Add-on Manager in Firefox 3

This is all good, but the killer feature has yet to be released. At the moment, it's still necessary to restart the browser when you're installing an add-on. Apparently this feature was planned but didn't make the cut because it involved core changes that were too invasive.

Opera doesn't require you to restart the browser once you install a new theme, but of course Opera doesn't support plugins. If you just want to update the design of the current Firefox theme, you can always use Personas,[6] after all!

---

[6] http://labs.mozilla.com/2007/12/personas-for-firefox/

Options Dialog

A few relevant changes were also made to the **Options** dialog (still accessible via **Tools** > **Options…**). As Figure 7 illustrates, the first thing to notice is the new OS-dependent icon set used for the top buttons. However, if you look more carefully, you'll see that a new **Applications** button has been added.

The **Applications** tab lets you customize the way Firefox handles known file types and protocols.
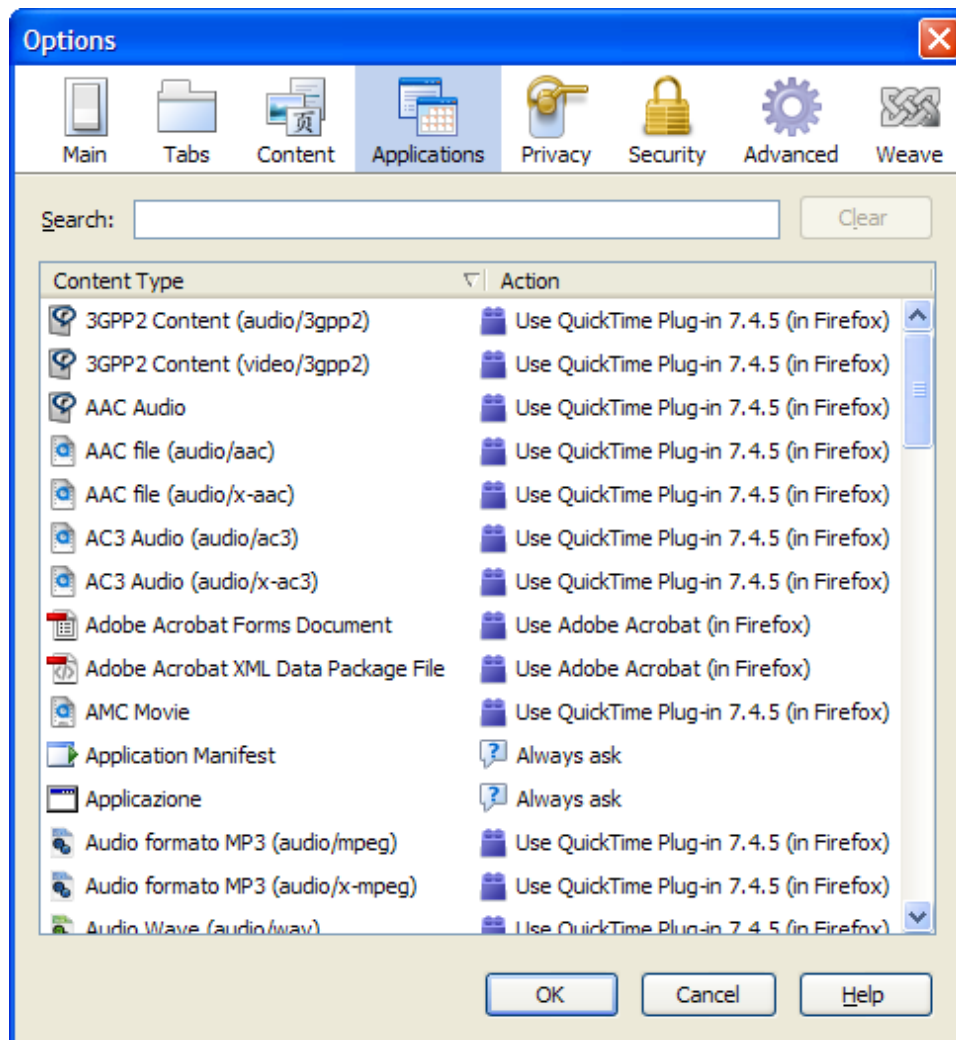
Figure 6. Viewing Application Options in Firefox 3

The view is divided into two columns, the first listing content types, while the other displays the action that's to be performed when that particular content type is accessed. The big improvement is that now it's no longer necessary to hack through **about:config** settings to set content handles: now, you're allowed to browse for a specific set of predefined actions using a GUI.

I was actually very impressed by how refined the application's GUI is—in this release, it appears much more refined then Opera's. Firefox will try to find an appropriate plugin (if installed) to handle file content, but you can also select your favorite application via a dialog that shows all the installed applications which can open that particular file type.

Additionally, if you prefer, you can set the action to **Always ask** if you want Firefox to display the usual **Open/Save** dialog. Besides file actions, you can also set the default action for common Internet protocols and service handlers like `mailto`, `webcal`, or web feeds. Some default online apps can also be selected. For example, for web feeds it's possible to specify Bloglines, Yahoo, or Google. What if your online app isn't listed? Tough luck. Apparently, you'll have to go through the usual **about:config** settings like you did in the old days. Unfortunately, this also applies to the `mailto` handler, for which only Yahoo Mail is provided as a default. If you'd like to use Gmail, you might want to check out how to do so on LifeHacker.[7]

Other minor, but still worthwhile, improvements include the following:

- You have the ability to use the search-as-you-type feature when searching the contents of your browser's cookies. This can be very handy, but unfortunately the search is performed on the cookie contents, not on the domain, so if you type in **google**, you'll get a lot of results (that is, any cookie that has "google" in its content, and there are many).

- In the **Advanced** tab, it's possible to view the history of your Firefox updates, and to configure the way the browser checks for updates.

Page Info Dialog

The old **Page Info** dialog underwent a major overhaul for this release of Firefox, as Figure 7 shows. It features some pretty icons instead of tabs, and it is now accessible via:

- **Tools** > **Page Info**—the same as always

- the address bar—just click on the new "identity button" to bring up the identity overlay dialog and click **More Information…**; you'll be taken directly to the **Page Info** > **Security** tab

Why would you want to access **Page Info** frequently, though? Because it can do amazing things, of course! Perhaps the most important new feature is the ability to save site-specific preferences via the **Permissions** tab. More specifically, the following site-specific settings have been added:

- **Load Images**
- **Open Pop-up Windows**
- **Set Cookies**
- **Install Extensions or Themes**

What Firefox actually does, though, is nothing but set an appropriate exception (which will appear immediately in **Options** > **Content** > **[specific content type] Exception**) for the current web site. That's all? Yes, that's all, but believe me, it's a big time saver.

---

[7] http://lifehacker.com/392287/set-firefox-3-to-launch-gmail-for-mailto-links/
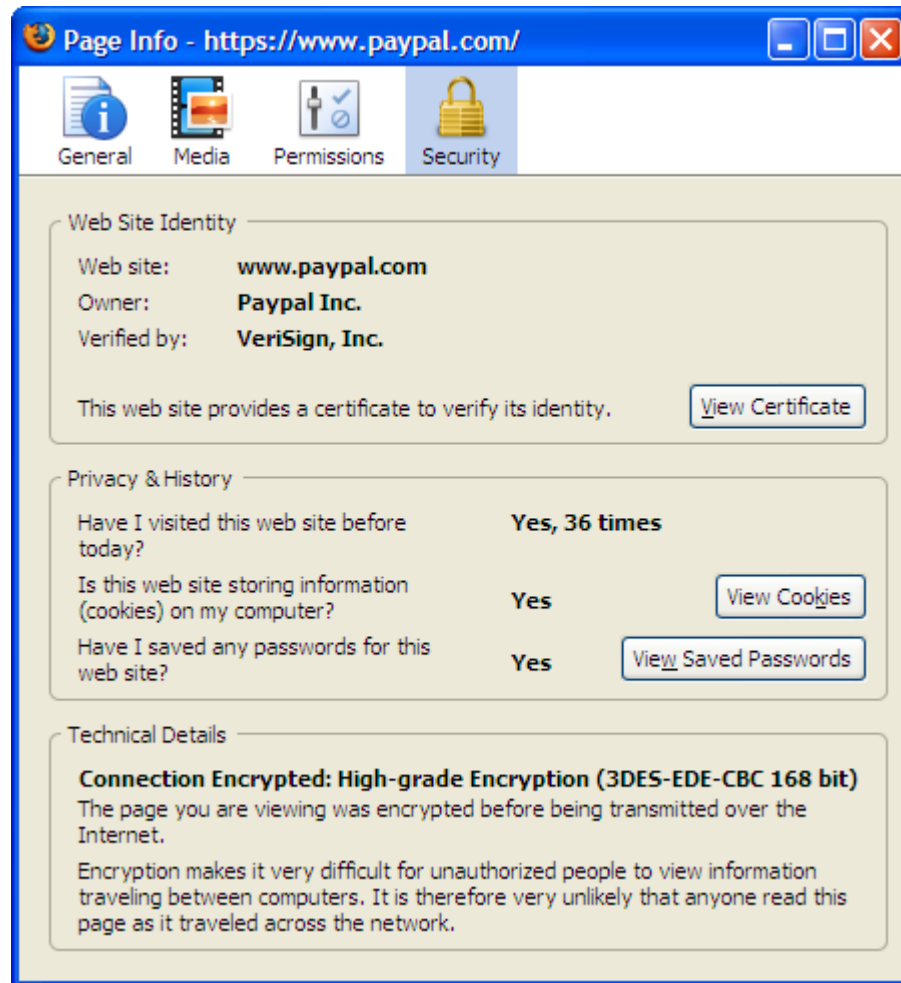
Figure 7. The new Page Info dialog for Firefox 3

A similar behavior can also be found in the **Security** tab—clicking on **View Cookies** and **View Passwords** will lead you to the cookie and password manager respectively, but with a filter already set for the current domain name. Now you realize why Mozilla developers have implemented search filters everywhere—they provide a very smart and elegant solution, delivering two new features for the price of one.

A Few Little Things ...

Additionally, a few small but still significant things were improved or added to the GUI:

- discontinuous text selection, like in MS Word or MS OneNote and other fancy office programs

- full-page zoom: it's possible to zoom in "properly," not only on text, but on images as well. Moreover, Firefox will remember the zoom level you set for a particular page (although the page will actually be resized a fraction of a second after loading)

- smoother tab scrolling, via the mouse wheel

- tab cloning—just drag a tab holding Ctrl to copy it instead of moving it

- resizable search bar—just drag the separator between the address and search bars

- "real" full screen is accessible simply by hitting F11 (even the address bar and tabs will be hidden!)

That's it for the user interface. Okay, so it looks pretty and it's more usable, but what about under the hood? In the next section, we'll take a look at a crucial part of the new Firefox—security.

Security

Security has always been one of the main concerns of the Mozilla Foundation. In fact, one of the most common Firefox marketing slogans was "Browse the web more securely."

Web Site Identification

Firefox 3 delivers more security-oriented features than previous versions of the browser. Whenever you visit a web site that's authenticated via a SSL certificate, if the certificate is valid and verified you'll see the name of the organization that owns the site on a green background right next to the site icon in the address bar.

Try it yourself by visiting paypal.com. Clicking the new "identify web site" icon (the favicon next to the address bar) will open an overlay dialog that displays more information about the owners of the web site and the authority that verified its identify. Additionally, a green icon displaying a passport officer will be shown, as Figure 8 illustrates. That's the signal Firefox uses to say "this site is safe, carry on browsing." Personally, I'd have opted for a less cheesy image, but this one certainly does the job.
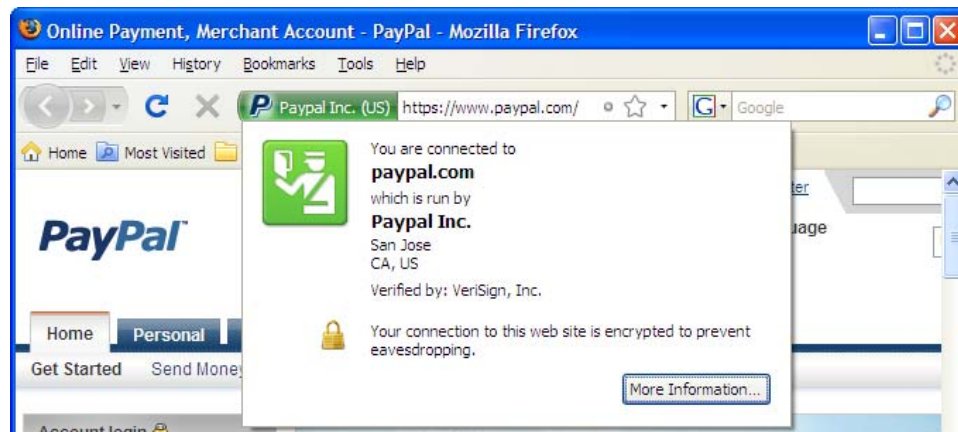
Figure 8. Using Firefox's new web site identification features

If you're still not satisfied with the information provided about the site, you can click the **More Information…** button to access the **Security** tab of the **Page Info** dialog, where it's possible to view the site's security certificates, cookies, technical details, and more.

If you're visiting a web site that doesn't have a SSL certificate, no name is shown next to the icon in the address bar, and the background remains gray. Clicking it will display an overlay dialog to indicate that the web site doesn't supply identity information.

Things change slightly if you visit a web site with which a security certificate is associated that uses the https protocol, but there's a problem with the certificate (usually it has expired: this may even happen with corporate intranet sites). When you visit such a site, *the site won't display*, and a standard Firefox error

page will show. The message says "Secure Connection Failed," and it's accompanied by the usual icon featuring the policemen, this time with a yellow background.

Apparently there's no way to view the site you're trying to access (which scared me to death when I tried to access one of our intranet sites at work!). Luckily, if you click the **Or you can add an exception…** link at the bottom of the dialog, you can choose whether to **Get out of here…** (yes, it literally says that). Doing so will take you back to the standard Firefox start page, or prompt you to **Add an exception**.

This turns out to be a really useful feature if you frequently have to browse web sites whose security certificates expire because nobody bothered to pay the yearly renewal: once you add an exception, you'll be able to browse the site with *no warnings at all*—which is something neither Firefox 2 nor IE 7 allow you to do.

Adding an exception is relatively painless:

1. Click the **Add Exception…** button.
2. In the Add Security Exception dialog, click the **Get Certificate** button.
3. Once the certificate has been retrieved, you can optionally view it by clicking the **View** button.
4. Click **Confirm Security Exception** (the **Permanently Store This Exception** checkbox is enabled by default).

For more information about web site identification in Firefox 3, check out Deb Richardson's blog post[8] (it also contains a precise explanation of each "passport officer" icon).

Password Management

The other security-related area which has been improved in Firefox 3 is the password management system. What's wrong with the way Firefox 2 stored and managed passwords? Let's see …

■ An obtrusive, extremely annoying modal dialog asked you whether you want to save the credential you just provided, *before* they were submitted. The number of incorrect passwords I stored as a result of this feature was embarrassingly high.

■ If you wanted to view a password for a specific site, either you remembered the *exact* URL of the site, or you had to manually scroll down the list of sites for which a password was stored.

Both issues have been fixed in Firefox 3:

■ Password Dialogs are no longer modal. Instead, they slide down from the top of the page allowing you to continue browsing. Furthermore, they're smart enough to appear only after you actually logged in.

■ Search capability has been added to the **Saved Passwords** dialog. What a great idea: a simple change that saves precious minutes of users' time.

What's left to improve? Well, it would be nice to have the secure login extension built in,[9] maybe with an icon in the address bar (you can do virtually anything in it, even post to Twitter).[10]

---

[8] http://www.dria.org/wordpress/archives/2008/05/06/635/

[9] https://addons.mozilla.org/en-US/firefox/addon/4429

[10] https://addons.mozilla.org/en-US/firefox/addon/4664

Now that we've explored the ways in which Firefox 3 is more secure than its predecessor, let's see how it performs.

Performance

Firefox 3 is noticeably faster and more lightweight than its predecessors. This isn't the usual "this is the fastest Firefox yet" announcement: in this release, the browser has finally become usable without being a memory hog. I'm not being harsh: this is perhaps the biggest complaint that I and a lot of other users have had about the browser since Firefox 1.5 came out.

Finally, everything seems lightning fast:

- It starts … er … instantly—meaning that Opera and Safari are still faster on startup, but you don't have to wait half a minute for Firefox to finally come to life.

- "Heavily-Ajaxified" pages are *much* faster—I was particularly impressed by the speed of Gmail: you don't really have to wait more than a fraction of a second when you archive emails, for example.

- You can leave Firefox 3 on for hours without having it eating more and more of your RAM.

This is the short story, from the point of view of the everyday user.

Memory Management Improvements

As a matter of fact, Firefox developers really did an outstanding job this time. There's a very interesting post on pavlov.net[11] explaining what was done in detail.

The memory problems with Firefox 2 were apparently attributable, in part at least, to memory fragmentation. This is similar to the phenomenon that affects hard drive memory when you install and uninstall programs: your disk becomes fragmented, which can degrade performance. A similar thing occurs with RAM, only that in Firefox's case, it happens whenever you open and close web pages in tabs: whenever a new page is loaded, if it's bigger than the largest space in the heap, it won't fit in; the only solution for the program is to increase the heap size.

Fortunately, the adoption of jemalloc (the same memory allocator used by FreeBSD) reduced memory fragmentation and, apparently, caused a 22% drop in the total amount of memory used by Firefox (on Windows Vista, at least).

Another important improvement to memory management was the introduction of a new "cycle collector" to automatically break mutual references between two objects when they're no longer being used. This, in addition to other tweaks, such as improved expiration policies for Firefox's caches and optimized image storage, have all helped to successfully rein in the browser's previous memory usage woes.

Really Fast JavaScript Execution

Besides improving memory management, Firefox developers also worked hard to improve JavaScript performance. According to the benchmarks conducted by a few people, Firefox 3 seems to be significantly faster than its predecessor when it comes to executing JavaScript code. Most of the benchmarks were con-

---

[11] http://blog.pavlov.net/2008/03/11/firefox-3-memory-usage/

ducted using WebKit's SunSpider JavaScript Benchmark,[12] and in most tests, Firefox seems to perform better than its main competitors (Opera 9.5 beta, Safari 3.1 beta, and Internet Explorer 8 beta).

Of course, benchmarks shouldn't be taken too seriously, but personally, I noticed that sites like Gmail and Google Reader seemed snappier in Firefox 3 than in Opera 9.5 beta, for example. Try using Gmail in Firefox 3: there are virtually no delays when performing "simple" Ajax calls to archive or load emails.

Cairo

I remember when I installed the first beta versions of Firefox 3: about 70% of the text on web sites seemed to have lost crispness and antialiasing for no apparent reason. This happened because at the time Firefox was using an unpatched version of the Cairo graphics library,[13] which didn't work that well on LCD displays. But as of Firefox 3 RC1, the issue was fixed.

Despite the initial problems, the integration of Cairo into Firefox core brought quite a few advantages to both users, namely:

- hardware-accelerated 2D image rendering
- fewer graphic rendering bugs
- better-looking graphics

Also, Gecko developers won't have to worry too much about maintaining graphics-related code, as most of their burden will be lifted by Cairo's development team: this appears to be another very significant victory for the Open Source model, that's for sure.

Reflow Refactoring

Cairo isn't the only big change concerning Firefox rendering, though. Firefox 3 sports a brand new (and still beta) release of the Gecko layout engine. Gecko 1.9b introduces improvements in many areas, but the so-called "reflow refactoring" seems to be one of the most notable ones.

The concept of **reflowing** has been defined in CSS 2.1 specifications to indicate the way browsers should position elements on a web page. More specifically, earlier versions of Firefox suffered a lot of bugs relating to the way the browser performed **incremental reflows**—the repositioning of elements on the page while the page is loading.

Incremental reflow problems were often noticeable on pages in which the width of some elements (like tables and images) were not defined or defined via percentages. In such cases, the browser was caused to perform an incremental reflow to determine the correct width based on other elements on the page.

The most famous of these bugs was the Slashdot Rendering Bug,[14] which appeared in Firefox 0.8. This caused the incorrect rendering of the text on the Slashdot web page.

Apparently quite a few similar bugs were discovered (and fixed) in the last few years, and they were all caused by the way the Gecko engine performed incremental reflows. For a more detailed explanation of

---

[12] ???

[13] http://cairographics.org/

[14] http://www.deftone.com/blogzilla/archives/firefox_and_rendering_slashdot.html

the situation, refer to the reflow refactoring page of Mozilla wiki.[15] It's quite an interesting read, but is also a little technical for the average reader, as it mentions various implementation aspects in detail.

What really matters is that as of Firefox 3, a lot (all?) bugs related to incremental reflow and intrinsic width were fixed, the browser is now able to display page elements more consistently, and it passes the Acid 2 test.[16]

Unlike Opera and Safari developers, Firefox developers didn't really try to build the browser to pass the Acid 3 test, the main reason being that Firefox 3 release was too close to bother with it.

Color Profiles

Designers will be happy (or maybe not…) to discover that Firefox 3 supports color profiles.[17] This means that Firefox is now able to render images (pictures in particular) with their *proper* colors, for instance, when they're viewed in professional photo editing programs like Adobe Photoshop. Deb Richardson wrote a very interesting article[18] about this new hidden feature, which is worth reading.

Color support can be enabled either by installing the Color Management extension[19] or by editing two settings in your **about:config** page (if you know what you're doing).

This feature can be a great thing for Flickr users and photographers, but others may be a little bit reluctant to enable it, due to two significant trade-offs:

- It affects the browser's rendering speed (it becomes 10-15% slower).

- Color profiles aren't supported by plugins, and this may break color matching with certain web pages (designers, beware!).

These are the reasons why this feature is disabled by default (and why I'm going to disable the Color Management extension in my browser for the time being).

A Developer's Dream

If you think that this new Firefox release is all about user interface improvements and performance, you're wrong. This is perhaps the most notable developer-oriented release the Mozilla Foundation has produced so far: the number of open-source APIs and developer goodies that ship along with Firefox 3 is unprecedented and very impressive.

CSS Improvements

Firefox 3 delivers higher CSS 2.1 support. A lot of CSS-related bugs have been fixed, and a few new features have been added.

---

[15] http://wiki.mozilla.org/Gecko:Reflow_Refactoring

[16] http://www.webstandards.org/files/acid2/test.html

[17] http://en.wikipedia.org/wiki/Color_profile

[18] http://www.dria.org/wordpress/archives/2008/04/29/633/

[19] https://addons.mozilla.org/en-US/firefox/addon/6891/

Kevin Yank wrote an informative post about two of Firefox 3's new CSS features:[20] its support for the `­` HTML entity (or "soft hyphen"), and for the `inline-block` display type.

Soft hyphens can be used to tell the browser where it's acceptable to break a word—a feature that's particularly useful for long words which wouldn't otherwise be hyphenated. Apparently Firefox was the only major browser not to support this capability, so it will be possible and relatively safe to start using soft hyphens from now on.

The `inline-block` value for the `display` property, on the other hand, is not supported by IE7, so you still need to take precautions (such as creating a separate style sheet for "other" browsers), but it's still a cool feature: if the display property of an element is set to `inline-block`, the element will be rendered as `inline`, but its content as `block`. For more details and examples, check out Kevin's post.

Other CSS improvements include:

- color translucency support for the `color` property via `rgba()` and `hsla()`

- `font-size-adjust` now works on non-Windows platforms too

- `ime-mode` property to set the default state of form fields to normal, active, inactive, or disabled

- `text-rendering` property to instruct the Gecko engine to render text optimizing speed, legibility, or geometric precision (see some examples on the Mozilla developer site)[21]

- `:default` CSS selector, to set the default properties of similar elements. For other CSS selectors, you'll have to wait till Firefox 3.1, David Baron reports.[22]

- the ability to set `content` and `cursor` to `none`

- support for negative `z-index` values (finally!)

JavaScript Goodies

Firefox 3 supports JavaScript 1.8, which includes a few interesting new advantages, such as:

- expression closures; we can now write `function(x) x * x` instead of the more verbose `function(x) { return x * x; }`

- generator expressions; it's no longer necessary to create custom generator functions like we did in JavaScript 1.7

- the `reduce()` and `reduceRight()` methods; these let arrays run a function over all elements of an array, collecting results from previous calls

Head over to the Mozilla for Developers web site for more details.[23]

---

[20] http://www.sitepoint.com/blogs/2008/05/23/two-hidden-features-new-in-firefox-3/

[21] http://developer.mozilla.org/samples/cssref/text-rendering.html

[22] http://dbaron.org/log/20080603-new-selectors/

[23] http://developer.mozilla.org/en/docs/New_in_JavaScript_1.8

What really makes the difference, though, is the number of new APIs and interfaces that have been released by Mozilla developers.

The Firefox User Extension Library (FUEL)[24] includes a set of very useful objects that will simplify the life of those who want to develop Firefox extensions, such as:

- The `Annotation` object, to store custom data about a web page
- The `Bookmark`, `BookmarkFolder` and `BookmarkRoot` objects, to handle bookmarks
- The `BrowserTab` object, to control browser tabs, their state and contents
- The `Console` object, to send messages to Firefox Error Console
- The `Events`, `EventItem` and `EventListener` objects to manage browser events
- The `Extension` and `Extensions` objects, to access the currently-installed extensions
- The `Preference` and `PreferenceBranch` objects, to access Firefox options
- The `Window` object, to manage browser windows and their tabs
- The `SessionStorage` objects, to manage storage items

Additionally, quite a few new public interfaces have been released:

- `nsIJSON`—to encode and decode JSON strings.
- `nsIIdleService`—to monitor user (in)activity.
- `nsIZipWriter`—to compress data into ZIP files.
- `nsIThreadManager`—to create and manage threads and dispatch events to them.
- `nsIParentalControlService`—to access Windows Vista's build-in parental control.
- `nsILoginManager` and `nsILoginInfo`—to access the password manager.

Last but not least, a new shiny Places API[25] to access the new bookmarking and history features.

Offline Applications Made Easier

The HTML 5 Working Draft[26] has been around for a while, and although it doesn't yet qualify as a new, standardized HTML specification, it's on its way to doing so. A clear sign of this possibility is the increasing support browsers are rolling out, tag by tag, as time goes on.

Firefox 3 puts into practice quite a few interesting HTML5 features, and **offline caching** is definitely one of them. Imagine being able to use your favorite web applications offline as well! In fact Google Gears was created for that very purpose.[27] But what we have here is a more standardized, native form of support for offline applications, which could potentially change the way such applications are developed (for instance, they might not require a third-party API).

Firefox 2 already offered support for DOM storage,[28] which provides a way to overcome the most annoying limitations of browser cookies. In a nutshell, DOM storage allows developers to programmatically store resources on the client machine without having to worry too much about their size or making sure that they're secure. The idea behind DOM storage is to provide storage via an easy-to-use and standardized API.

---

[24] http://developer.mozilla.org/en/docs/FUEL

[25] http://developer.mozilla.org/en/docs/Places

[26] http://www.w3.org/TR/2008/WD-html5-20080122/

[27] http://gears.google.com/

[28] http://developer.mozilla.org/en/docs/DOM:Storage

Firefox 3, on the other hand, actually makes DOM storage support useful, by providing a few other "missing pieces" necessary to develop offline applications. The first one is a way to identify offline resources via a cache manifest file, which may look something like this:

```
CACHE MANIFEST
# v1
# Comments must be preceded by a hash symbol
http://www.example.com/index.html
http://www.example.com/figure1.png
http://www.example.com/figure2.png
http://www.example.com/figure3.png
http://www.example.com/effects.js
http://www.example.com/ajax.jar
```

This manifest file is linked directly in the `<html>` tag:

```
<html manifest="http://www.example.com/cache-manifest">
```

In this way, you can tag resources for offline use: static HTML pages, images, and scripts! Furthermore, by using the JAR: protocol,[29] it's also possible to compress scripts to save space (and download time).

Note that as of Firefox 3, offline resources must be listed within the cache manifest, rather than being included in the page header using `<link rel="offline">` or similar. This decision was taken in order to make Firefox compliant with the actual WHATWG specification.[30]

At the moment, Firefox 3 support for offline caching lacks:

- resource versioning
- fallback entries
- opportunistically cached entries

Other than that, everything seems to be on track.

Finally, the `navigator.onLine` event has been added, which allows a page's JavaScript code to determine when the browser is online or offline. One way to make sure you're offline is to select **File** > **Work Offline**, but it should also be fired when the network connection is lost or not available.

Success stories? Well, there's at least one for now: Chris Double added offline support to the Zimbra email client[31] (a screencast is available[32]).

Now all we need is other browsers to fully embrace the Web Application 1.0 specification and offer similar support …

---

[29] http://docs.sun.com/source/819-0913/author/jar.html#jarprotocol

[30] http://www.whatwg.org/specs/web-apps/current-work/#offline

[31] http://www.zimbra.com/

[32] http://www.bluishcoder.co.nz/offlinezimbra/

Additional HTML 5 and Web Application 1.0 Support

Besides partial support for offline resources, Firefox developers hand-picked a few other interesting HTML 5 features which have been implemented in Firefox 3.

Did you ever try to develop a cross-browser WYSIWYG HTML editor? If you did, chances are that you know that Firefox offers Midas, a JavaScript rich-text editor.[33] In particular, a collection of DHTML commands (the same used by IE) have been implemented to perform the most usual editing operations like making text bold, italic, and so on.

The problem with Firefox versions up to 2 was that we had to display the portion of the page we wanted the user to edit inside an IFrame, because only the `designMode` property of the document object was implemented, allowing users to edit a whole document, rather than just a section of it.

But we don't have to do that any more: it's now possible to set the `contentEditable` property of a particular HTML element to `true` to make that element editable. This means, for example, that you can make even a single `div` element editable, as shown in the demo by Mark Finkle.[34] The good news is that this functionality now works in all major browsers (including Firefox, IE, and Opera).

Other features implemented in Firefox 3 concerning HTML 5 and Web Application 1.0 drafts include:

- support for `document.activeElement` and `document.hasFocus` to determine whether an element has focus or not

- support for `drag` and `dragend` events to monitor drag-and-drop operations (check out the demo[35])

- native support for `getElementsByClassName`

- enhanced `<canvas>` support with new methods to perform transformations[36] and draw text[37]

- support for cross-document messaging via `window.postMessage` to send string messages across browser windows (as outlined by a tutorial by John Resig[38])

- support for the `ping` attribute for anchors and `area` elements; Darin Fisher explains: "When a user follows a link via one of these tags, the browser will send notification pings to the specified URLs after following the link."[39]

Unfortunately, the most hyped HTML 5 elements, `<video>` and `<audio>`, didn't make the cut. Chris Double has reported that he's still working on them,[40] and that they're likely to be shipped later on, as they are marked as a top-priority feature scheduled for Gecko 1.9.1.

---

[33] http://developer.mozilla.org/en/docs/Midas

[34] http://starkravingfinkle.org/blog/wp-content/uploads/2007/07/contenteditable.htm

[35] http://developer.mozilla.org/samples/dragdrop/dragevents.html

[36] http://developer.mozilla.org/en/docs/Canvas_tutorial:Transformations#Transforms

[37] http://developer.mozilla.org/en/docs/Drawing_text_using_a_canvas

[38] http://ejohn.org/blog/postmessage-api-changes/

[39] http://weblogs.mozillazine.org/darin/archives/009594.html

[40] http://www.bluishcoder.co.nz/2008/05/firefox-html5-video-and-audio-update.html

Still, it must be noted that video and audio support will only be offered for *free* codecs, like Theora and Ogg/Vorbis, mainly because of the patent issues associated with proprietary formats like MPEG-4 and MP3.

Improved SVG Support

Firefox supported Scalable Vector Graphics (SVG) since version 2. SVG is an XML-based language used to represent vector images. Some examples can be found on the W3Schools web site,[41] including the following:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
   "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="100%" height="100%" version="1.1"
   xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" rx="20" ry="20" width="250" height="100"
   style="fill:red;stroke:black;stroke-width:5;opacity:0.5"/>
</svg>
```

This code, if parsed by an SVG-compatible browser such as Firefox 2 or 3, produces a rectangle with rounded corners,[42] like the one shown in Figure 9.

Figure 9. The SVG code above renders a rectangle with rounded corners

Arguably, SVG (and SVG manipulation via JavaScript) can be considered a replacement for proprietary technologies such as Adobe Flash and Microsoft Silverlight, with the important advantages that it's an open standard (SVG 1.1 is a W3C recommendation[43]) in plain text format rather than binary.

Firefox 3 extends Firefox's SVG support by implementing the `pattern`, `foreignObject`, and `mask` elements, and all the filters defined in SVG 1.1 specification.[44] Additionally, a lot of SVG-related bugs have been fixed in this release, so that it's now possible to interact in an acceptable way with complex SVG-based applications such as the Lively Kernel.[45]

---

[41] http://www.w3schools.com/svg/svg_examples.asp

[42] http://www.w3schools.com/svg/rect4.svg

[43] http://www.w3.org/Graphics/SVG/

[44] http://developer.mozilla.org/en/docs/SVG_in_Firefox

[45] http://livelykernel.sunlabs.com/

Still, Firefox 3 isn't as fast as Safari 3 and WebKit at rendering SVG, although it's definitely faster than Firefox 2.

Animated PNG Images

Another very important feature related to graphics and animations is native APNG support.[46] A key incentive for implementing APNG support is to be able to display animation without relying upon Compuserve's proprietary Graphics Interchange Format (GIF).

If you were browsing the web in the nineties, you certainly remember those cute but very annoying, poor-quality GIF animations that plagued the Web. They only had 256 colors, and they didn't really support transparency. More recently, they have been rediscovered as throbbers[47] for almost any Ajax web application: whenever you have to wait for the end of an `XMLHttpRequest` call, chances are that you'll see a throbber spinning to indicate that you have to wait a little longer.

Although GIF animations look ugly for anything more complex than a throbber, APNG images do not. APNG images are definitely smoother, and they support transparency and 24-bit colors. You can see a few examples on the Mozilla Labs site, including a spinning version of the Firefox logo in Figure 10.[48]

Figure 10. The Firefox logo, an animated version of which appears on the Mozilla Labs site

Unfortunately APNG is only supported by Firefox 3 and Opera 3.5 for now (don't forget that IE barely started to support static PNGs with IE7!), so you can't expect everyone to see your fancy APNG animations as you intended. The good news is that APNG is fully backwards compatible with PNG, so any browser that's able to display PNG correctly (or incorrectly, like IE6 does), will render an APNG animation as a static PNG.

How do you create APNGs? Well, you can just use the appropriate Firefox Add-on.[49]

---

[46] http://animatedpng.com/
[47] http://en.wikipedia.org/wiki/Throbber
[48] http://labs.mozilla.com/2007/08/better-animations-in-firefox-3/
[49] https://addons.mozilla.org/en-US/firefox/addon/5519

Microformats

Firefox 3 comes with native Microformats support. That's good to know, but what does it actually mean?

Microformats, as defined by microformats.org[50], are:

"[…] a set of simple, open data formats built upon existing and widely adopted standards. Instead of throwing away what works today, microformats intend to solve simpler problems first by adapting to current behaviors and usage patterns (e.g. XHTML, blogging)."

More specifically, a microformat is a representation of meaningful information using semantic XHTML and following a set of rules and conventions. Microformats.org defined a few open standards for microformats, including:

- `hCard`, for people and organizations
- `hCalendar`, for calendars and events
- `VoteLinks` and `hReview`, for opinions, ratings and reviews
- `XFN`, for social networks
- `rel-license`, for licenses
- `rel-tag`, for tags, keywords and categories
- `XOXO`, for lists and outlines

For example, an `hCard` looks like this (the example is taken from the Microformats Wiki[51]):

```
<div class="vcard">
  <a class="fn org url" href="http://www.commerce.net/">CommerceNet
  </a>
  <div class="adr">
    <span class="type">Work</span>:
    <div class="street-address">169 University Avenue</div>
    <span class="locality">Palo Alto</span>,
    <abbr class="region" title="California">CA</abbr>
    <span class="postal-code">94301</span>
    <div class="country-name">USA</div>
  </div>
  <div class="tel">
    <span class="type">Work</span> +1-650-289-4040
  </div>
  <div class="tel">
    <span class="type">Fax</span> +1-650-289-4041
  </div>
  <div>Email: <span class="email">info@commerce.net</span>
  </div>
</div>
```

If you take a closer look, this is nothing but ordinary XHTML, which follows a set of conventions concerning nesting, CSS class names, and tag types. As a matter of fact, despite its fancy name, the previous `hCard` will be rendered just like ordinary XHTML code.

---

[50] Microformats.org definition of Microformats

[51] http://microformats.org/wiki/hcard

This is important because if a developer decided to put an `hCard` on a page, it would follow the same standard as other `hCards`, which would make it easy to parse the `hCard` data automatically, for example. To understand what this means, you'd do well to install the Operator extension for Firefox (available since version 2).[52]

Operator is able to parse microformats and present them to the user in an organized way via a toolbar, as shown in Figure 11.
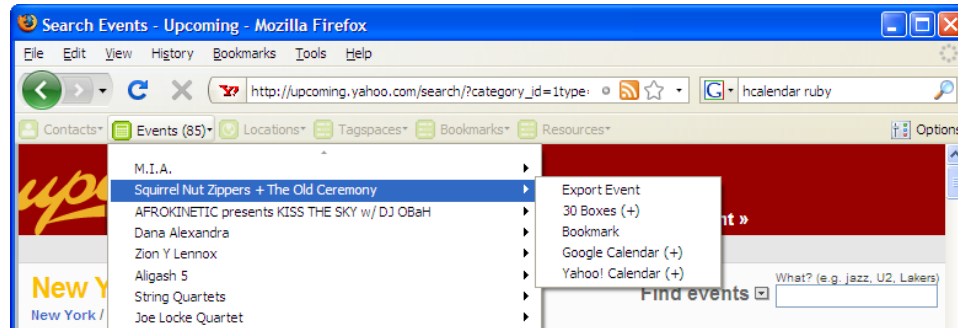

Figure 11. The Operator toolbar in action

As you can see from Figure 11, the information is still present on the actual web page, but it has also been parsed by Operator, which enables the user to perform a set of default actions that are available for the specific information type. In short, as outlined on ReadWriteWeb,[53] Firefox may turn into a sort of information broker by using microformats to connect the users to many popular web and desktop services. The Operator extension is still available, along with a few other ones,[54] so Firefox 3 can parse and display microformats via extensions, exactly like Firefox 2.

So where's the long-awaited native support for microformats? Apparently it's only under the hood: Firefox 3 exposes a brand new API for parsing microformats using JavaScript.[55] That's definitely a good thing, but it's a shame that microformats aren't parsed and rendered in some way natively by the browser. Apparently this omission was due only to the inability of some of the folks at Mozilla to make up their mind and define a standard GUI: a lot of ideas were proposed,[56] and a few cool mockups were created, but none made it to the end product.

Beyond Firefox

Firefox is definitely the most well-known of the Mozilla Foundation's products, but there are a few other interesting projects worth keeping an eye on.

Fennec

Fennec[57] is the codename for a new Mozilla project aimed to create a competitive web browser for the mobile phone market. This is not the first time the Mozilla Foundation attempted to build a mobile browser:

---

[52] https://addons.mozilla.org/en-US/firefox/addon/4106

[53] http://www.readwriteweb.com/archives/mozilla_does_microformats_firefox3.php

[54] http://microformats.org/wiki/firefox-extensions

[55] http://developer.mozilla.org/en/docs/Using_microformats

[56] http://wiki.mozilla.org/Microformats/UE/ideas

[57] http://wiki.mozilla.org/Mobile/FennecVision

Minimo[58] actually worked, but compared to competitors such as Opera Mobile and Internet Explorer Mobile, it's very slow and heavy on the memory usage—or at least it is on my WM6 phone.

Fennec, on the other hand, aims to become a lightweight, feature-rich, and innovative mobile browser, and although it's still in pre-alpha stage, Ars Technica reports that it's already quite usable.[59] Figure 12 shows the Fennec browser in action.

Figure 12. Fennec: Firefox for the mobile web

This would be quite an achievement for the Mozilla Foundation, as it would be the first significant free and open source mobile browser ever made.

Prism

Mozilla Prism[60] is another lesser-known Mozilla project aimed to bring web applications to the desktop. Essentially, it's nothing more than a single-tab, extremely minimalist browser which can be used to run common web applications like Gmail or Google Reader on the desktop.

In its current state, it's nothing more than a fancy way to create and handle shortcuts to web applications; those shortcuts, when clicked, will open Prism and load the web address for the web application specified by the user.

Why bother? Well, it's a matter of perception more than anything: it's about having a separate, distraction-free browser to use just for a single web application. Things will actually become more interesting when proper support for offline applications is included.

At the moment, there are quite a few interesting features in Prism's wish list:[61]

---

[58] http://www.mozilla.org/projects/minimo/

[59] http://arstechnica.com/news.ars/post/20080409-first-look-mozilla-fennec-targets-handheld-browser-market.html

[60] http://labs.mozilla.com/2007/10/prism/

[61] http://wiki.mozilla.org/Prism#Wishlist

- Enable extension support so Prism can be tweaked or enhanced. Also, web apps themselves can be tweaked via extensions.

- Add menus for page setup, open profile, and work offline.

- Add url/domain/subdomain restrictions (external links won't work).

- Load "sidebar services." That is, users have the option to load services that are intended to load in a sidebar (Google Chat and Google Notebook have specific web pages designed for the sidebar, for example); such a page will load in Prism as a fixed-size window on the left of the screen.

- Optionally configure a web app so that it doesn't share cookies with other web apps in the same Prism. This could allow multiple Google apps to be open with different accounts (an a@gmail.com Gmail account; a b@msn.com Reader account; and c@hostedgmail.com for hosted Gmail).

When those features are implemented, Prism will definitely become more interesting.

Weave

If Prism aims to bring the Web to the desktop, Mozilla Weave[62] is focused on the very opposite: bring your browser (Firefox) to the Web.

You can think of it as a smart synchronization platform, to synchronize multiple instance of Firefox through the Internet, in a similar way as Opera Link does.[63]

Weave is, as a matter of fact, just an ordinary Firefox extension. All you have to do is install it, register for an account,[64] and off you go.

At its current stage, Weave gives the user the possibility to synchronize:

- bookmarks
- cookies
- passwords
- history
- form data

Compared to Opera Link, which lets you synchronize your bookmarks, speed dial, personal bar, and notes, Weave seems to be ahead in term of features, and perhaps maturity as well. I have been using it for a few months, and I must say I'm quite satisfied, even if I hardly use Firefox bookmarks at all.

The aim of this project—to me at least—seems crystal clear: offer such a useful service that it makes you stick to Firefox for life. That's how it feels, anyway.

Think about it: Firefox 3 comes with a really fancy bookmarks and history manager, which lets you bookmark pages with one click and tag them with two. The AwesomeBar lets you search your bookmarks and history

---

[62] http://labs.mozilla.com/2007/12/introducing-weave/

[63] http://labs.mozilla.com/2007/12/introducing-weave/

[64] https://services.mozilla.com/

in a really smart way, and Weave makes sure that wherever you go, you'll always be able access all the pages you need quickly.

I must admit that I'm seriously tempted to drop del.icio.us and just start using Firefox Bookmarks instead: if you use del.icio.us or other social bookmarking services just to store your bookmarks online and make them available wherever you are, then Weave is ideal.

There's only one problem with it: it's Firefox-dependent. That's the reason why I'm not using it extensively: I don't want to become dependent on it, and I don't want to become dependent on Firefox either! That's not because I don't like Firefox, but because I'm a browser addict and I love experimenting with as many browsers as possible.

Weave isn't feature-complete, of course. To be feature-complete, it would have to be able to synchronize your add-ons, extensions, and themes: at that point, it would become almost irresistible. Imagine the possibilities for both users and the Mozilla Foundation: Firefox your web platform, wherever you are! Neat, and scary, too.

The Bottom Line

Firefox 3 is—as with every major release—the best release of Firefox ever. This time, the number of new features that were released is truly impressive:

- End users will be pleased to use a much faster and prettier browser.

- Web developers won't change the way they develop web sites, but may at least try to experiment with new HTML 5 elements and features.

- Extension developers will be able to interact with the browser's internals more easily, thanks to the huge number of new APIs that have been made available (and are fully documented!).

Personally, I felt that this release was focused very much on accessibility, both in terms of user interface and new developer APIs.

In particular, you can search and filter everything from almost anywhere in the GUI: the Firefox team added search support to almost every single browser module, and they reduced the number of clicks you need to use to access certain functions. This is something that is not yet available in other browsers, and is perhaps one of the most significant improvements on past releases.

The progressive adoption of new standards—and even drafts—is also quite remarkable. As a matter of fact, Opera is still ahead in this case, but Firefox is catching up quickly.

When it comes to web development, it doesn't really matter which browser supports more features; the important consideration is which features are supported by more browsers. For this reason, web developers will definitely be pleased to hear that they can safely use the `contentEditable` attribute because it's supported by three major browsers, rather than discovering that Firefox only offers embryonic support for offline applications.

Compared to Opera and Safari, in particular, Firefox is likely to remain a slower and heavier browser, for a simple reason: its very architecture. Firefox is very open and modular, and allows users to install a plethora of extensions, while others don't.

Opera and Safari are not likely to open up any time soon (even if WebKit, as a matter of fact, is open source), and they won't offer extensions either: those, it seems, are unique to Firefox, and will remain so.

It all boils down to what users want. If you want a browser that can be customized in every possible way, and can become anything you want, from an MP3 player to an FTP client, Firefox is definitely the best (and the only) choice. On the other hand, if you want "just a browser", which does its job really well, you may want to check out Opera and (when it becomes a bit more stable) Safari.

Conclusion

To conclude, I'd like to congratulate everyone who made Firefox 3 possible: developers, testers, and all active users who "do their part," in one way or another. It really feels like the Mozilla team is doing its very best to listen to users and their feedback, and this release is a clear example that open source can still deliver the most outstanding software, just the way *you* want it.