

Firefox Tweak Guide



By [Thomas McGuire](#) on June 28th, 2005
Editor: [Julio Franco H.](#)

Mozilla Firefox [website](#)

Find [software prices](#).

Microsoft has reigned the browser market for a long time, ever since it took the original Netscape browser out of the picture; there has been no competitor able to keep up with Internet Explorer. Unfortunately for end users however, the software giant has been sitting duck for about half of its lifetime, adding almost no new features since IE 6 was introduced in 2001, and until very recently, barely keeping up with security holes found in its browser.

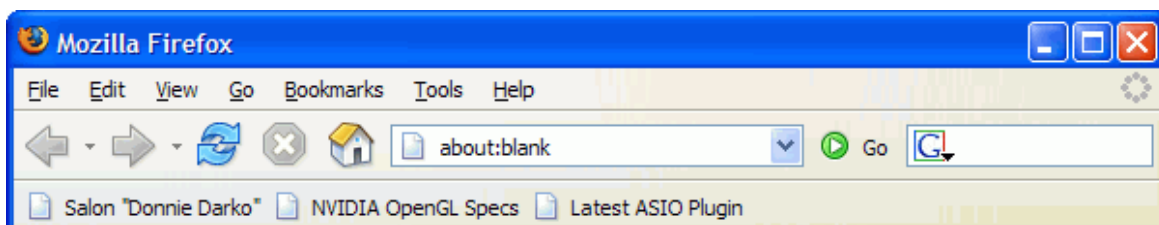
Just as Linux is to Windows, Mozilla's open-source Firefox is to Internet Explorer. The fact is, although IE is still the top used browser, Firefox is by far the best out there (Opera is fantastic, too, just not completely free).

Just to name a few reasons why Firefox is better than IE... [free download](#), small (4.7mb), built-in tabbed browsing, better web standards support, cross-platform, better security, highly customizable, open-source community = extensions for everyone.

So without further ado, we bring you our Firefox tweak guide. We will be covering every functionality aspect found in Firefox, from the very basic appearance and menu options, to advanced customizations in Firefox's behaviour such as caching, net connection handling, plug-ins and web page rendering. Also make sure to check our Firefox extensions top 10 list.

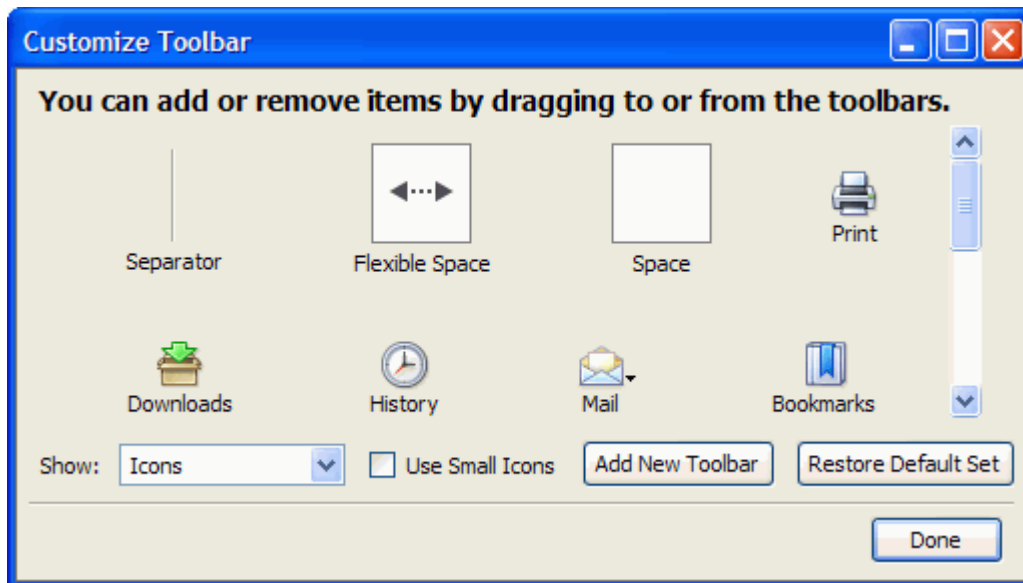
Toolbar Setup and Positioning

Like Internet Explorer, Firefox features highly customizable toolbars. These can be altered to better suit your needs and reduce clutter. By default Firefox will display something like this.



By selecting **View, Toolbars** you can select the toolbars you wish to have visible (**Selected**). The standard ones available are **Navigation Toolbar** and **Bookmark Toolbar**. Personally I disable (**Unselect**) both.

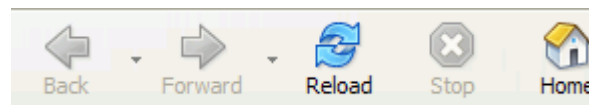
Now select **View, Toolbars, Customize**:



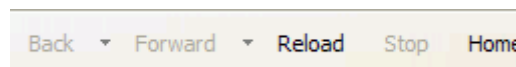
Toolbar buttons. The list here can be used to select buttons to **Add** to the toolbar. With the right-hand list you can re-arrange currently visible buttons or **Remove** unwanted ones.

Show. This option specifies how descriptive text for toolbar buttons is displayed:

- **Icons and Text.** This sets that descriptive text should be displayed beneath all toolbar buttons. Toolbar size is greatly increased when using this option.



- **Text.** This sets that descriptive text *only* is displayed. This provides a vertically smaller toolbar.



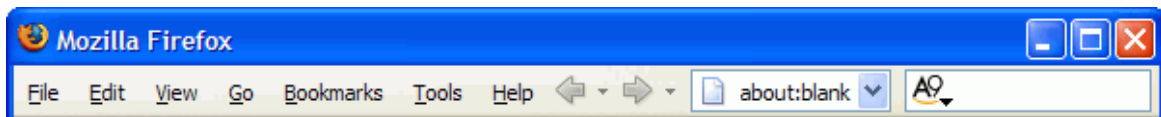
- **Icons.** This sets that no descriptive text is to be displayed. This provides the smallest toolbar and is recommended once you know all the buttons' functionality.



- **Use Small Icons.** Select this option to use smaller sized toolbar icons, as compared with the *default (Unselected)* versions. The advantages/disadvantages of each size should be apparent enough.

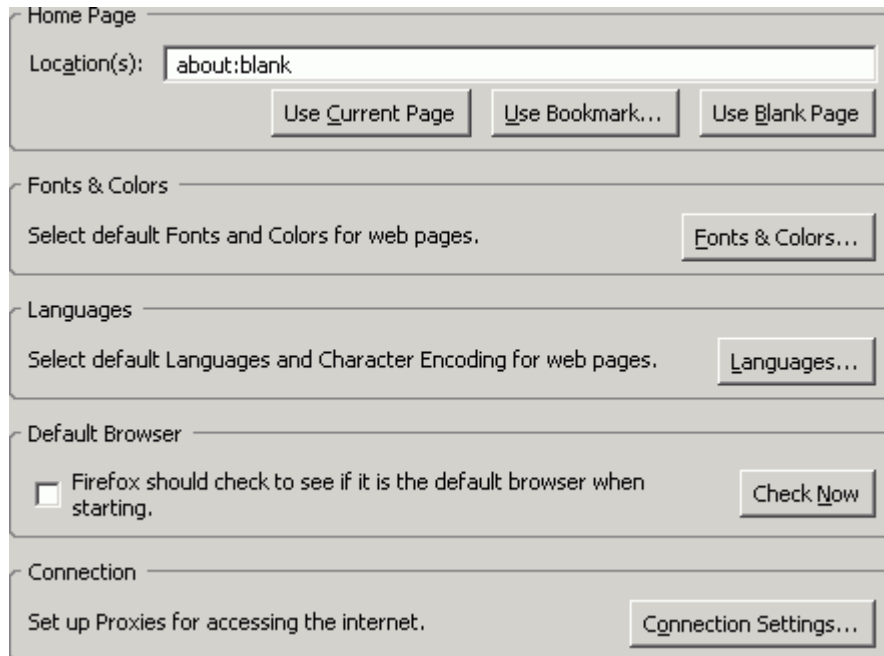


Pressing **Restore Default Set** will restore the *default* installation options for the toolbar, if necessary. Once finished, press the **Done** button. Pictured beneath is the current toolbar setup which I use, as you can see, it's a lot smaller than the initial setup and contains only the buttons which I need.



General Options

Load Firefox, click **Tools, Options**, and select the **General** tab.

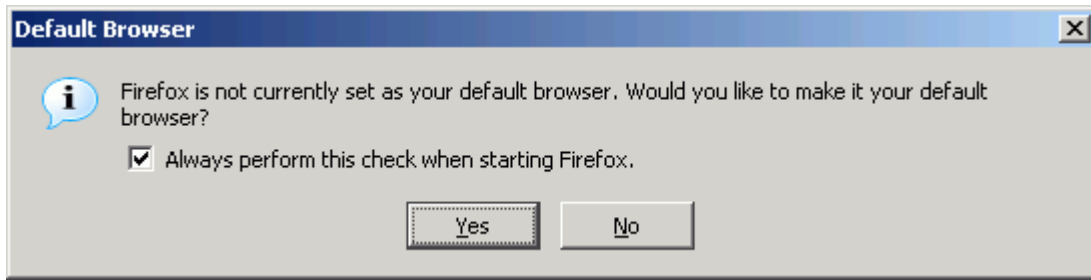


Home Page. This specifies Firefox's homepage. A specific address can be entered in the **Location** field if desired. Alternatively, pressing the **Use Current Page** button will set the currently loaded page as homepage. **Use Bookmark** allows you to select a link from your bookmarks as the homepage, while **Use Blank** specifies that a blank page be used as the homepage.

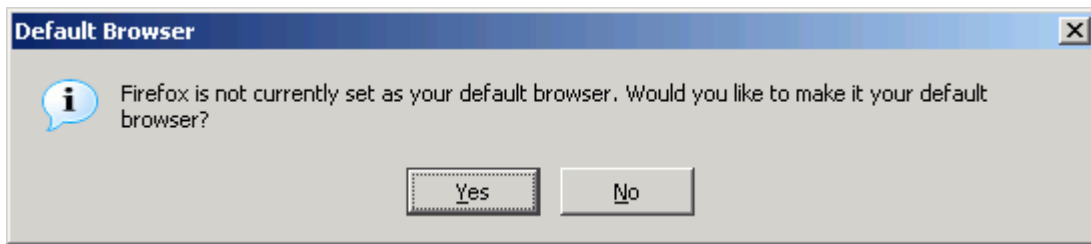
Fonts and Colors. Selecting this button allows you to adjust font and colour settings for webpages. This isn't something you'd want to adjust unless you have a specific need to do so (Perhaps due to eyesight problems say).

Languages. Selecting this button allows you to specify the default character encoding and priority for languages installed for webpages which are available in different languages. As with the previous option, you shouldn't adjust this unless you have a specific need to do so.

Default Browser. **Selecting** the option sets Firefox to check whether it is currently set as the *default* browser when you next launch it. If it isn't, the following window will appear.



Again, **Selecting** the option that appears here will have Firefox check whether it is set as the *default* browser when you next launch it. Selecting **Yes** will change Firefox to the *default* browser, whereas **No** will leave it unchanged. Alternatively you can also use the **Check Now** button to check if Firefox is the *default* browser, if it is you will be informed so, whereas if it isn't the following appears.

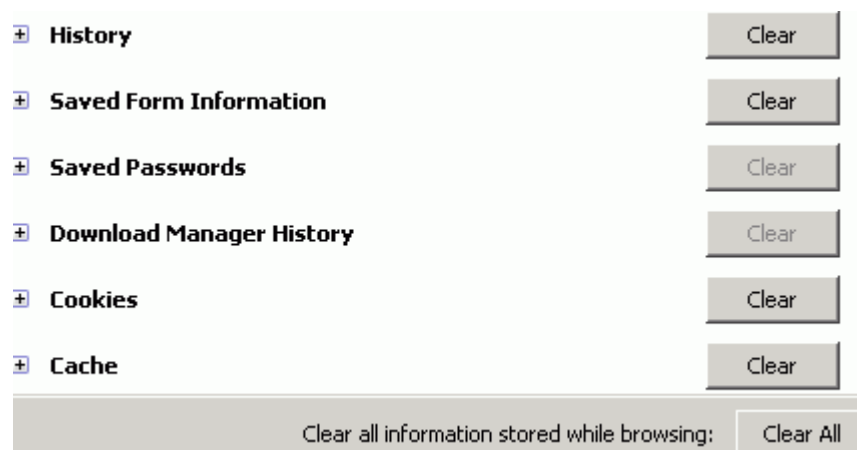


As before, selecting **Yes** will change Firefox to the *default* browser, whereas **No** will leave it unchanged.

Connection. Selecting the **Connection Settings** button allows you to configure Proxy server settings for Firefox to use. You'll need to consult your ISP or network documentation for specific settings if necessary. For most people out there though **Direct connection to the internet** should suffice.

Privacy Options

Now select the **Privacy** tab.



History. This field sets the number of days that links for pages viewed are stored for. This may be useful if you want to keep track of what others using the PC are viewing or if you want to be able to check what out a recently viewed webpage whose URL you've forgotten. If this doesn't seem like much use then set it to **0**.

Click the **Clear** button to delete the history files stored on the PC. This is presumably useful if other persons have access to your computer and you don't want them to know what webpages you've been visiting ;) History can be viewed in Firefox by clicking the **History** button.

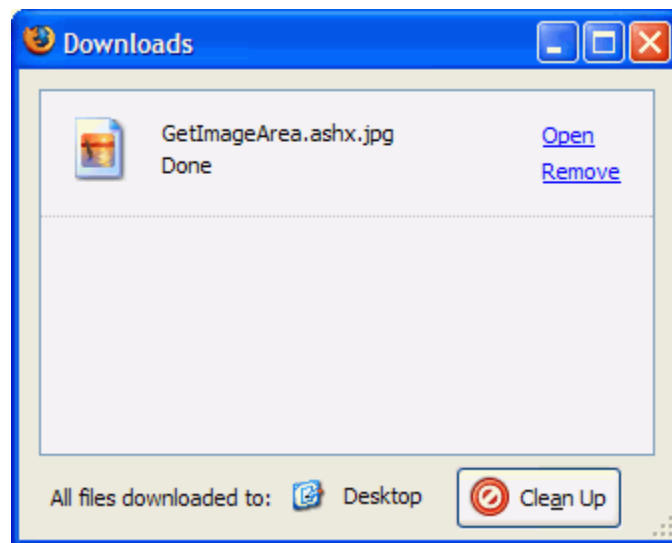
Saved Form Information. Select **Save information I enter in web page forms and the Search Bar** to do just that. This can prove a convenient time saver for filling out forms on webpages and such. Then again, if you value your privacy (especially if others have access to your system) then it may be best to **Unselect** this option.

Click the **Clear** button to delete stored form information.

Saved Passwords. Select **Remember Passwords** to *enable* saving of Usernames/Passwords. Similar to the previous setting, this will auto-fill login details for websites you wish to save login details for whenever prompted.

Download Manager History. This drop-down menu determines how the Download Manager history is cleared. Three options are available:

- **Upon successful download.** Self-explanatory really, this removes the item from the Download Manager when the download has finished.
- **When Firefox exits.** Again, fairly self explanatory, downloads are removed from the Download Manager whenever a Firefox session is closed.
- **Manually.** Selecting this options saves download history until items as removed manually, this can be done individually using the **Remove** button in the Downloads window, or by selecting the **Clean Up** button (or the **Clear** button in the **Privacy** tab).



Click the **Clear** button to delete stored download manager history information.

Cookies. Cookies are (generally) useful text files stored on your system that can be used to record information such as login details so you won't need re-login every time you visit certain sites. They can also be used for tracking and other purposes, which might not be beneficial to you. [Google's Privacy Policy](#) for example states the following regarding to cookies:

Upon your first visit to Google, a cookie is sent to your computer that uniquely identifies your browser... We use cookies to improve the quality of our service and to better understand how

people interact with us. Google does this by storing user preferences in cookies and by tracking user trends and patterns of how people search... However, some Google features or services may not function properly without cookies.

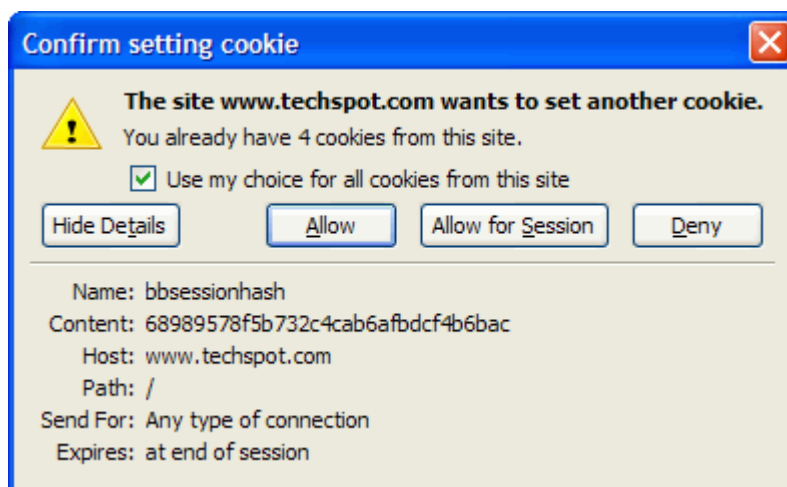
Further information can be found on cookies at sites such as [SpywareInfo](#). Firefox offers the following options with regard to cookie support:

Allow Sites to Set Cookies. Select this option to *enable* the creation of cookies by webpages that you view. This is recommended as most sites require cookie support in order to function correctly. The next few options can be used to lockdown on undesirable cookies getting onto your system.

For the originating web site only. Select this to accept cookies *only* originating directly from the webpage you are visiting. **Unselecting** this option *enables* cookies from both the originating website and other third-party cookies, while this wouldn't perhaps be the preferred choice the next 3 options can be used to further lockout undesired cookies entering/residing on your system.

Keep Cookies. This option allows you to specify how cookie storage is to be handled. Three options are available:

- **Until they expire.** Selecting this option sets that cookies are stored until they expire, as per their expiration date. This can vary greatly from site to site (and for what purpose). In my own case, of three cookies the expiration dates were 2006, 2010 and 2037. As a result selecting this option is most likely to lead to a whole heap of cookies stored on your PC until either you use the **Clear** button or start deleting them manually. That said, this option also ought to provide the most hassle free mode of operation.
- **Until I close Firefox.** This operation performs as above, accepting cookies without prompting you, albeit only until the Firefox session is closed. This offers a decent trade-off between convenience and privacy/cluttering. As you may have realised from earlier though, downsides may include things such as having to re-entering login information for sites each time you wish to enter them.
- **Ask me every time.** Selecting this option prompts you with the beneath whenever accepting a cookie is concerned.



Use my choice for all cookies from this site. Selecting this option sets Firefox to apply the option chosen (Allow/Allow for Session/Deny) for all further cookies originating from the site in question. This

would be recommended where you would visit a website regularly or incidentally receive a lot of prompts regarding their cookies and wish to specify how to handle all their cookies.

Show/Hide Details. Clicking this button allows you to view/hide details on the cookie that confirmation is currently being prompted for. This may help you decide whether to accept/deny it.

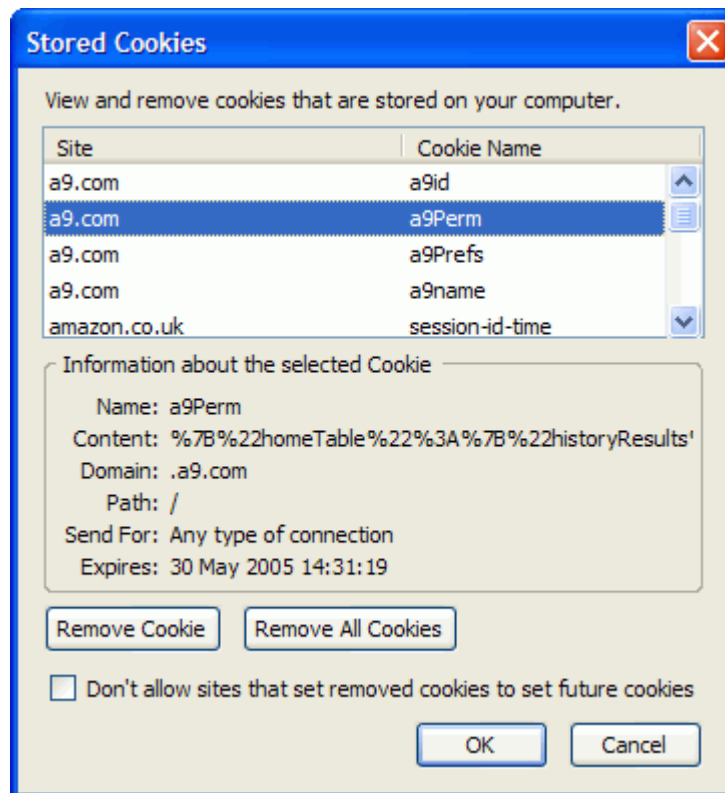
Allow. Select this button to *enable* the creation of a cookie from the site. This would be recommended where the cookie is benign in function, e.g. for storing forum login details so you won't need to login every time.

Allow for session. Select this button to *enable* the creation of a cookie from the site, but which lasts *only* for the current Firefox session.

Deny. Select this button to *disable* the creation of a cookie from the site

Exceptions. Clicking this button allows you to specify how to handle cookies from specific websites. Enter the website address into the **Address of web site** field and select **Block**, **Allow for session** or **Allow** as appropriate. Should you change your mind at a later time, you can remove individual sites using the **Remove Site** button or **Remove All Sites** if you wish to clear the entire listing.

View Cookies. Selecting this option allows you to view cookies currently stored on the system.



It would be wise to check this regularly if you've set **Keep Cookies until they expire** so you can view and remove any unnecessary ones. **Selecting 'Don't allow sites that set removed cookies to set future cookies'** allows you to toggle permanently blocking receiving cookies from sites you select **Remove Cookie** for (so make sure this is **Unselected** before you select **Remove All Cookies**). **Remove All Cookies** deletes all cookies stored.

Selecting the **Clear** button will delete all cookies currently stored by Firefox.

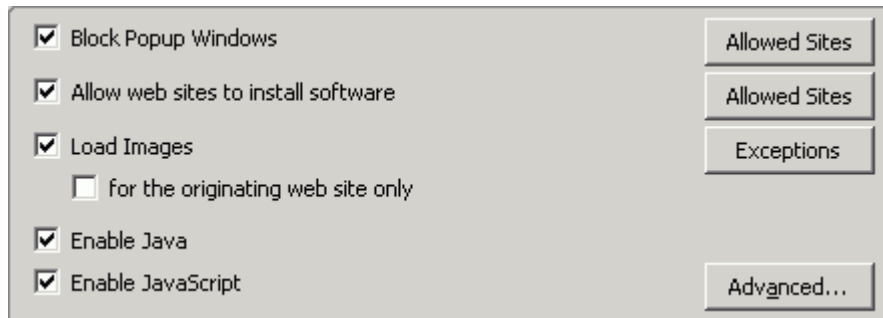
Cache. This field specifies the size, in KBs (1024KB = 1MB), of the cache folder. This is used to cache all data for the webpages you view. Should you visit certain websites repeatedly this cache can improve loading times as data for the webpage may already exist in the cache. That said, there's no point going nuts with this setting and setting aside a large amount of space – it's merely intended as a *temporary cache*.

Though this may vary depending on how many repeat viewing webpages you visit there should no real need to set aside more than one or two dozen megabytes for this. Personally I use just **10240**. As discussed later, Firefox also checks for newer versions of cached data so there is no need to store too much anyway as you'll probably be downloading newer versions regularly.

Selecting the **Clear** button deletes cached data (Though not cookies). You should consider doing this every once in a while to remove unneeded cached data (From a webpage you once visited once for example).

Web Features Options

Now select the **Web Features** tab.



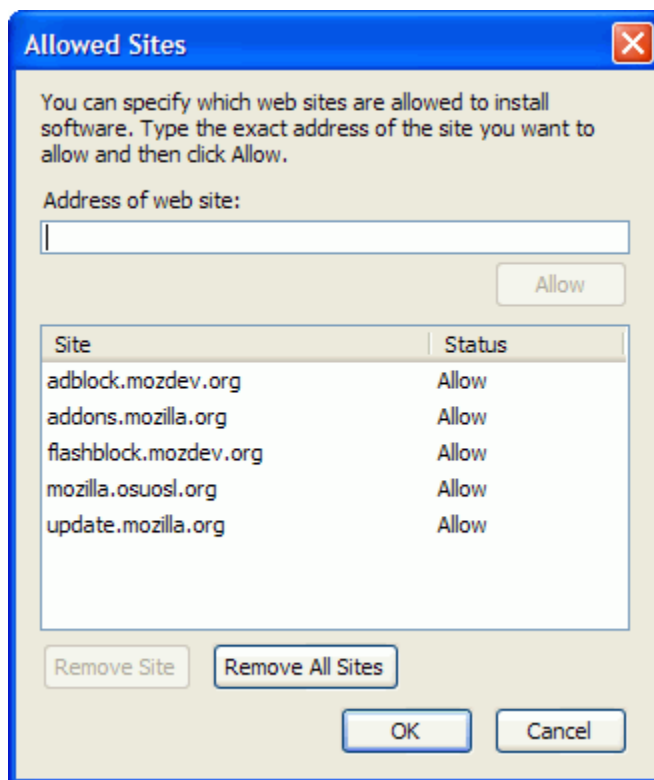
Block Popup Windows. Select this option to *enable* Firefox's built-in pop-up blocker, which *disables* the loading of much pop-up content on sites – which will mostly be advertisements (it's worth noting this is not perfect and will also *disable* many legitimate pop-ups). Should a pop-up be blocked in this way an information bar will appear at the top of the window, from which you can select what action to take.

Allowed Sites. To allow pop-up from specific websites click this button and enter the address into the **Address of web site** field and click **Allow**, e.g. I've entered pigsback.com as the pop-up blocker *disables* the displaying of pop-up windows for quizzes and such. Use the **Remove Site** button should you change your mind at a later date, or **Remove All Sites** if you wish to clear the entire listing.

Allow web sites to install software. **Selecting** this option *enables* sites to install software (e.g. extensions) and would be highly recommended for best functioning of Firefox. Unlike Internet Explorer Firefox is more secure as regards installing software, what with if you aren't installing software from a trusted source, i.e. update.mozilla.org, then you'll be prompted as to whether to add the site to the **Allowed Sites** list, after which you'll be prompted whether or not you wish to allow installation (though this has been exploited, and since, fixed [recently](#)). **Unselecting** this option *disables* sites from installing software, which might prove useful should you be interested in locking down Firefox.

Allowed Sites. To allow software to be installed from specific sites click this button and enter the address into the **Address of web site** field and click **Allow**, e.g. I've entered flashblock.mozdev.org, so as to

allow installation of Flashblock. Use the **Remove Site** button should you change your mind at a later date or **Remove All Sites** if you wish to clear the entire listing.



Load Images. **Unselecting** this option *disables* the displaying of images in web pages. This can enhance loading speed of webpages as images won't be downloaded but as you're probably well aware, images form an integral part of many sites now and could severely restrict navigation (If you're still interested you can always **right click** over an image placeholder and select **View Image** to see it). **Select** this option to *enable* the download/display of images in webpages (recommended).

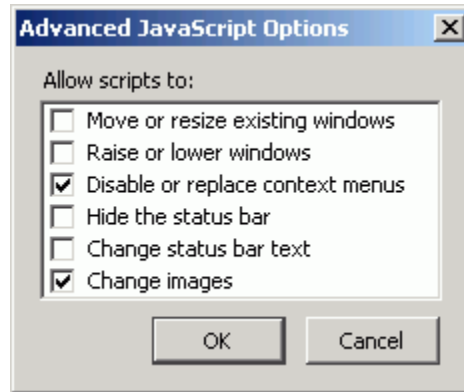
For the originating web site only. **Selecting** this option *disables* the loading of images that do not originate from the website currently loaded.

Exceptions. To *enable/disable* images being loaded from specific sites regardless of the previous 2 settings, click this button and enter the address into the **Address of web site** field and click **Allow** or **Block** as desired. Use the **Remove Site** button should you change your mind at a later date or **Remove All Sites** if you wish to clear the entire listing.

Enable Java. **Selecting** this option *enables* support for Sun MicroSystem's [Java](#) in Firefox. If you haven't got this installed then obviously you may leave this **Unselected**, or if you merely do not want any Java content loaded on web pages. Personally I'd recommend downloading the Sun's Java VM and *enabling* this feature (this can also be used to provide Internet Explorer with a much better Java VM too).

Enable JavaScript. JavaScript is an open, cross-platform object scripting language (Not to be confused with Sun MicroSystem's [Java](#)) and has proven repeatedly to be a security risk with several (now fixed) vulnerabilities attributed to it (e.g. [firetabbing](#) and [code execution via JavaScript](#)). **Unselecting** this option provides improved security, although you will likely find many webpages that don't function correctly as a result. **Select** this setting should you desire support for JavaScript.

Advanced. In the event you *enabled* JavaScript support, further options are available by clicking this button.



Move or resize existing windows. This option controls whether JavaScript can be used to move and/or resize windows, whereby **Selecting** this *enables* scripts to do this. It would perhaps be best to **unselect** this, allowing only yourself to resize/move windows.

Raise or lower windows. This option controls whether JavaScript may be used to bring windows into the foreground/background via `focus()`. **Unselecting** this *disables* such actions, which won't affect new popups from loading in the foreground, though can force existing ones to remain in the background unless switched to manually. **Select** this to allow the script to determine what happens.

Disable or replace context menus. This option controls whether JavaScript can be used to alter, or even disable the context menu, e.g. right clicking could be disabled on certain pages. **Select** this if you wish to allow sites to be able to do this, while **unselecting** it ensures scripts can't be used to alter this functionality.

Hide the status bar. This option controls whether JavaScript can be used to hide the status bar, whereby **Selecting** this *enables* scripts to do this. **Unselect** this option to force the status bar to be displayed at all times.

Change status bar text. This option controls whether JavaScript can be used to display custom text in the status bar, e.g. moving the mouse over a hyperlink normally would display where the link points to, though a script could be used to display something else instead. **Select** this to allow such custom status bar text displayed, while **unselecting** it will *disable* hiding the status bar in this way.

Change images. This option determines whether JavaScript is allowed to changes images. **Select** this to *enable* the feature, or **unselect** to *disable* it (recommended).

Accessibility Options

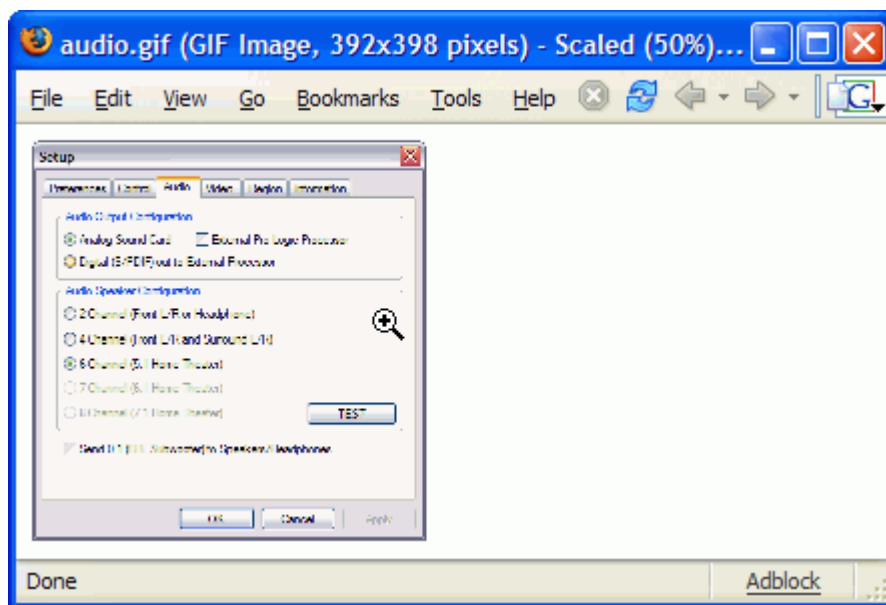
Now select the **Accessibility** tab.



Move system caret with focus/selection changes. This setting is only really needed if you are using Accessibility Aids. If you use such aids then **select** this setting, otherwise leave it **Unselected**.

Begin finding when you begin typing. Selecting this option *enables* you to **Find in this page** by typing letters, eliminating the need to manually start **Find in this page** (e.g. by pressing Ctrl + F). **Unselecting** this *disables* such automatic initialisation, which you might find to prove less troublesome (as searches will only occur should you want them to).

Resize large images to fit in the browser window. Selecting this option *enables* Firefox to scale images that don't fit into the current window. The image beneath illustrates this in action.



Such scaled images can be viewed at their original size by moving the cursor over them, which changes to a magnify glass and clicking on it. Clicking again will rescale it once more. **Unselecting** this option *disables* such scaling from occurring.

Use autoscrolling. Autoscrolling is the function whereby if you click/depress the middle mouse button (Scroll wheel more commonly) you can scroll up/down the page by moving the mouse up or down. You can determine if it's active by the following icon appearing:



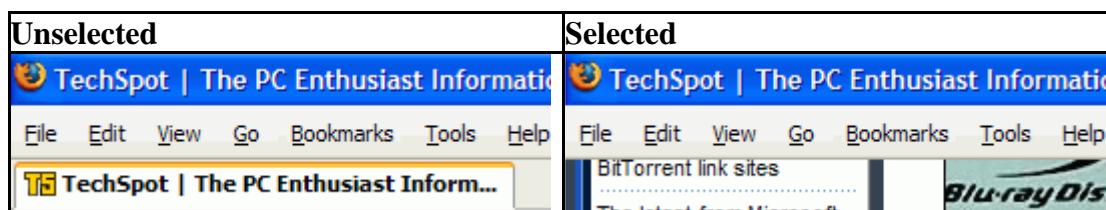
If you wish to *enable* Autoscrolling then **Select** this option, otherwise **unselect** it to *disable* it.

Use smooth scrolling. **Select** this option to *enable* smooth scrolling of webpages, whereby Firefox slides smoothly up/down a page rather than jerking up/down (**Unselected**).

Open links from other application in. This option specifies how Firefox opens links launched externally, e.g. e-mail or Microsoft Word. Three options are available:

- **a new window.** Selecting this option launches a new Firefox window to view the link. This would perhaps be the safest option to select (if you launch something questionable you may be able to end the process without affecting other windows).
- **a new tab in the most recent window.** Selecting this option launches a new Firefox tab in the most recently active Firefox window to view the link. This would be recommended if you don't wish launching such links to affect your most recently active webpage and you aren't concerned about the links content.
- **the most recent window/tab.** Selecting this option launches the link in the most recently active Firefox window/tab. This would be recommended if you aren't concerned about launching links in the most recently active window/tab (you can always use the back button to view the previous page if you need).

Hide the tab bar when only one web site is open. **Selecting** this option *enables* Firefox to hide the tab bar when only a single page is open in Firefox, which makes perfect sense – if only 1 page is open then there's no need for tabs in the first place and it *increases* viewing area slightly. **Unselecting** this setting forces Firefox to display tabs even when only a single page is being viewed in a window. The images beneath illustrate this in effect.

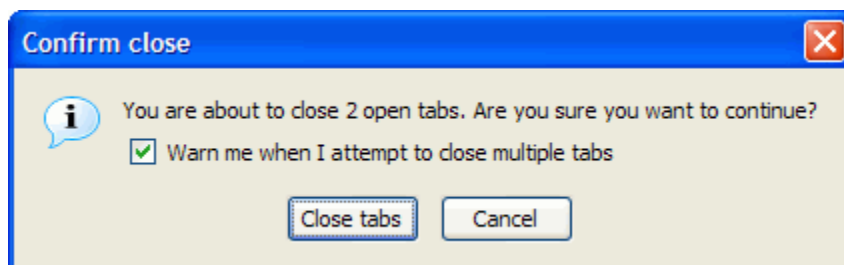


As can be seen, **Selecting** this option allows for more site content to be visible than otherwise would be when tabs are hidden for single page windows.

Select new tabs opened from links. **Selecting** this option specifies that new tabs launched from links in pages are to be loaded in the foreground, i.e. **right clicking** on a link and selecting **Open Link in New Tab** launches the link in a new tab and switches you to that tab. **Unselecting** this option launches new tabs in the background, i.e. **right clicking** on a link and selecting **Open Link in New Tab** launches the link in a new tab but you remain in the current one. While this is a matter of preference, for most it's probably best to leave this **Unselected** as more often than not you'll probably want to open more than one tab from a page.

Select new tabs opened from bookmarks or history. This option performs exactly as per the previous option, albeit it relates to launching bookmarks/history items in tabs, i.e. **Selecting** this option specifies that new tabs launched from bookmarks/history items are to be loaded in the foreground, while **Unselecting** this option specifies that new tabs launched from bookmarks/history items are to be loaded in the background.

Warn when closing multiple tabs. **Selecting** this option forces Firefox to confirm closure of a window if it contains multiple tabs.



This may be useful as you may accidentally choose to close such windows, forgetting there were other active tabs you wished to finish viewing also. If you do wish to proceed closing the window select **Close tabs**, otherwise select **Cancel**. **Selecting Warn me when I attempt to close multiple tabs** leaves this confirmation request in place for the future.

Software Update. Firefox features a built-in update scanner, 3 options are available:

- **Firefox.** **Select** this option to *enable* Firefox to check at regular intervals for updated versions of Firefox. Of late the main Firefox releases have been due to security issues and as such it would be recommended to leave this **Selected**.
- **My Extensions and Themes.** **Selecting** this option *enables* Firefox to check at regular intervals for updated versions of installed Extensions and Themes.
- **Check Now.** Pressing this button initiates an update scan should you not wish to wait for the scheduled scans.

Use SSL 2.0, SSL 3.0 and TLS 1.0. **Selecting** these options *enables* support for **SSL (Secure Sockets Layer) 2.0/3.0**. This is used to provide secure access to many websites (SSL 2 is supported by all secure websites), with digital certificates issued by certifying authorities to the sites to allow their identity to be verified. SSL 3.0 is an updated version of SSL 2.0 and provided new ciphers and encoding schemes (and more). **TLS (Transport Layer Security) 1.0** is essentially an updated version of SSL 3.0. These can be viewed by double clicking the gold lock that displays in Firefox when accessing a secure site, e.g. PayPal.



Ensure these are **Selected** at all times.

Plugins/Extensions

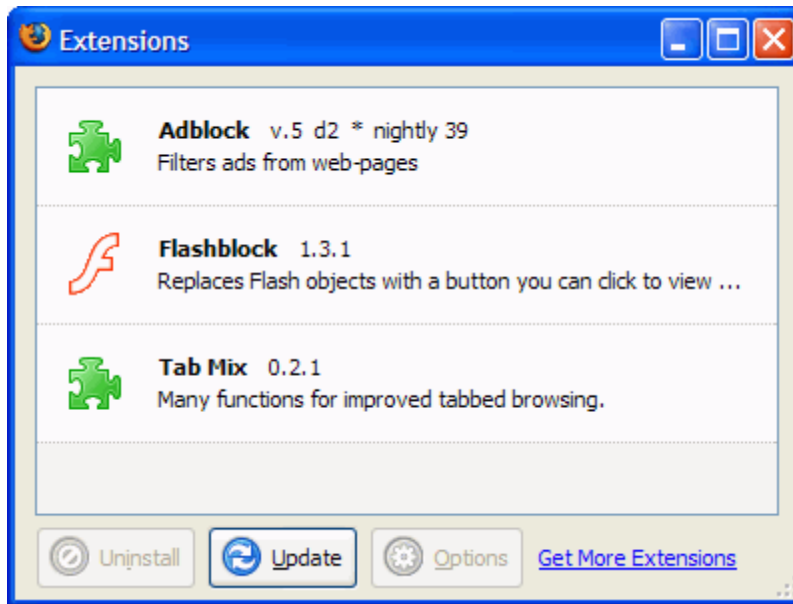
Firefox can be further enhanced by installing new Extensions, small add-ons that add new functionality, and plug-ins, which allow websites to provide content to you and have it appear in your browser. The possibilities available by using these are immensely varied. The [GetRight](#) download manager can install a plugin (NPGetRt.dll) that allows it to monitor clicks and catch any downloads in Firefox. Not happy with the search engines included by *default*? Go to [Mycroft](#) and download additional search engine Plugins.

For starters you may wish to consider some of the following extensions (not ten but eleven actually), suggested by myself and other TechSpot staff members:

- [Flashblock](#). *Flashblock is an extension for the Mozilla and Firefox browsers that takes a pessimistic approach to dealing with Macromedia Flash content on a webpage and blocks ALL Flash content from loading. It then leaves a placeholder on the page that allows you to click to view the Flash content.*
- [Netcraft Toolbar](#). *The Toolbar community is effectively a giant neighbourhood watch scheme, empowering the most alert and most expert members to defend everyone within the community against phishing frauds. Once the first recipients of a phishing mail have reported the target URL, it is blocked for community members as they subsequently access the URL. Widely disseminated attacks (people constructing phishing attacks send literally millions of electronic mails in the expectation that some will reach customers of the bank) simply mean that the phishing attack will be reported and blocked sooner.*
- [Adblock](#). *Adblock is a content filtering plug-in for the Mozilla and Firebird browsers. It is both more robust and more precise than the built-in image blocker. Adblock allows the user to specify filters, which remove unwanted content based on the source-address.*
- [SessionSaver](#). *SessionSaver restores your browser -exactly- as you left it, every startup, every time. Not even a crash will phase it. Windows, tabs, cookies + scrolling -- they're all saved.*
- [Mouse Gestures](#). *Mouse Gestures are a fast way to execute commands without using the keyboard, menus or toolbars. Instead, the user holds down a mouse button (usually the right one), moves the mouse in a certain way to form a gesture, and then releases the mouse button. In web browsers like Mozilla Firefox or the Mozilla Suite, gestures are used to go back or forward a page, switch between tabs, open multiple links at once, control text or image size, and numerous other functions.*
- [ForecastFox](#). *Forecastfox is an extension that brings international weather from AccuWeather to your Firefox web browser. You can use your US zip code or you can easily find your town or city by using the built in search tool. It is unobtrusive and lightweight, yet can be customized to display nearly as much data as the leading desktop weather programs.*
- [Web Developer](#). *The Web Developer extension for Mozilla Firefox and Mozilla adds a menu and a toolbar to the browser with various web developer tools.*
- [FoxyTunes](#). *FoxyTunes allows you to control your favorite media player without leaving the browser. The controls are positioned on the status bar or one of the toolbars, so no extra space is wasted. FoxyTunes has several nice features. You can control the playback, adjust the volume, see what's playing, and more.*
- [About This Site](#). *About this site is a Firefox extension by Gina Trapani which gives you 1-click access to various services that provide information about the web page you are viewing.*
- [Tab Mix](#). *Tab Mix is an extension for the Mozilla Firefox browser which enhances its tab browsing capabilities. Tab Mix's main objectives are to be small, fast, easy to use, logical, and without bugs while containing the most natural features related to tab browsing.*
- [Greasemonkey](#). *Greasemonkey lets you add bits of DHTML ("user scripts") to any web page to change its behavior. In much the same way that user CSS lets you take control of a web page's style, user scripts let you easily control any aspect of a web page's design or interaction.*

Beyond those suggestions, the main listing of Firefox plugins/extensions can be [found here](#). So be sure to check it out, as there's bound to be *something* there that'll interest you.

To view installed extensions, load Firefox and click **Tools, Extensions**.



Selecting an extension allows you to **uninstall** it, use the **Update** button to check for, and install, any available updates, while the **Options** button allows you to adjust any options available for the extension.

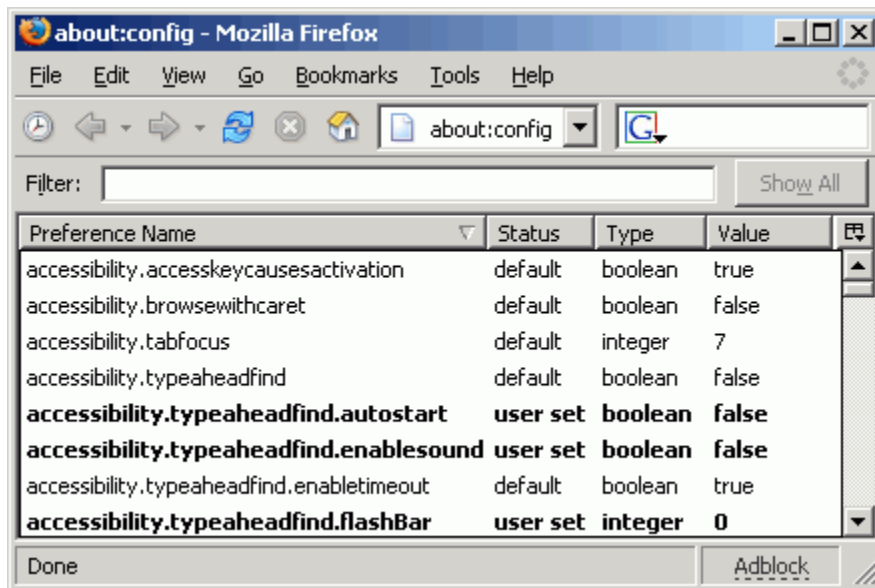
Themes

If you're tired of how Firefox looks, that's another problem easily solved – just download a new theme. The main listing of Themes for Firefox can be [found here](#). To view installed themes, load Firefox and click **Tools, Themes**.

Selecting a theme allows you to **uninstall** it, use the **Update** button to check for, and install, any available updates, while the **Use Theme** button allows you to switch to that theme.

Advanced Configuration

Firefox's **Options** menu provides access to a decent selection of options, though there are more (much more) available. Perhaps the simplest method to access and modify these is to type **about:config** into the Firefox Address bar and press **Enter**. This displays a nearly complete list of preferences available.



The **Filter** field can be used to search for entries which contain the keyword entered, e.g. cache, while pressing **Show All** will display all available preferences. Modifying the **values** for preferences is achieved by **right clicking** on a preference and selecting **Toggle** (for **true/false** type options) or **Modify** (for options which accept other values/text) as appropriate. In the event the preference you are searching for is not listed, it can be created by **right clicking** and selecting **New** then **String**, **Integer** or **Boolean** as necessary.

Changes to preferences displayed here are indicated in **bold** and stored in the **prefs.js** file (located in the **Profile** directory). This file can be opened with any text editor and should appear similar to the shown beneath:

```
# Mozilla User Preferences
/* Do not edit this file.
 *
 * If you make changes to this file while the application is running,
 * the changes will be overwritten when the application exits.
 *
 * To make a manual change to preferences, you can visit the URL about:config
 * For more information, see http://www.mozilla.org/unix/customizing.html#prefs
 */
user_pref("accessibility.typeaheadfind.autostart", false);
user_pref("accessibility.typeaheadfind.enablesound", false);
user_pref("accessibility.typeaheadfind.flashBar", 0);
```

As per the warning, there's no need to edit this file if you intend to solely use **about:config** (and if it does become necessary for whatever reason, close all Firefox windows before doing so, otherwise changes made may be overwritten).

Alternatively, you can store custom preferences in the **user.js** file, this file must be created manually. A simple way to do this is to copy the **prefs.js** file, rename it to **user.js** and delete whatever is currently listed in it. Preferences entered here are copied into the **prefs.js** file whenever Firefox is loaded.

Preferences are entered in the format as the **prefs.js** file, i.e. **user_pref("preference", value)**; For example **user_pref("accessibility.typeaheadfind.autostart", false)**; If you're feeling fancy you can even add comments by entering **//** at the start of a line, like the following:

```
//Controls caching of image/chrome data into RAM
//Default: true
user_pref("browser.cache.memory.enable", false);
```

It's important to note that if you delete any preferences that were stored in the **user.js** file, they will not be changed in the **prefs.js** file, e.g. if you had set **user_pref("accessibility.typeaheadfind.autostart", false)**; and delete it from **user.js** several days later, that value will remain the same in **prefs.js**.

Regardless of which method you choose to use, some of the more relevant/interesting preferences available are as follows (divided into appropriate sections).

Note – For the purposes of creating entries the type (**String**, **Integer** or **Boolean**) is included in parenthesis after the preference name.

Auto-Updates

app.update.autoUpdateEnabled. (Boolean) Setting this option to **true** *enables* Firefox to check at regular intervals for updated versions of Firefox. Of late the main Firefox releases have been due to security issues and as such it would be recommended to leave this set to **true**. If you don't require this functionality for some reason, set this to **false**. Note – This is the same as the **Software Update, Firefox** option in the **Advanced** tab.

app.update.url. (String) The entry here specifies where to check for updated extensions. You shouldn't need to alter this.

app.update.interval. (Integer) The value entered here specifies the amount of time, in milliseconds, that Firefox waits before checking for updates. The *default* being **86400000**, aka 1 day. Of late the main Firefox releases have been due to security issues and as such it would be wise to leave this at the *default* to ensure you get notified of such updates ASAP.

extensions.update.autoUpdateEnabled. (Boolean) Setting this to **true** *enables* checking for updated version of the extensions/themes. Set this to **false** to *disable* such checks, this isn't recommended. Note – This is the same as the **Software Update, #my Extensions and Themes** option in the **Advanced** tab.

extensions.update.url. (String) The entry here specifies where to check for updated extensions. You shouldn't need to alter this.

extensions.update.interval. (Integer) The value entered here specifies the amount of time, in milliseconds, that Firefox waits before checking for updates. The *default* is **60480000** (7 days). This is a sufficient value as extension updates aren't likely to be of critical importance, though feel free to *increase/decrease* this if necessary.

Bookmarks

browser.chrome.favicons. (Boolean) Many websites feature icons that can be displayed in the address bar and for any bookmarks created. Setting this to **true** *enables* such icons to be displayed, which you might want to do as they add some variety, e.g. the Wikipedia and TechSpot forum bookmarks shown beneath use such icons, whereas the ASIO and NVIDIA bookmarks use the *default* icon as they lack a custom one.



Setting this to **false** will *disable* such icons, which can result in smaller bookmarks and minor bandwidth savings as the icons will not be downloaded.

browser.chrome.site_icons. (Boolean) This performs a similar function to **browser.chrome.favicons** and should be set to the same value as that is.

browser.tabs.loadBookmarksInBackground. (Boolean) Setting this option to **true** specifies that new tabs launched from bookmarks/history are to be loaded in the foreground, while **false** specifies that such new tabs in the background. Note – This is the same as the **Select new tabs opened from bookmarks or history** option in **Advanced** options.

Caching

browser.cache.disk.parent_directory. (String) This entry allows you to specify a location for the cache directory (The *default* location will be in the **Profiles** directory). If entering this location in **about:config** use the standard path format, e.g. **C:\Windows\Temp**, whereas if entering this into the **User.js** file via text editor *double* \ are required in the path, e.g. **C:\\Windows\\Temp**.

browser.cache.disk_cache_ssl. (Boolean) When set to **false** secure webpages (https://, etc.) are not saved in the cache folder. This is recommended so as to eliminate the possibility of others gaining access to important details by checking that folder. If you aren't worried about that then setting this to **true** will allow such data to be saved in the cache folder.

browser.cache.disk.capacity. (Integer) The value entered here specifies the size, in KB, of the browser's hard drive cache. This improves the rendering speed of a webpage as data (images, etc.) can be loaded directly from the hard drive as opposed to needing to be downloaded. On the negative side, this can potentially mean old data is loaded rather than newer versions, which can be countered with the **browser.cache.check_doc_frequency** setting. A dozen to a couple dozen megabytes (1024KB = 1MB) should prove sufficient for the browser cache, there's no need to go excessive with it, though if you browse a lot of sites daily then it may be more worthwhile. Personally I use **10240** (10MB).

browser.cache.check_doc_frequency. (Integer) The value entered specifies how often newer versions of cached files are checked for. **0** checks once per session. **1** checks each time the webpage is loaded. **2** *disables* checking for newer versions if a cached version already exists. While **3** checks automatically (Whenever it's out of date).

browser.cache.memory.enable. (**Boolean**) When set to **true**, this *enables* Firefox to cache data into RAM (This cache is not equivalent to the hard drive cache and stores images and chrome). The benefit of this is that RAM is much faster than the hard drive so loading of cached data should be further improved. If you're low on RAM or have better things for your RAM to be doing than caching browser data, e.g. simultaneously running other applications, then set this to **false**. If you do intend to *enable* this use **browser.cache.memory.capacity** to limit the RAM available for this.

browser.cache.memory.capacity. (**Integer**) The value entered here specifies the amount of RAM, in KB, to allocate to the Firefox memory cache. Setting this to **-1** (*default*) allows Firefox to determine the value based on RAM. Before considering adjusting this it's worth browsing as you normally would in a session, then enter **about:cache** into Firefox's address bar and hit Enter to display caching info - something similar to the beneath should be displayed.

Memory cache device

| | |
|------------------------------|-----------|
| Number of entries: | 44 |
| Maximum storage size: | 21504 KiB |
| Storage in use: | 706 KiB |
| Inactive storage: | 696 KiB |

Selecting **List Cache Entries** allows you to view what's currently being stored in the memory cache, along with the **Fetch count** (how many times an item has been retrieved from the cache).

Generally speaking, if you visit multiple image intensive sites during a session then a *large* memory cache can be of benefit as data can be quickly retrieved from RAM during Back/Forward operations, or if merely viewing multiple pages on a site (as the image content of the site, e.g. logos, would already be available in RAM) – this can be gathered from higher **Fetch counts**. In such cases you should compare the **Storage in use** value with the **Maximum storage size**, where if both are about the same then you should try *increasing* the value.

The opposite can also be the case. If you find that the **Fetch count** for images is **1** for the vast majority of items (don't include chrome entries in this) and/or the **Storage in use** value is significantly lower than the **Maximum storage size** then you may wish to *lower* the value, so that RAM isn't wasted caching items which have only been required once. This might also be the case if you have a lot of RAM but merely don't want Firefox to use too much RAM for this purpose. Again, bear in mind chrome items are also being cached, so if at all possible it's always worthwhile to set aside a few hundred KB for this purpose.

browser.sessionhistory.max_viewers. (**Integer**) This value specifies the number of content viewers cached for faster (instantaneous) Back/Forward operations. The *default* being **0**. A recommended value is **5**. However, this feature is currently experimental (and not available in all builds), if you encounter problems with it set it to **0** or even delete the setting itself.

browser.sessionhistory.max_entries. (**Integer**) The value entered here specifies the *maximum* number of history entries displayed for Back/Forwards operations, the *default* being **50**. Note this has no effect on the number of entries stored in History itself, merely how many are available for Back/Forwards operations in a window. There's no real need to change this value, but if you must, alter it to a value based on how many pages you're likely to view consecutively in one window and/or by how many pages you're likely to skip back/forwards in such cases (more than likely it'll be *less* than the *default*). Personally I use **10**.

Connection

network.http.keep-alive. (Boolean) Setting this to **true** *enables* the use of persistent connections. This can provide more efficient connections with legacy servers as it streamlines the process, e.g. fewer TCP connections/packets and pipelining support. In the event of problems arising with legacy servers set this to **false**. HTTP/1.1 supports persistent connections by *default*.

network.http.max-persistent-connections-per-server. (Integer) The value entered here determines that *maximum* number of persistent connections to a server, with valid values being **1 – 255**, the *default* being **2** as recommended in the HTTP/1.1 specification.

network.http.proxy.keep-alive. (Boolean) Setting this to **true** *enables* the use of persistent connections over proxy servers. This can provide more efficient connections with legacy servers as it streamlines the process, e.g. fewer TCP connections/packets and pipelining support. In the event of problems arising with legacy servers set this to **false**. HTTP/1.1 supports persistent connections by *default*.

network.http.max-persistent-connections-per-proxy. (Integer) The value entered here determines that *maximum* number of persistent connections to a proxy server, with valid values being **1 – 255**, the *default* being **4** as recommended in the HTTP/1.1 specification.

network.http.pipelining. (Boolean) HTTP/1.1 pipelining is a feature that allows multiple HTTP requests to be issued simultaneously. Without pipelining, HTTP requests are issued sequentially, with each further request not being issued until the previous has been completed. While this may seem like a more optimal approach, it's not without its [detractors](#). Firefox also contains an internal listing of servers that don't support pipelining and *disables* it for those to avoid problems. As such, you might find some experimentation worthwhile with this feature rather than just assuming it's always beneficial. As stated on W3C's [Network Performance Effects of HTTP/1.1, CSS1, and PNG](#):

HTTP/1.1 implemented with pipelining outperformed HTTP/1.0, even when the HTTP/1.0 implementation uses multiple connections in parallel, under all circumstances tested. In terms of packets transmitted, the savings are typically at least a factor of 2, and often much more, for our tests. Elapsed time improvement is less dramatic, but significant.

Set this to **true** to *enable* pipelining, **false** to *disable*. Further information can be found at [Mozilla.org's Pipelining FAQ](#).

network.http.proxy.pipelining. (Boolean) This feature performs exactly as per the above (**true** to *enable*, **false** to *disable*), albeit it applies to connections over a proxy. As before, this is a feature you may want to experiment with to determine its benefits, if any.

network.http.pipelining.maxrequests. (Integer) This specifies the *maximum* number of requests in a pipeline, the *default* being **4**. If you've set both **network.http.pipelining** and **network.http.proxy.pipelining** to **false**, this option can be ignored. The [Pipelining FAQ](#) makes the following point on the matter, *pipelining many requests can be costly if the connection closes prematurely because we would have wasted time writing requests to the network, only to have to repeat them on a new connection. Moreover, a longer pipeline can actually cause user-perceived delays if earlier requests take a long time to complete*. So while in general it would make sense to set a *higher* value (limit being **8**), it may also be something you should experiment with to determine an optimal value.

network.http.max-connections. (Integer) The value entered here specifies the *maximum* number of HTTP connections Firefox can make simultaneously, the *default* being **24**, with valid values being **1 – 65535**. On high bandwidth connections *increasing* this value can prove effective in reducing the amount of time required to download multiple pages, e.g. If you open multiple tabs at once/in short succession, as it allows Firefox to create more connections. That said, setting this *too* high can have adverse affects also. As such, a decent *maximum* value for broadband users would be **48**.

network.http.max-connections-per-server. (Integer) The value entered here specifies the *maximum* number of simultaneous connections to a server, the *default* being **8**, with valid values being **1 – 255**. *Increasing* this value can improve the loading times of certain sites as more data can be fetched at once. This is of most potential benefit to those with broadband connections, who will have the bandwidth to take advantage of this. Similar to the previous option, a decent *maximum* to set this to would be double the *default*, i.e. **16**. Bear in mind however that the more connections you are tying up, the less that will be available to others wishing to connect to the same server – so don't set this excessively high just because you can.

network.prefetch-next. (Boolean) Mozilla provides support for a “neat” feature called Link Prefetching, *a browser mechanism, which utilizes browser idle time to download or prefetch documents that the user might visit in the near future*. This is a [cause of concern](#) for many, who aren't happy with their browser downloading content from sites which they may not even view anyway. Despite what you may have read elsewhere, there's no need to get *too* hysterical about this. The most notable site which uses it currently is [Google](#) and where it does (not all searches do) it only uses it for the top search result, e.g. searching for **Firefox** yielded [Firefox - Rediscover the web](#) as the top result, and that's *only* one to use the **rel="prefetch"** tag for prefetching. Of perhaps more concern is that prefetching occurs during [Firefox idle time](#), e.g. you could have an Internet Explorer download active, but if Firefox is idle, it will still initiate prefetching. Further details are available [Mozilla.org's Link Prefetching FAQ](#). Set this to **true** to *enable* link prefetching for potentially faster loading times of prefetched sites, while if you're more privacy/security conscious set this to **false** instead to ensure only content for sites you intended to view is ever downloaded.

network.http.redirection-limit. (Integer) The value entered here specifies the maximum number of redirects that can be attempted consecutively, e.g. after replying in a forum you may be redirected to your response. As noted in [Bug 153888](#) the *default* value has *been increased from 10 to 20 (on account of nytimes.com)*. While you may prefer to *decrease* this value to avoid exploitation by certain sites, it may be necessary to *increase* the value in order for some to work. If this is the case an error page stating Redirection limit for this URL exceeded ought to appear.

network.http.sendRefererHeader. (Integer) The value entered here controls the transmission of the Referer HTTP header. This Referer header states the URL from which the browser was referred, e.g. if you click a link in this guide to another website, then techspot.com would be stated as the referrer, etc. This can be useful for sites as they can determine where traffic generates from, e.g. Google. Where no URL exists as the referral source, e.g. typing in the address, none is sent. Three options are available.

0. This *disables* the sending of the Referer header, which yields a minimal bandwidth saving. This is perhaps more so useful if you're privacy conscious and simply don't want it to be known where you were linked from.

1. This specifies that the Referer header is only sent for clicked hyperlinks.

2. This specifies that the Referer header is sent for clicked hyperlinks and images.

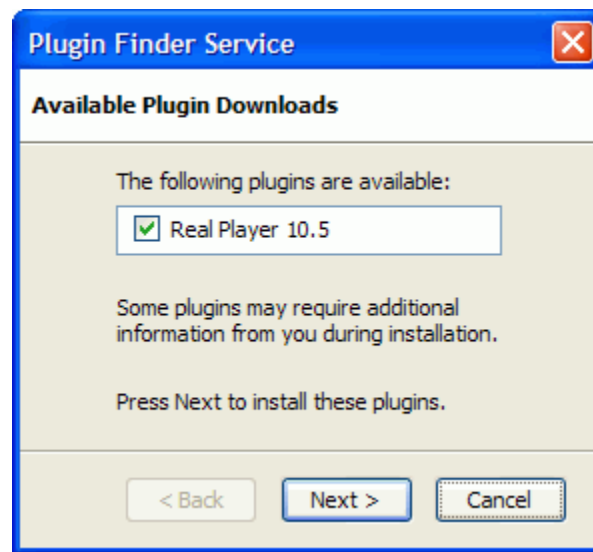
network.http.sendSecureXSiteReferrer. (Boolean) This setting controls the transmission of the Referer HTTP header between secure sites. It'd be recommended to set this to **true** as many secure sites will not function without the Referer header present, as part of a security check (As discussed in [Bug 141641](#)), e.g. some sites use a third party company for processing credit card transactions and may fail without the header. It shouldn't be necessary to set this to **false**.

Plugins

plugin.expose_full_path. (Boolean) Typing **about:plugins** into Firefox's Address bar and hitting Enter displays a listing of plugins installed. With this set to **true** the complete path to the plugin is displayed, e.g. File name: C:\Program Files\Mozilla Firefox\plugins\npnl32.dll. This may be of some use as not all plugins are located in the Firefox\plugins directory. When set to **false** only the plugin names are displayed, e.g. File name: npnl32.dll.

security.enable_java. (Boolean) Setting this option to **true** enables support for Sun MicroSystem's [Java](#) in Firefox. If you haven't got this installed then obviously you may leave this set to **false**, or if you merely do not want any Java content loaded on webpages, e.g. news selectors. Personally I'd recommend downloading the Sun's Java VM and *enabling* this feature. This can also be used to provide Internet Explorer with a much better Java VM too.

plugin.default_plugin_disabled. (Boolean) Setting this to **true** enables Firefox to prompt you whenever you need a plugin not already installed to view content on a page, e.g. Real Player's "*Click here to download plugin*"



You can choose to ignore this prompt, or install the required plugin. Set this to **false** to disable such prompts.

privacy.popups.disable_from_plugins. (Integer) This option allows you to control how popups from installed plugins are treated, three options are available:

0. This setting enables all popups from plugins, which isn't particularly recommended.
1. This setting enables popups from plugins, up to the limited specified by **dom.popup_maximum** (covered later).
2. This setting disables creation of popups from plugins, unless the site is listed as an **Allowed Site** in the **Web Features** tab.

Rendering

nlayout.initialpaint.delay. (Integer) This value specifies the amount of time in milliseconds before Firefox should attempt rendering a page (Default is **250**). This is perhaps the most misunderstood setting

in Firefox, with numerous websites recommending you set this to **0** for fastest browsing. Somewhat ironically this actually increases the total time rendering a webpage, as explained by [Asa Dotzler](#):

*Setting the initial paint delay at 0, may get you some content on the screen faster, but it's worth noting that **it will dramatically slow down the time it takes the entire page to display**. Here's what's going on. Gecko, Firefox's rendering engine, is trying to optimize between the cost of waiting for a bit more data versus doing more painting and reflows as new data comes in. Waiting a bit longer before it starts painting the page gives Gecko a chance to receive more content before chewing up CPU cycles to render and reflow the document. If you drop this value down to 0 or near 0, that means you'll see the page start displaying a bit earlier, but not having received much data in that short interval, you'll have a lot more paint and reflow cycles to complete rendering of the page.*

So if you don't care about overall rendering time then yes, setting this to **0** can result in some content being displayed immediately (giving the perception of improved speed). If however, you'd prefer improved overall page rendering time then you'll want to adjust this value based on bandwidth available. Low bandwidth (dial-up) connections can benefit noticeably by increasing this value, with **750/1000** (The former being what I use currently) proving good. High bandwidth connections should be just fine at the default (**250**) – besides, is ¼ second really that noticeable a delay anyway?

content.notify.interval. (Integer) This option specifies the amount of time, in microseconds (1/1000 millisecond), between reflows (Default of **120000**). If you intend to adjust this value then bandwidth should play some role – low bandwidth connections are unlikely to receive much data over 0.12 seconds so you might see some performance benefit to giving it increased time to receive data before attempting a reflow. With high bandwidth connections the opposite may likely prove to be true, although reducing the time between reflows further can raise CPU use, so there's no need to reduce this excessively, if at all, in such cases. If you do wish to use a lower value then **100000** (0.1 seconds) should be the lowest you consider going.

content.max.tokenizing.time. (Integer) The value entered here determines the amount of time, in microseconds, after which parsing is interrupted to return to the application's event loop. It's recommended to set this to **3 x content.notify.interval**, e.g. with a content.notify.interval of 150000, set content.max.tokenizing.time or 450000.

browser.display.show_image_placeholders. (Boolean) Setting this to **true** enables the display of an image placeholder while loading images on a webpage (Getting replaced by the images as they download). Setting this to **false** disables such placeholders, which is recommended if you've a fast internet connection (As more than likely the placeholder will be replaced with the image almost immediately). On the downside however, the placeholder for broken images also no longer displays (Though personally it's not something I need).

image.animation_mode. (String) This setting controls rendering of animated images. **Normal** specifies that the image is animated as many times as the file specifies (Infinitely probably). **Once** (amazingly enough) sets that an image is animated only once, i.e. not looped; the final frame of the image is left displayed when the animation has finished, while **none** sets that the image is not animated, displaying only the first frame of the image instead.

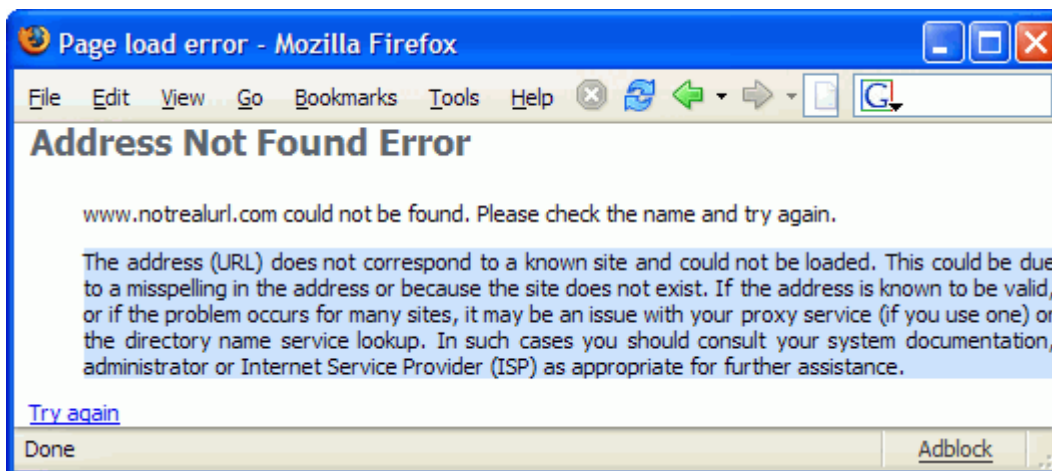
images.dither. (String) Dithering can be used to create the illusion of enhanced colour depth in images with a limited colour palette, e.g. gifs. That said, dithering can also make images appear grainier. As such it's mainly a matter of preference whether you prefer the appearance of dithered images or not. Firefox provides 3 options to control image dithering:

auto. This option allows Firefox to determine whether an image should be dithered.

true. This option enables image dithering in Firefox.

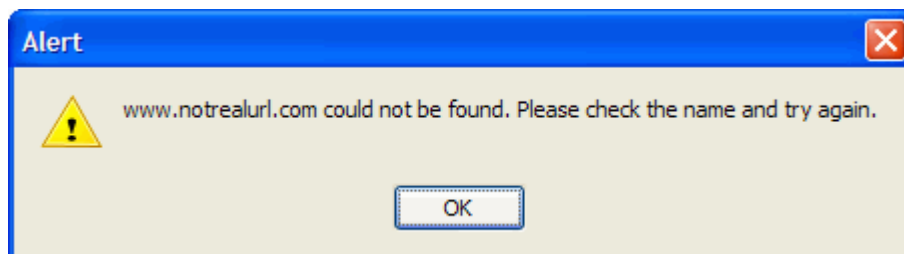
false. This option disables image dithering in Firefox.

browser.xul.error_pages.enabled. (Boolean) When set to **true** Firefox displays complete error pages in the event of browsing errors e.g. typing in a non-existent URL would result in the following.



This is much like what Internet Explorer currently does although isn't enabled by due to [some problems](#). If you wish to use it in this way then it's recommended to install the complimentary [Show Failed URL](#) extension.

When set to **false** (default) Firefox will instead display an error window if an error occurs.



Scripting

javascript.enabled. (Boolean) JavaScript is an open, cross-platform object scripting language (Not to be confused with Sun MicroSystem's [Java](#)) and has proven repeatedly to be a security risk with several (now fixed) vulnerabilities attributed to it e.g. [firtabbing](#) and [code execution via JavaScript](#). Setting this to **false** will provide improved security, although you will likely find many webpages that don't function correctly as a result. Set this to **true** should you desire support for JavaScript. Note – This is the same as the **Enable JavaScript** option in the **Web Features** tab.

dom.allow_scripts_to_close_windows. (Boolean) This option controls which windows can be closed via script. Setting this to **true** specifies that any window may be closed and isn't recommended. Entering **false** specifies that only windows opened by script can be closed via `close()`.

dom.disable_image_src_set. (Boolean) This option determines whether JavaScript is allowed to change images. Set this to **true** to enable this feature and **false** to disable it (recommended). Note – This is the same as the **Change images** option in **Advanced JavaScript Options**.

dom.disable_open_click_delay. (Integer) This option specifies the amount of time, in milliseconds, that must be surpassed before a popup window created by JavaScript `setInterval()` or `setTimeout()` calls aren't managed by current Popup Blocker settings. Beneath this threshold existing Popup Blocker settings are applied. By default this is **1000** (1 second).

dom.disable_open_during_load. (Boolean) Set this to **true** to enable Firefox's built-in pop-up blocker, which disables the loading of much pop-up content on sites – which will mostly be advertisements (it's worth noting this is not perfect and will also disable many legitimate pop-ups). Should a pop-up be blocked in this way an information bar will appear at the top of the window, from which you can select what action to take. Setting this to **false** disables the pop-up blocker (not recommended). Note – This is the same as the **Block Pop-up Windows** option in the **Web Features** tab.

dom.disable_window_flip. (Boolean) This option controls whether JavaScript may be used to bring windows into the foreground/background via `focus()`. Setting this to **true** disables such actions, which won't affect new popups from loading in the foreground, though can force existing ones to remain in the background unless switched to manually. Set this to **false** to allow the script to determine what happens. Note – This is the same as the **Raise or lower windows** option in **Advanced JavaScript Options**.

dom.disable_window_move_resize. (Boolean) This option controls whether JavaScript can be used to move and/or resize windows, whereby setting this to **false** enables scripts to do this. It would perhaps be best to set this to **true**, allowing only yourself to resize/move windows. Note – This is the same as the **Move or resize existing windows** option in **Advanced JavaScript Options**.

dom.disable_window_open_feature.close. (Boolean) Set this to **false** to enable the use of scripting to hide the close button of windows, **true** forces the close button to always be displayed (recommended).

dom.disable_window_open_feature.directories. (Boolean) Set this to **false** to enable the use of scripting to hide the bookmarks toolbar, **true** prevents the bookmarks toolbar from being hidden in this way.

dom.disable_window_open_feature.location. (Boolean) Set this to **false** to enable the use of scripting to hide the Location (Address) bar, **true** prevents the Address bar from being hidden.

dom.disable_window_open_feature.menubar. (Boolean) Set this to **false** to enable the use of scripting to hide the Menu bar, **true** disables the hiding of the Menu bar.

dom.disable_window_open_feature.minimizable. (Boolean) Set this to **false** to enable the use of scripting to disable the minimizing of windows, **true** enables the minimizing of such windows (recommended).

dom.disable_window_open_feature.resizable. (Boolean) Set this to **true** to enable the use of scripting to hide the close button of windows, **false** forces the close button to always be displayed (recommended).

dom.disable_window_open_feature.scrollbar. (Boolean) Set this to **false** to enable the use of scripting to hide the Scroll bar in windows, **true** disables the hiding of the Scroll bar in windows.

dom.disable_window_open_feature.status. (Boolean) This option controls whether JavaScript can be used to hide the status bar, whereby setting this to **false** enables scripts to do this. Set this to **true** to force the status bar to be displayed at all times. Note – This is the same as the **Hide the status bar** option in **Advanced JavaScript Options**.

dom.disable_window_open_feature.titlebar. (Boolean) Set this to **false** to enable the use of scripting to hide the Title bar of windows, **true** forces the Title bar to always be displayed.

dom.disable_window_open_feature.toolbar. (Boolean) Set this to **false** to enable the use of scripting to hide the Navigation toolbar, i.e. Back, Forward, etc. buttons, **false** prevents the hiding of the Navigation toolbar.

dom.disable_window_status_change. (Boolean) This option controls whether JavaScript can be used to display custom text in the status bar, e.g. moving the mouse over a hyperlink normally would display where the link points to, though a script could be used to display something else instead. Set this to **false** to allow such custom status bar text displayed, while **true** will disable this. Note – This is the same as the **Change status bar text** option in **Advanced JavaScript Options**.

dom.event.contextmenu.enabled. (Boolean) This option controls whether JavaScript can be used to alter, or even disable the context menu, e.g. right clicking could be disabled on certain pages. Set this to **true** if you wish to allow sites to be able to do this, while **false** ensures scripts can't be used to alter this functionality. Note – This is the same as the **Disable or replace context menus** option in **Advanced JavaScript Options**.


dom.max_script_run_time. (Integer) This specifies the amount of time, in seconds, that a script may run – the default being **5**. It'd be recommended you leave this at **5** unless you have a specific reason to change it, in which case **-1** allows scripts to run as long as required (and I'd doubt most reading this would need to do that).

dom.popup_maximum. (Integer) This value specifies the maximum number of pop-up windows that may be open simultaneously. By default this is set to **20**, which no doubt might seem like an excessive amount. Unless you have a particular reason to allow so many to be open at once, it would be a good idea to lower this value (Considering certain sites out there try flooding you with numerous pop-ups). Personally I've set this to **5**.

Miscellaneous

config.trim_on_minimize. (Boolean) Setting this to **true** specifies that Firefox can be swapped out of memory when minimized. This yields more RAM available for other applications. That said, as discussed in [Bug 76831](#), Firefox has a tendency to respond slowly whenever brought to the foreground again. As such you might find it best to set this to **false** to disable such swapping, thereby avoiding the responsiveness issues.

extensions.dss.enabled. (Boolean) Setting this to **true** enables **DSS (Dynamic Skin Switching)** support, which allows you to change themes at will without requiring Firefox to be restarted to apply the new theme. As discussed in [Bug 226791](#), it's still a highly problematic implementation and would be best left set to **false**.

browser.throbber.url. (String) The value entered here specifies the URL which the activity indicator (Or throbber), i.e. this thing , loads when clicked upon. By default this'll be <http://www.mozilla.org/products/firefox/central.html>. Specify a new address here if you want to.

alerts.totalOpenTime. (Integer) This value specifies the amount of time, in milliseconds, that Firefox alerts, e.g. Download complete, should be displayed. The default is **4000**. Increase/Decrease accordingly depending on how long you feel would be more appropriate to notice such alerts.

browser.chrome.toolbar_tips. (Boolean) Setting this to **True** enables the display of tooltips when the cursor is idle over certain icons/buttons, e.g. Back/Forward buttons. These can be useful enough if you're new to using Firefox. More experienced users however may feel these unnecessary – if so set this to **false** to disable them.

accessibility.typeaheadfind. (Boolean) Setting this to **true** enables you to **Find in this page** by typing letters, eliminating the need to manually start **Find in this page** (e.g. by pressing Ctrl + F). Set this to **false** to disable such automatic initialisation, which you might find to prove less troublesome. Note - This is the same as the **Begin finding when you begin typing** option in the **Advanced** tab.

browser.tabs.opentabfor.middleclick. (Boolean) When set to **true**, clicking/depressing the middle mouse button (Scroll wheel more commonly) over a link opens that link in a new tab. When set to **false** this does not occur (You can still open the link in a new tab by **right clicking** on it and selecting **Open link in new tab**).

Final Comments

Hopefully this guide will have assisted you well in tweaking Firefox to suit your needs and tastes. Remember to check out our top choices of [Firefox extensions](#), and why not, try a few others that you may find attractive in order to enhance your daily browsing.

Finally, we are always looking to improve our published articles, especially this kind of guides, so don't hesitate to [get in touch](#) with us for any suggestions or feedback.