

Firefox Nightly Build Adds About:Permissions For Managing User Privacy

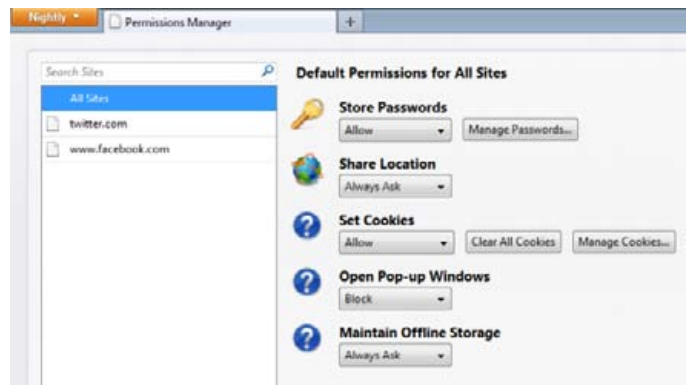
Written by Amit Banerjee on Friday, May 27th, 2011

When it comes to the privacy and security of your online profiles, email, social accounts or your website – the first thing to take note is your browser.

Since every other web activity is performed on the browser, it is really important to configure your browser for maximum security so that you don't end up giving away your data to third party websites. The option's panel of your browser may be configured to store your passwords, share your location, handle cookies and other privacy related features.

At the end of the day, most novice users are not well aware about how secure their browser is. They either don't care whether a website is tracking their location, whether they have saved a whole bunch of passwords on their browser (which can be risky) or whether third party websites are accessing their [browsing history or cached data](#) in the background.

Thankfully, the latest nightly build of Firefox has introduced a new tool which can be used to tweak the necessary permissions given to third party websites. Here is how the about:permissions page in Firefox looks like:



To open the about:permissions page in Firefox, you must install the latest nightly build from the Firefox nightly website. Trying to access this page on your current Firefox installation e.g Firefox 4 might give a page not found error.

The about:permissions page, as the name suggests can be used to tweak privacy related features of all third party websites from a single page. Although the same preferences are available under Tools > options, it seems that the Firefox team wants to provide a dedicated page to users so that they can control the default permissions for websites from a single page.

Here are a few things which can be done from Firefox's about:permissions page:

1. Choose the websites with whom you want to share your location: If you frequently use Google Maps, Foursquare and other location sharing apps, you might want to add these sites to Firefox's about:permissions page and choose "Always share location with these sites".

For a specific site which you don't trust and don't want to give them your current location data, it's

Firefox Nightly Build Adds About:Permissions For Managing User Privacy

wise to choose “Always ask” or “Block”. Choosing ‘Always ask’ will show a notification like “This site wants to know your location” when you open it on Firefox. I would advise choosing the “Always ask’ option over “Block” so that you can decide to share your location data when you open or use a site, rather than choosing “Block” which will prevent sites from tracking your location.

2. Allow Only Specific sites to Store passwords:

Again I would advise not to store passwords on the browser and use a dedicated [password manager application like Keepass](#). But if you don’t want to use a password manager program and have the habit of saving passwords in Firefox, you can use the about:permissions page to define for which sites Firefox would store and manage passwords.

Example: If you have 6 Gmail accounts and you want to store all the passwords only for Gmail and no other site, it’s a good idea to add mail.google.com to Firefox’s about:permissions page, save all the username and passwords one by one and then choose the “Block” option.

This will configure Firefox to never save any passwords for any other site except Gmail. Your saved username and passwords will continue to work as before, it’s just that Firefox will not throw the dialog box ‘ Do you want this password to be saved?’ in future.

3. Blocking Popup Windows for specific sites:

The same is true for allowing or blocking popups from sites you don’t trust. If you use a lot of Firefox add-ons and don’t want to block pop ups from Mozilla, add the pop up blocking exception to the about:permissions page and choose “Block Pop up windows” option. This will block pop up windows from every other site while allowing popups from the sites that are present in the exception list.

Other options are allowing and blocking cookies, and maintaining offline storage for sites that support it. The newer dashboard will surely help novice users who need a simple page for managing passwords, location, cookies and other privacy related options in Firefox.

Remove Unnecessary Codes From Head Section of Wordpress Theme

Written by Amit Banerjee on Monday, May 23rd, 2011

The following tips are applicable for both self hosted WordPress blogs and blogs that are hosted on WordPress.com's free blogging platform.

If you have a blog or a website powered by WordPress content management system, you might have noticed some lines of code before the `</head>` section of your website's WordPress theme. While some of the codes are useful and have a purpose, a few of them are not.

This tutorial shows you how to remove unnecessary codes from the head section of a WordPress theme.



Tips & Tricks

`</head>`

Let's take a look at some example codes one by one

1. `<link rel="EditURI" type="application/rsd+xml" title="RSD" href="http://example.com/xmlrpc.php?rsd" />`

WordPress supports an XML-RPC interface which is used by blogging clients like Windows Live writer, Ecto etc for publishing blog posts to your website directly from desktop. If you always use a browser to [write blog posts](#) within your WordPress administration area, the above line of code is not useful.

Remove Unnecessary Codes From Head Section of Wordpress Theme

2. `<link rel="wlwmanifest" type="application/wlwmanifest+xml" href="http://example.com/wp-includes/wlwmanifest.xml />`

The `wlwmanifest.xml` file is used by Windows Live writer to fetch Tags and Categories of your WordPress blog on the desktop blogging client. Again, if you are not using [Windows Live writer](#) to write blog articles, this code is meaningless.

3. `<link rel='index' title="" href='http://example.com/' />`

The recent version of WordPress 3.0 automatically inserts a link rel tag on the head section of your WordPress theme which points to your homepage. This is not to be confused as an SEO improvement.

Instead, some WordPress themes might use this tag for varied reasons. If your theme does not refer to this tag or makes a call, you should better remove the index page referrer from the head section of your WordPress theme.

4. `<link rel='start' title="" href='http://example.com/first-post/' />`

This is another annoying piece of code which points to the first post published on your current WordPress installation. The annoying part is that WordPress automatically embeds this code on the head section of every page on your site. Again, this code does not help in SEO or [better crawling](#) of your pages.

5. `<link rel='prev' title="" href='http://example.com/previous-post/' />`

`<link rel='next' title="" href='http://example.com/next-post/' />`

Above pair of codes are another rel link which points to the previous and next blog entry. Serves no real purpose.

6. `<meta name="generator" content="WordPress 3.1.2 />`

This one is added from the WordPress header hook and shows the WordPress version installed on your site. It can be a [security concern](#), so you should remove it as well.

7. `<link rel='shortlink' href='http://wp.me/abcde' />`

This is the [shortlink version](#) of a published post on your blog, which is embedded from the WordPress header hook. Some themes are coded to use it, so before dropping this one, please double check and verify that you are not using this value anywhere on your theme.

Removing these unnecessary codes from the head section of your WordPress theme is fairly easy. Open your theme's `functions.php` file and paste the following codes (be careful, take backups first).

```
remove_action( 'wp_head', 'wp_shortlink_wp_head', 10, 0 );
remove_action( 'wp_head', 'rsd_link' );
remove_action( 'wp_head', 'wlwmanifest_link' );
add_filter( 'index_rel_link', 'remove_code' );
add_filter( 'parent_post_rel_link', 'remove_code' );
add_filter( 'start_post_rel_link', 'remove_code' );
add_filter( 'previous_post_rel_link', 'remove_code' );
add_filter( 'next_post_rel_link', 'remove_code' );
add_filter( 'post_comments_feed_link', 'remove_code' );
```

```
function remove_code( $data ) {
    return false;
}
```

While keeping the above codes in the head section of your theme **won't harm your site** in any sense, but it's always better to reduce overhead of HTML markup and serve only what is required. Why in-

Remove Unnecessary Codes From Head Section of Wordpress Theme

crease the overall size of all the pages of your site with useless machine generated stuff which serves no purpose at all?

Important note: It's important to note that all of the above codes can be removed by editing core WordPress files (e.g wp-includes/wlmanifest.xml). But I would suggest you to use theme specific codes for two reasons. First, when you upgrade WordPress to it's latest version, all your customized hooks will be overwritten with their default code which will bring the earlier codes back.

Second, isn't it always easy to enable or disable stuff from a theme or a plugin when compared to editing [core WordPress files](#), which can break existing features?