

Firefox Gear WordPress Theme Guide

The Firefox Gear theme installs like any other WordPress theme, with just a few extra steps required by the design and architecture of this particular website.

Step 1: Upload the theme files

We'll assume you've already uploaded, installed and configured WordPress and are ready to install and activate the Gear theme on an otherwise pristine blog. We'll also assume you're working with WordPress 2.8.5, which is the current release as of this writing. Older versions may not be compatible with some plugins and theme features, so WordPress 2.8 is the minimum requirement.

The theme is bundled with one dependent plugin called Sociable, which dynamically generates links to a range of social networking sites. This third-party plugin may occasionally release its own updates. It's always possible a future release will remove and/or break existing functionalities, so keep backups and upgrade with care.

The Zip archive is organized into folders corresponding to the folders you'll find in `/wp-content/` in a standard WordPress installation; use that as your guide to know which files go where.

Add the entire *FirefoxGear* theme folder to `/wp-content/themes/`

Add the included *Sociable* plugin folder to `/wp-content/plugins/`

Step 2: Activate the theme

In the WordPress administration panel, navigate to Appearance → Themes. Select the Firefox Gear theme and activate it. This part is easy.

Step 3: Activate and configure Sociable

Navigate to Plugins → Installed and, with either "All" or "Inactive" plugins showing, activate Sociable. As soon as it becomes active, a new link will appear at the end of the "Settings" menu in the WP admin sidebar, granting access to Sociable's own options page.

When you visit the Sociable options page you'll first see a large grid comprising dozens of social networking sites. The Gear site will only support five of them, so locate and check those five, unchecking all others. The five we're interested in are:

1. Twitter
2. Facebook
3. Del.icio.us
4. Digg
5. StumbleUpon



Sociable Options

Sites:

Check the sites you want to appear on your site. Drag and drop sites to reorder them.

<input checked="" type="checkbox"/> Twitter	<input checked="" type="checkbox"/> Facebook	<input checked="" type="checkbox"/> del.icio.us
<input checked="" type="checkbox"/> Digg	<input checked="" type="checkbox"/> StumbleUpon	<input type="checkbox"/> Add to favorites
<input type="checkbox"/> BarraPunto	<input type="checkbox"/> Bitacoras.com	<input type="checkbox"/> BlinkList
<input type="checkbox"/> blogmarks	<input type="checkbox"/> Blogosphere News	<input type="checkbox"/> Blogplay
<input type="checkbox"/> blogtercimlap	<input type="checkbox"/> connotea	<input type="checkbox"/> Current

This is the default order shown in the original design, but you can drag and drop the blocks to arrange them in any order you prefer. The Sociable plugin includes support for many other networks, but these are the only five that have been specifically styled in the theme. If you do add another, its link will display with a generic icon, though you can just as easily edit the style sheet to add more icons.

The plugin's intended use is to automatically add a row of icons at the end of posts and pages, but we're doing a bit of customization to bend the plugin to your purposes.

- Check the box to disable sprite images (we're providing our own icons).
- Check the box to disable alpha mask (we don't be needing that effect).
- Delete the default tagline, ensuring that the tagline field is blank (we're using our own title).
- Uncheck all the position options (we're positioning it ourselves in the template code).
- Uncheck the Sociable style sheet option (we're using our own style sheet in this theme).
- Check the option to use text links without images (we just need the text links and we'll add our own icons with CSS).
- Do not specify an image directory, leaving that field blank (we have our images coded into the CSS).
- Do not use thickbox/iframe on links (we're not using that effect).

The remaining options aren't of consequence so you can toggle them however you like. Last but not least, click "Save Changes" to finish configuring Sociable.

NOTE: You can disable Sociable on individual posts. **However**, there is a bug in the current version (3.5.2 as of this writing) that prevents it from being re-enabled once it's disabled. A future update will hopefully fix this problem, but for the immediate future, disabling Sociable on a single post must be considered a near-permanent decision. You can reverse it by editing entries directly in the database, or by deleting and republishing the post (back-dated if necessary), but neither of those solutions is ideal. So disable with care until a fix is released.

Step 4: Set image sizes

Navigate to Settings → Media and modify the maximum widths for both medium and large images:

- Medium: max width = 444
- Large: max width = 672

The height can be any reasonable limit, or set to 0 for no maximum height. The thumbnail size can be whatever you like, but I recommend leaving the default 150x150 size with the cropping option active.

Setting these dimensions allows WordPress to scale image uploads automatically to fit the site's layout without the need to scale them manually before uploading.

Step 5: Set the date format

The design calls for a particular date format with abbreviated days and months, e.g. "Fri, Nov 6, 2009." To achieve this you'll need to set a custom date format under Settings → General. WordPress uses standard PHP date/time notation, which is a bit cryptic if you're not familiar with it. The format setting should read **D, M j, Y**.

This format isn't a technical requirement of the theme so you can use any date format you prefer, but it's what was in the original design so it's noted here for reference.

Step 6: Create the Add-ons category

Most posts on Firefox Gear will be covering and promoting a particular add-on. However, if there's ever a need to publish a more general post (news, announcements, etc) it would be unfair to assume **all** posts will promote an add-on. The simplest means to distinguish add-on posts from general posts is with categories.

Create a new category called "Add-ons" with a slug of "add-ons" (in fact the category title can be anything you like, so long as the slug is "add-ons" including the hyphen; the theme relies on that slug to differentiate types of posts). If you like, you can set the Add-ons category as the default post category in Settings → Writing to save a step when publishing new content.

WordPress installs with a single category named "Uncategorized," which you can rename but can't remove (WordPress requires at least one category). That default category acts as a generic catchall for any posts that aren't assigned to any other category.

Step 7: Publish the About page

WordPress installs with an example Page titled "About," so you can either edit this existing page or delete it and create a new one. The title can be anything you prefer, but it's important that the page slug is "about." There is some conditional logic in the theme that depends on that slug.

If for some reason you prefer a different slug (perhaps because it's visible in the URL), you'll also need to make two minor changes to the theme code.

In the Header template (header.php), around line 60, find:

```
if ($aboutpage == 'about')
```

That value is the slug in question, so simply change it to your preferred slug. Repeat the same change around line 100 (still in the Header template). These two simple arguments merely determine whether to display links to the About page, so the links only render if said page exists (to prevent broken navigation). But if you stay with a slug of "about" you won't need to modify any code.

Step 8: Create the custom fields

Add-on posts on Firefox Gear demand a few extra bits that WordPress doesn't provide out of the box, namely the prominent "Try It Now" button and the add-on profile links. These are easily added to posts by making use of WordPress custom fields. Additionally, the site's design and page layout imposes a slightly unorthodox means of attaching a header image to posts, so that image will also be added via a custom field. More on that later. Right now we just need to establish the field names so they'll be saved for future use.

To create new custom fields, at least one post or page must exist first. WordPress includes one example post, so you can either add these custom fields to that existing post before you delete it, or delete the post and add them when you create your first new post. The five fields to create are:

- **amo_url** – this will be the address of the add-on's page at addons.mozilla.org, and will be the URL assigned to the large "Try It Now" button.
- **original_listing** – The URL of the add-on's original listing, if it exists. This link appears in the add-on profile in the post's sidebar.
- **addon_home** – The add-on's official home page, if it has one. This link appears in the add-on profile in the post's sidebar.
- **developer_home** – The add-on developer's home page, if she has one. This link appears in the add-on profile in the post's sidebar.
- **main_image** – the URL for the main banner image that appears in the post header.

Once these field names are established in the database they'll be available as options in a dropdown menu on every new post, so this is a one-time setup.

And with that, the initial installation and configuration of the Firefox Gear theme is complete. Any other WordPress settings are at your discretion (permalinks, commenting, feed syndication, etc.). The only thing to do now is to start publishing content.

Publishing a Post

Posts promoting specific add-ons get some special treatment in the form of a prominent “Try It Now” link and a few other supplemental links in an add-on profile (other posts – that is, any post not in the Add-ons category – will not display these additional links). The links themselves are added to the post as custom fields, which we’ve already established in the previous step (or you may be establishing them now with the first new post).

Note that all of these custom fields are in fact optional, so nothing gets displayed if a given field is missing or empty. For example, not every add-on developer will have his or her own home page, so in those cases simply don’t add that field. If none of the profile links are present, the entire profile block is omitted. Similarly, if for some reason the `amo_url` field isn’t present, the post will appear without a button.

The particular page layout and structure of this theme also imposes a few special requirements when authoring posts. Each post leads with an introduction and a large header image, distinct from the post body. Furthermore, the height of that post header must influence the placement of the post sidebar (which contains the Try button, comment status, share links, and add-on profile). The simplest way to accomplish this particular layout is to treat the header section as separate content from the post body, so we’ll use the custom excerpt field for the introduction and a custom field for the header image.

The Excerpt

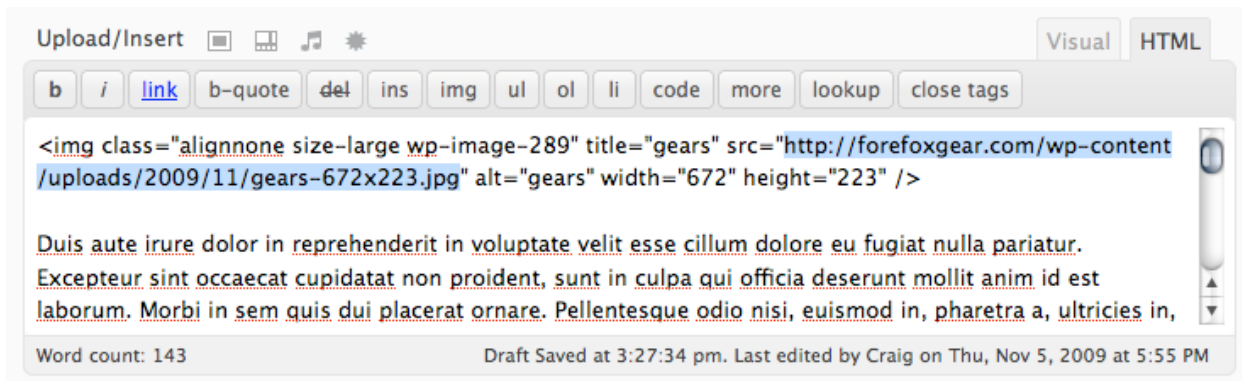
If you don’t manually provide a custom excerpt, WordPress automatically generates an excerpt consisting of the first 50 words or so of the post. Because we’re using the excerpt as a post introduction, we don’t want to repeat the same content, so the introduction only appears on a published post if a custom excerpt has been provided. No custom excerpt, no intro blurb. There’s no length limit on custom excerpts, but you should keep it brief, no more than a few short sentences.

The Header Image

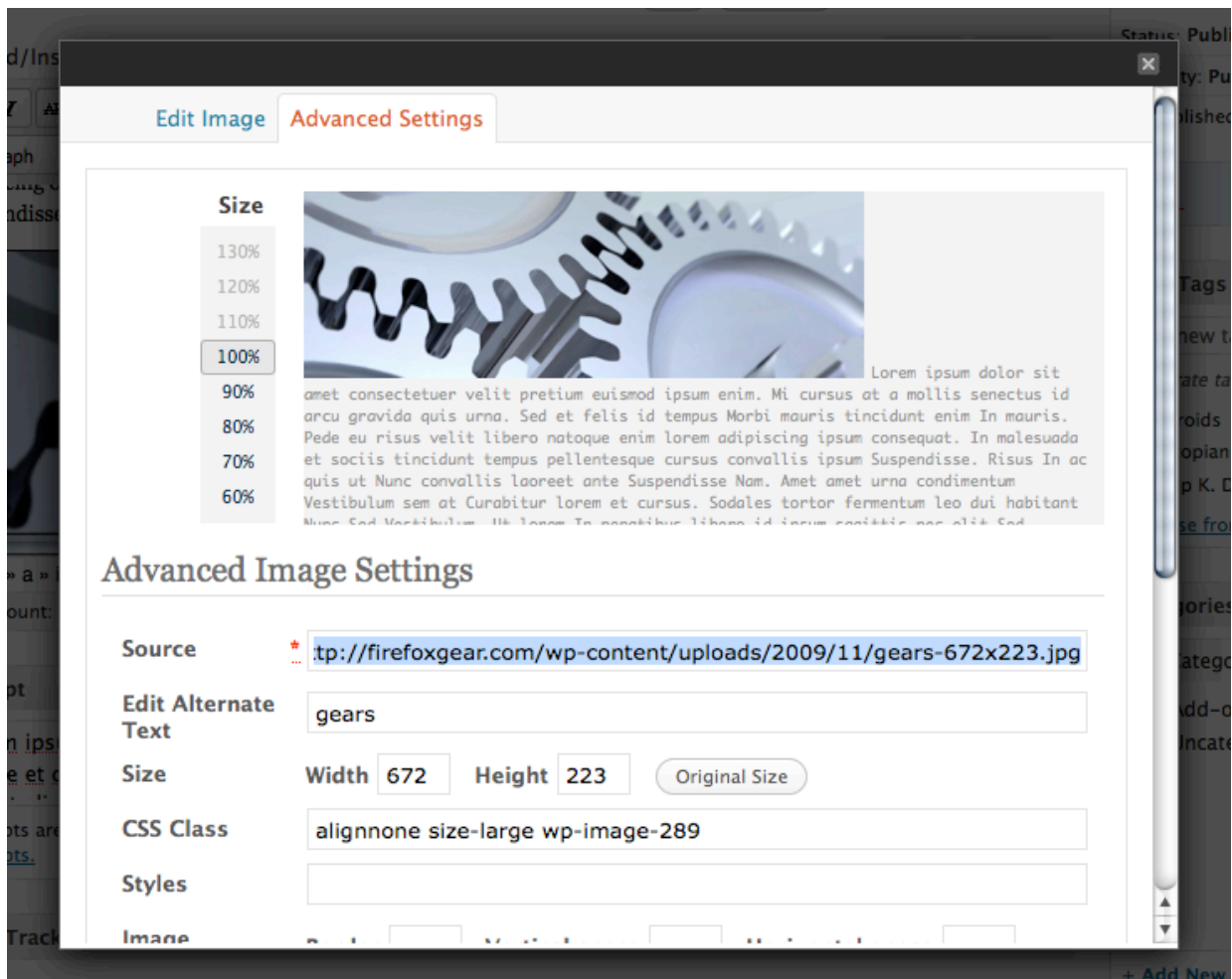
Because the layout separates the head from the body, the main banner image must also be embedded separately from the main post content. This requires using a custom field to pass along the URL for the image. No image will be rendered if that field isn’t present, so the header image is entirely optional on a post-by-post basis.

Since the header image is embedded via a custom field, you unfortunately won’t have the ease and luxury of attaching it as painlessly as you would an image embedded directly into a post. The simplest approach may be to first embed the image in the post normally (selecting the Large size since that has been scaled to 672 pixels wide), then copy the image source URL to paste into the `main_image` custom field before deleting the embedded image. It’s a bit of an obstacle course, but it’s probably the easiest means of getting that all-important source URL.

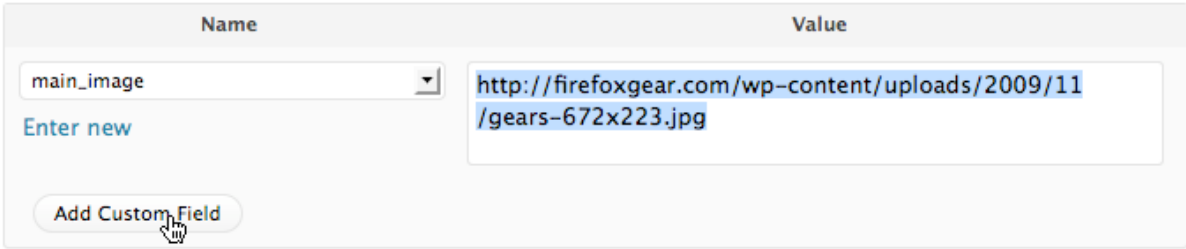
To get the image URL, it may be easier to switch the editing pane to HTML mode:



Or from Visual mode, edit the embedded image and find the source in the resulting dialog, under the Advanced Settings tab:



This is the URL you'll insert into the main_image custom field:



The screenshot shows the 'Add new custom field' interface in WordPress. It features a table with two columns: 'Name' and 'Value'. The 'Name' column contains a dropdown menu with 'main_image' selected. The 'Value' column contains a text input field with the URL 'http://firefoxgear.com/wp-content/uploads/2009/11/gears-672x223.jpg' entered. Below the table is a button labeled 'Add Custom Field' with a mouse cursor hovering over it. There is also a link 'Enter new' and a note at the bottom stating 'Custom fields can be used to add extra metadata to a post that you can [use in your theme](#).'

The HTML that embeds the image on the page is taken care of in the theme template, so that's all there is to it. Just be sure you remove the one you embedded in the post.

You can, of course, embed other images into posts as usual. Just remember that the column is narrow so choose the Medium size (or smaller) to prevent any unwanted layout issues when you publish.

Enjoy!

Checklist

1. Upload the files
2. Activate the theme
3. Activate and configure the Sociable plugin
4. Set the image sizes
5. Set the date format
6. Create the Add-ons category
7. Publish the About page
8. Create the custom fields