



DIGGING INTO WORDPRESS

by Chris Coyier & Jeff Starr

Supplemental Bonus Content

[Awesome Features](#)

[WordPress MultiSite](#)

[Theme Customizer](#)

[Bonus Tricks](#)

Acknowledgements

- Thank you to Thane Champie and James Starr for their help with proofreading.
- Thanks also to the many readers who have helped with further improvements.
- Thank you to everyone who allowed screenshots to be printed in this book.
- Thank you to the incredible WordPress community for making WordPress #1.
- And most of all, thanks to you, the reader, for sharing this adventure with us.

Chris would like to thank

- My mom for all the excellent printing advice (and for being a good mom).
- Jeff Penman for insisting the book was a good idea.
- Tim Chatman for giving me the time and assistance I needed to get it finished.

Jeff would like to thank

- My wife, Jennifer, for her loving support and encouragement.
- My two children, Josh and Lorelei, for being so awesome.
- Friends, family, and everyone who helps along the way.



DIGGING INTO WORDPRESS
Supplemental Bonus Content
By Chris Coyier and Jeff Starr
Version 3.7s • October 2013
© 2013 Digging Into WordPress
Visit us online @ DigWP.com

DIGGING INTO
WORDPRESS

SUPPLEMENTAL BONUS CONTENT

CHRIS COYIER & JEFF STARR

Bonus Content!

Digging Into WordPress is updated frequently, with new and revised material to help you stay current with the latest and greatest. Now in its 11th edition, the book has grown to over 400 pages in length — which is awesome, but expensive to print.

So to lighten the load and streamline the book, we've moved its three peripheral chapters (basically WP update information and bonus tricks) to this supplemental PDF. This helps keep the core book focused, while the original material remains available for those who want to dig deeper into WordPress.

It's a win-win!

3.7

That is the current version of WordPress at the time this book was published. So if we say something like "...the current version of WordPress," we are talking about 3.7. If we need to mention an older version, we'll be specific about that.

So what if you are reading this and 3.8 is already out? **Don't worry about it!** The information in here will still be valid. WordPress does a good job about not breaking existing stuff for new versions.

But there will be changes, and we intend to keep this book updated with those new things. All current owners of this book will get free PDF updates as it is updated!



See that? That's dog food. It's a metaphor.

*We, the authors of *Digging Into WordPress*, eat our own dog food. We aren't just here to stand on a pedestal and preach about how you should do things. We practice these things in the sites we work on every day.*

Much of what you will read in this book is put into practice on the WordPress blog that accompanies this book.

<http://digwp.com>

Contents

1 Bonus Tricks!

1.1 Everybody Loves Bonus Tricks	7
1.1.1 Add Author Bios to Single Posts	7
1.1.2 Adding a Theme Options Panel	10
1.2 Free WP Theme: Lines & Boxes	14
1.2.1 Child Themes	14
1.3 AJAXing WordPress: The “All AJAX” Theme	16
1.4 Free WP Theme: Plastique	17
1.4.1 Layout Options, Widgets, & Custom Content	17
1.4.2 Child Themes and Category Styles	19
1.4.3 Other Plastique Features	20

2 WordPress 2.9 Update

2.1 Like a River...	23
2.2 New in WordPress 2.9	23
2.2.1 Image Editor	24
2.2.2 Trash Can	25
2.2.3 Embedding Videos with oEmbed	26
2.2.4 Database Maintenance Tools	27
2.2.5 Canonical Meta Tags	28
2.2.6 Featured Images (Post Thumbnails)	29
2.2.7 Metadata API	33
2.2.8 Widgetized Sidebar Descriptions	33
2.2.9 Custom Post Types	34
2.2.10 New Theme Templates	35
2.2.11 Register Feature Support	35
2.2.12 Custom Theme Directories	36
2.2.13 Other Cool Changes in Version 2.9	37

3 WordPress 3.0 Update

3.1 Giant Leap Forward...	39
3.2 New in WordPress 3.0	39
3.2.1 Goodbye Kubrick, Hello Twenty Ten	40
3.2.2 Goodbye "admin", Hello Custom Username	41

3.2.3 Custom Background Support	41
3.2.4 WordPress MultiSite: The Merging of WordPress with WPMU	44
3.2.5 Using Custom Taxonomies	49
3.2.6 Creating and Using Custom Menus	50
3.2.7 Custom Post Types	52
3.2.8 Shortlinks	54
3.2.9 Other Awesome 3.0 Features	57
Just the Beginning...	58
3.3 Welcome to WordPress 3.1	59
3.3.1 Custom Post Formats	59
3.3.2 Advanced Custom-Field Queries	62
3.3.3 Advanced Taxonomy Queries	65
3.3.4 Streamlined User Queries	67
3.3.5 Awesome New Insert-Link Tool	68
3.3.6 Admin Area for MultiSite Networks	69
3.3.7 Theme Feature Filter	69
3.3.8 The Admin Toolbar	70
3.3.9 Other Awesome Features & Improvements	71
3.4 WordPress 3.2 Update	72
3.4.1 Requirement Changes	72
3.4.2 Redesigned Admin Area	72
3.4.3 New Default Theme: "Twenty Eleven"	73
Moving on...	74

3.5 WordPress 3.3 Update	75
3.5.1 Sleek New Admin Area	75
3.5.2 New Toolbar	76
3.5.3 Feature Pointers	77
3.5.4 Better Contextual Help	77
3.5.5 Drag-n-Drop Media Uploader	78
3.5.6 Other Awesome Features in WP 3.3	79
 3.6 WordPress 3.4 Update	 82
3.6.1 Live Theme Customizer	82
3.6.2 Custom background and header images	88
3.6.3 More Theme-related Goodness	91
3.6.4 Other awesome improvements in WP 3.4	92
3.6.5 Bug-fixes and enhancements	93
Until next time...	95
Thank you	96

In the beginner's mind there are many possibilities, but in the expert's there are few.

— SHUNRYU SUZUKI

1

Bonus Tricks!

1.1 Everybody Loves Bonus Tricks

Right?! This chapter is new to version 2.0 of this book. If you are reading this, you have at least that version. And with version 2.0, you get exclusive **bundled themes**. We needed a place to talk about some of the things you'll find in those themes, hence the new chapter. Beyond that, we are going to share with you some fun WordPress tricks that you can use in *any* theme.

1.1.1 Add Author Bios to Single Posts

BY: Sarah Hagerman

Sarah lives a relatively quiet existence in Austin, Texas. She enjoys dancing to bluegrass, trolling through sales bins at record stores, hiking, camping and attending screenings of old movies.



Have you ever seen an article on a site end with a block like this? It can be a nice personal touch, especially for multi-author sites. WordPress has some built-in functionality

to get it done, and there are some plugins to help as well. Let's go through how you might implement this into your own site.

Username	chriscoyer	<i>Your username</i>
First name	Chris	
Last name	Coyier	
Nickname <i>(required)</i>	chriscoyer	
Display name publicly as	Chris Coyier	
Contact Info		
E-mail <i>(required)</i>	chriscoyer chriscoyer Chris Coyier Chris Coyier Coyier Chris	
Website	http://chriscoyer.net	

About Yourself

Biographical Info	Chris has been wrestling alligators since has been 13, and won the Louisiana State Gator Wrestling Championship at 27. When he's not wrestling gators he likes watching gator wrestling on TV and eating alligator tacos.
-------------------	---

Share a little biographical information to fill out your profile. This may be shown publicly.

There are three bits of data that we need to get our hands on here:

1. The author's name
2. The author's short bio
3. The author's photo

#1, getting the author's name, is the easiest. There is a built in WordPress function for displaying it:

```
<?php the_author(); ?>
```

The important thing to remember here is to set up a "Display name" that is the nicest version of the Author's name possible. If you don't do this, it will show the username of the Author, which among other limitations, doesn't allow spaces. In other words "Chris Coyier" is a lot nicer than "chriscoyer" or worse, "admin."

#2, getting the author's bio, is similarly easy. It's just a lesser-known WordPress function:

```
<?php the_author_meta('user_description'); ?>
```

This biography information you can add directly to any user from the Admin area. Just edit the user, type in their bio, and save.

#3, getting the author's photo, is slightly trickier. There are a few different routes you can take. Let's take a look...

Display Author Gravatar

Your authors might already have Gravatars (Globally Recognized Avatars, <http://gravatar.com/>). Gravatars are the most common way to handle user thumbnails in the WordPress world. The vast majority of themes support them in the comments area, so it can make sense to use it for author photos as well. This also puts the control of that photo in the author's hands.

You can use the following code to display the Gravatar of the author of the current post. As with the author name and bio, use this *within* the WordPress Loop (see sidebar note for more info):

```
<?php echo get_avatar(get_the_author_meta('ID'), 32); ?>
```

User Photo Plugin

Perhaps your site's design calls for author photos, but Gravatars aren't a good solution. Like you want to have a consistent style for all of them and leave that control up to you instead of your authors. Or, your authors already have Gravatars that they don't want to use for this purpose.

With the User Photo plugin <http://digwp.com/u/436>, you can bring author photo control into WordPress. With the plugin active, each user's profile in the Admin has a new area for uploading a photo for them.

User Photo


Full size

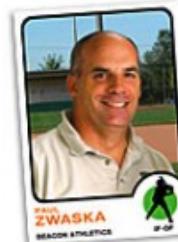

Thumb

Upload image file: (max upload size 2M)

Approval status:

Delete photo?

The plugin then gives you some new functions to use for outputting the photo in your theme. See the documentation.



By: Paul Zwaska
February 9th, 2010

We just returned from the Sports Disney's Coronado Springs Resort us and 2010 was no exception.

I'm sure those of you that had the year ago it was all gloom and doom spending freezes. But this year the attendance. They were excited to see what we exhibitors had on display.

I hope some of you were present for the event. I will be sure to keep you posted on what we did.

2010 Catalog is Here!



The onset of a new year means Beacon Athletics has a new catalog in the works. We know customers look forward to receiving their new copy of Beacon's annual product catalog but sometimes something happens and catalogs don't make it. It may be as simple as a broken link or a file that is too large.

[get_avatar\(\)](#)

This function has some useful parameters for setting the source, size, and more for the gravatar to be displayed. Visit the Codex to learn more:

<http://digwp.com/u/627>

1.1.2 Adding a Theme Options Panel

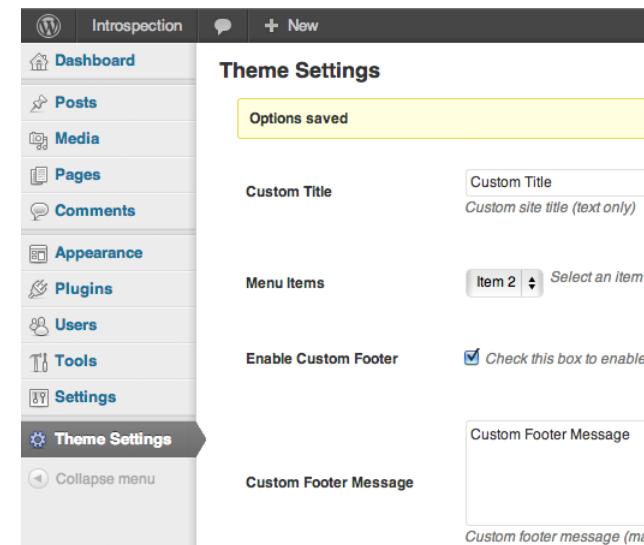
Both of the themes that are now bundled with this book feature theme options panels. Theme options panels are trendy right now, and some themes cross the line of providing functionality that should probably be left to plugins. In general, however, theme options panels are a great way to extend the functionality of a theme. They can make the theme easier to customize should the theme be for sale or control ultimately given to a client, easier for beginners to change things, and even easier for yourself to pop in and change things from an easy interface rather than wrangling code.

What is a theme options panel? It is a new option that shows up in the WordPress Admin menus. Clicking it takes you to the theme options panel, where a variety of options are available. What options? Could be anything. Text boxes to control areas of text in your theme. Dropdowns to select behavioral options, checkboxes... you name it. It is completely up to the theme author of what to offer here and how those options affect the theme.

Both of the bundled themes not only come with a theme options panel, but it is built in such a way that it is almost like a theme options framework. In other words, adding your own new options or altering the existing ones is fairly easy. Theme options panels are built with code that lives in the functions.php file of a theme. This code does a lot, including:

- Adds the options page and its menu item/link
- Ensures that submitted values are properly saved to the database; provides the user with feedback (success message, error message, etc.)
- Makes it easy to add additional options

Then once that is all in place, the theme should be able to access its options easily. Here are the functions we will need at the most basic level:



```

/*
Simple Theme Options Panel
Learn more @ http://codex.wordpress.org/Settings_API
*/
// register settings
function theme_settings_init(){
    register_setting('theme_settings', 'theme_settings');
}
add_action('admin_init', 'theme_settings_init');

// add settings page
function add_settings_page() {
    add_menu_page(__('Theme Settings'), __('Theme Settings'), 'manage_options', 'settings', 'theme_settings_page');
}
add_action('admin_menu', 'add_settings_page');

// variables for dropdown menu
$menu_items = array('Item 1', 'Item 2', 'Item 3');

// theme settings page
function theme_settings_page() {
    global $menu_items; ?>

    <!-- markup for the settings page -->
<?php }
// validate & sanitize input
function options_validate($input) {
    global $menu_items;

    // plain text
    $input['custom_title'] = wp_filter_nohtml_kses($input['custom_title']);

    // dropdown
    if (!isset($input['menu_items'])) $input['menu_items'] = null;
    if (!array_key_exists($input['menu_items'], $menu_items)) $input['menu_items'] = null;

    // checkbox
    if (!isset($input['enable_footer'])) $input['enable_footer'] = null;
    $input['enable_footer'] = ($input['enable_footer'] == 1 ? 1 : 0);

    // html/text
    $input['custom_footer'] = wp_filter_post_kses($input['custom_footer']);

    return $input;
}

```

This code is very stripped down, and is just meant to show you the very basics and the different functions necessary for getting this started. The best way to understand how it works is to just slap it into any theme that doesn't already have its own settings page. Once you've got the code in your theme's functions.php file, there is one more snippet to add: the "markup for the settings page", as indicated by the HTML comment in the previous code example. To complete your new settings page, replace that line with this block of code (continues next page):

```
<style type="text/css">
    th, td { padding: 20px; text-align: left; }
    label { color: #777; font-style: italic; }
    label, input, select, textarea { float: left; clear: both; }
    .adjacent label, .adjacent input, .adjacent select { clear: none; margin-right: 5px; }
</style>

<h2><?php _e('Theme Settings'); ?></h2>

<?php if (isset($_GET['settings-updated'])) : ?>
<div class="updated"><p><strong><?php _e('Options saved'); ?></strong></p></div>
<?php endif; ?>

<form method="post" action="options.php">
    <?php settings_fields('theme_settings'); ?>
    <?php $options = get_option('theme_settings'); ?>
    <table>
        <tr>
            <th scope="row"><?php _e('Custom Title'); ?></th>
            <td>
                <input id="theme_settings[custom_title]" name="theme_settings[custom_title]" type="text"
value="<?php esc_attr_e($options['custom_title']); ?>" />
                <label for="theme_settings[custom_title]"><?php _e('Custom site title (text only)'); ?></label>
            </td>
        </tr>
        <tr>
            <th scope="row"><?php _e('Menu Items'); ?></th>
            <td class="adjacent">
                <select name="theme_settings[menu_items]">
                    <?php foreach ($menu_items as $option) { ?>
                        <option <?php if ($options['menu_items'] == $option) echo 'selected="selected"'; ?>><?php
echo htmlentities($option); ?></option>
                    <?php } ?>
                </select>
                <label for="theme_settings[menu_items]"><?php _e('Select an item'); ?></label>
            </td>
        </tr>
        <tr>
            <th scope="row"><?php _e('Enable Custom Footer'); ?></th>
            <td class="adjacent">
                <input id="theme_settings[enable_footer]" name="theme_settings[enable_footer]" type="checkbox"
value="1" <?php checked('1', $options['enable_footer']); ?> />
            </td>
        </tr>
    </table>
</form>
```

```

        <label for="theme_settings[enable_footer]"><?php _e('Check this box to enable
the custom footer message.');?></label>
        </td>
    </tr>
    <tr>
        <th scope="row"><?php _e('Custom Footer Message'); ?></th>
        <td>
            <textarea id="theme_settings[custom_footer]" name="theme_settings[custom_
footer]" rows="5" cols="30"><?php esc_attr_e($options['custom_footer']); ?></textarea>
            <label for="theme_settings[custom_footer]"><?php _e('Custom footer message (may
use text and markup).');?></label>
        </td>
    </tr>
</table>
<p><input name="submit" id="submit" value="Save Changes" type="submit"></p>
</form>

```

Once the two pieces of code are put together in your functions file, you should see a new “Theme Settings” link in the admin menu (see screenshot at the beginning of this section). Visiting that link will take you to the **Theme Settings** page, where four types of inputs are displayed:

- Text input (specify custom title)
- Dropdown menu (choose an item)
- Checkbox (enable the custom footer message)
- Textarea (specify custom footer message with text and/or HTML)

What about radio boxes? We could do that... but if you think about it, dropdowns are essentially the same thing, just presented with slightly different markup.

The code for adding settings pages to your theme may look complicated, but it makes a lot of sense once you familiarize yourself with the different functions. Once you dig into the code, you’ll see that adding new theme settings involves just a couple of steps: **1)** add a new form input to the `theme_settings_page` function, and **2)** add one or two lines (depending on input type) to the `options_validate` function. And there is no limit to the number and variety of settings that may be added to your theme, so roll up your sleeves and explore the possibilities!

Stay current

Creating settings is one of those aspects of WordPress that changes constantly, so don’t be surprised if you find numerous ways of doing the same thing. And if you’re ever unsure or want more details, just pay a visit to the official documentation at the Codex:

<http://digwp.com/u/628>

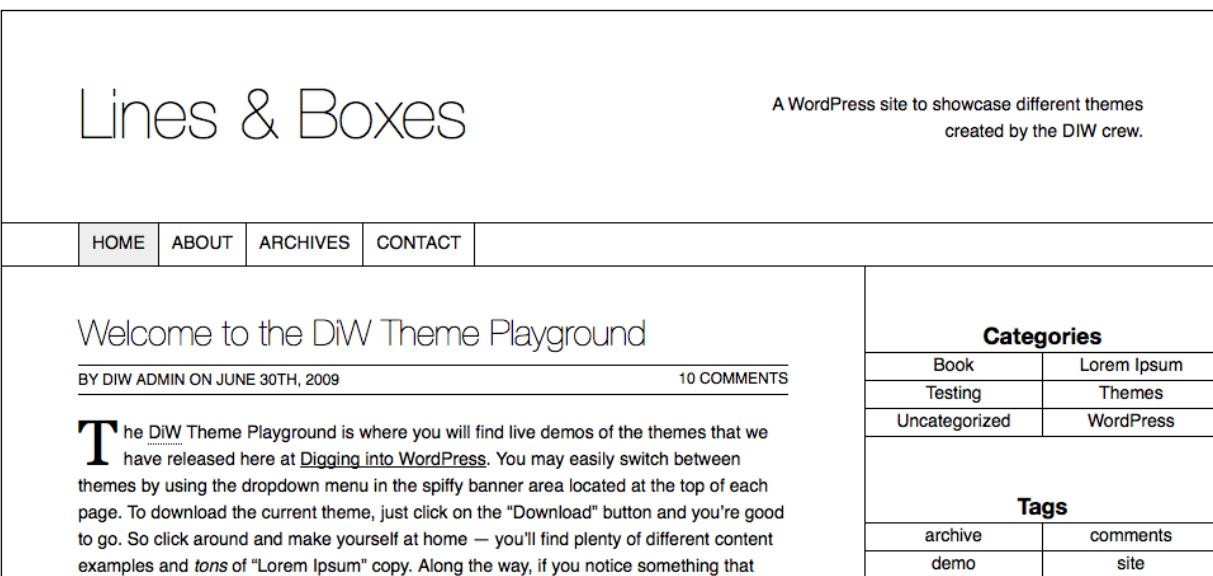
There you will also discover many ways to go further with settings pages and customize them to fit your exact needs.

1.2 Free WP Theme: Lines & Boxes

Have you ever drawn up wireframes for a website and then thought “this website looks good just like this”? That was the inspiration behind “Lines & Boxes”. It can be used as-is, as an extremely simple let’s-focus-on-the-content kind of theme. Or, it can be used as a “blank” style theme, a theme with enough styling in place where it makes customizing it an easier process.

1.2.1 Child Themes

The idea behind a child theme is that it uses all the theme files from a different theme (the “parent” theme), but uses the style.css and functions.php file from itself. This is particularly useful if, for example, you are using a WordPress theme framework. The framework releases updates to everyone using it. If you had that theme completely customized, upgrading would be difficult. If instead you used a child theme and kept your customizations to that, you could simply over-write the parent theme (the framework) with the new version and (probably) be fine.



The screenshot shows the homepage of a WordPress site using the 'Lines & Boxes' theme. The header features the theme's name 'Lines & Boxes' in a large, serif font. Below the header is a sub-header with the text 'A WordPress site to showcase different themes created by the DiW crew.' The main navigation menu includes links for HOME, ABOUT, ARCHIVES, and CONTACT. The main content area displays a single post titled 'Welcome to the DiW Theme Playground', dated June 30th, 2009, with 10 comments. The post content is a block of text explaining the theme's purpose and how to use it. To the right of the content are two sidebar boxes: 'Categories' and 'Tags'. The 'Categories' box lists 'Book', 'Lorem Ipsum', 'Testing', 'Themes', and 'Uncategorized'. The 'Tags' box lists 'archive', 'comments', 'demo', and 'site'.

Lines & Boxes

A WordPress site to showcase different themes created by the DiW crew.

HOME ABOUT ARCHIVES CONTACT

Welcome to the DiW Theme Playground

BY DIW ADMIN ON JUNE 30TH, 2009 10 COMMENTS

The DiW Theme Playground is where you will find live demos of the themes that we have released here at [Digging into WordPress](#). You may easily switch between themes by using the dropdown menu in the spiffy banner area located at the top of each page. To download the current theme, just click on the “Download” button and you’re good to go. So click around and make yourself at home — you’ll find plenty of different content examples and *tons* of “Lorem Ipsum” copy. Along the way, if you notice something that

Categories	
Book	Lorem Ipsum
Testing	Themes
Uncategorized	WordPress

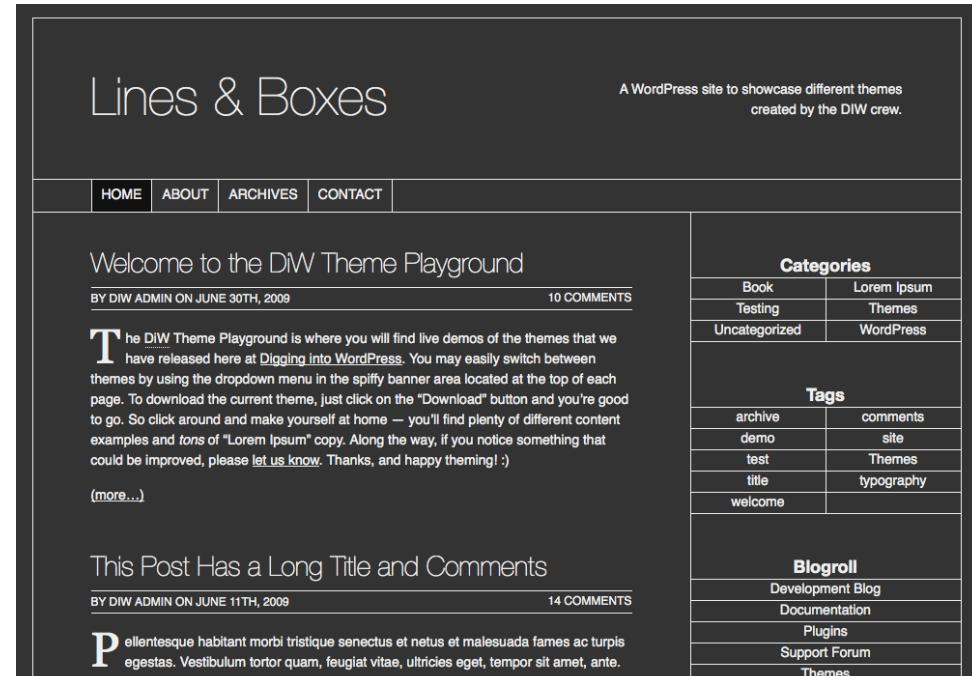
Tags	
archive	comments
demo	site

The other golden opportunity for child themes is making variations upon a base theme. That is exactly what we have done with Lines & Boxes. The original Lines & Boxes is a black-on-white theme. The background is light, the content and “lines” are dark. To illustrate the idea of child themes and variations upon themes, we provided a child theme called “Lines & Boxes (dark)” which is an inverse-color theme (light on dark).

Creating the child theme was almost trivially easy. We just created a new folder for the theme. Put inside it a style.css file (and an images folder as we needed one new image).

The style.css file does all the talking:

```
/*
Theme Name: Lines and Boxes (Dark)
Theme URI: http://digwp.com/
Description: Child Theme for Lines and Boxes
Author: Chris Coyier
Author URI: http://chrisc Coyier.net/
Template: Lines-and-Boxes
*/
@import url("../Lines-and-Boxes/style.css");
/* Overwrite CSS rules here */
body { background-color: #333; color: #eee; }
/* Reverse out colors for other things... */
```



The screenshot shows a dark-themed WordPress site. The header features the title 'Lines & Boxes' and a subtext 'A WordPress site to showcase different themes created by the DiW crew.' Below the header is a navigation bar with links for 'HOME', 'ABOUT', 'ARCHIVES', and 'CONTACT'. The main content area displays a post titled 'Welcome to the DiW Theme Playground' with a timestamp 'BY DIW ADMIN ON JUNE 30TH, 2009' and a comment count '10 COMMENTS'. The post content is a block of text starting with 'The DiW Theme Playground is where you will find live demos of the themes that we have released here at Digging into WordPress. You may easily switch between themes by using the dropdown menu in the spiffy banner area located at the top of each page. To download the current theme, just click on the "Download" button and you're good to go. So click around and make yourself at home — you'll find plenty of different content examples and tons of "Lorem Ipsum" copy. Along the way, if you notice something that could be improved, please let us know. Thanks, and happy theming! :)' followed by a link '(more...)'. Another post titled 'This Post Has a Long Title and Comments' is partially visible below it. To the right of the content area is a sidebar with sections for 'Categories' (Book, Testing, Uncategorized), 'Tags' (archive, demo, test, title, welcome, comments, site, Themes, typography), and 'Blogroll' (Development Blog, Documentation, Plugins, Support Forum, Themes).

Inverse Child Theme

The Lines & Boxes child theme sports light text on a dark background, shown here.

Learn more..

For all the details and specifics about creating and working with child themes, hit the WordPress Codex:

<http://digwp.com/u/629>

1.3 AJAXing WordPress: The “All AJAX” Theme

Also included with this book is a theme called “All AJAX” this is based on Lines & Boxes. It is highly experimental at this point, we just thought it was a fun idea and wanted to provide it as a starting point for you ambitious folks.

The idea is that loading content into an area on a page is trivially easy with jQuery:

All AJAX v2.0

Version 2 of the All AJAX theme is available for download in the DigWP Members Area. You can read about it on the blog:

<http://digwp.com/u/630>

```
$("#main-content").load("/contact/ #inside");
```

That tiny bit of code right there would find the element with the ID of main-content, load the contact page of your site, find the element with an ID of inside and grab that content, and chuck it into the main-content element.

In the “All AJAX” theme, that is exactly what we leverage. First it looks for all “internal” links (links to outside websites are ignored). When an internal link is clicked, it goes and grabs the content from that URL and replaces the main content area. This way you can click around the entire site never seeing a single page refresh. Even search!

Again, instead of dropping a ton of code here in the book, we encourage you to go view the code in the theme itself. You can see the JavaScript that powers it in the /js/allajax.js file inside the theme folder itself. Other features:

- Makes all internal links hash-tag links. This means you can click on links and it does change the URL, e.g., <http://your-site.com/#/contact/>. The theme then supports “deep-linking”, in other words, that URL when loaded for the first time will load the contact page properly.
- Search is AJAXed as well.
- Current page highlighting in all navigation is present, using the same .current_page_item class that WordPress uses by default.

1.4 Free WP Theme: Plastique

“Plastique” is one of the new themes that we are including with this update of Digging Into WordPress. The idea is to use the bundled themes as vehicles for sharing some of the awesome things you can do when designing your own. Let’s check out some of the cool things that you can do with the Plastique theme.

1.4.1 Layout Options, Widgets, & Custom Content

The Plastique theme enables you to completely customize the look, feel and functionality of your site. Choose from a number of **layout options** including:

- Single-column, fixed-width
- Two-columns, fixed-width, left sidebar
- Two-columns, fixed-width, right sidebar
- Three-columns, fixed-width, left and right sidebars (shown below)

The screenshot shows a website layout using the Plastique theme. The header features the title "Plastique Theme" and a subdescription "Extremely flexible, fully widgetized, multiple-layout theme". Below the header is a navigation bar with links for "About", "Archives", and "Contact". A sidebar on the left contains a "Site Search" field and a "Pages" menu with links to "About", "Archives", and "Contact". The main content area has a yellow header "Welcome to the DiW Theme Clubhouse" and a date "June 30th, 2009" by "Jeff Starr". The content text discusses the theme's features and how to download it. To the right of the content is a "Header Upper-Right (full-width box)" containing the text "This panel is active and ready for you to add some widgets via the WP Admin". A "Calendar" sidebar shows the month of January 2012 with days numbered 1 through 31.

Appearance

Themes

Widgets

Menus

Plastique Options

Editor

Choose Your Widgets

To customize any of Plastique's many widgetized areas, visit the "Widgets" screen from the "Appearance" submenu.

Each layout is fully equipped with a wide variety of "widgetized" areas that may be used to include virtually any type of widgetized functionality imaginable. **Widgetized areas** include the following:

- Four header panels
- Left and right sidebars
- Before and after posts
- Before comments
- Three footer panels

Plus several other widgetized areas that are exclusive to particular layout options. In addition to the layout options and widget functionality, Plastique also includes a multitude of **custom-content** areas throughout the design. These custom-content areas enable you to include virtually any content or markup into these locations:

- The `<head>` section
- Header area (top of page)
- After-header area
- Center column
- Left and right sidebars
- Footer area, multiple options
- Before closing `</body>` tag

Full Admin Control

All of the layout and custom-content options are easily controlled via the "Plastique Options" screen in the Admin.

Additionally, each of these different custom-content areas and widgetized areas may be toggled on or off in the Admin area, enabling you to display or hide anything you wish. Everything is completely customizable via the **Plastique Theme Settings** in the WordPress Admin area.

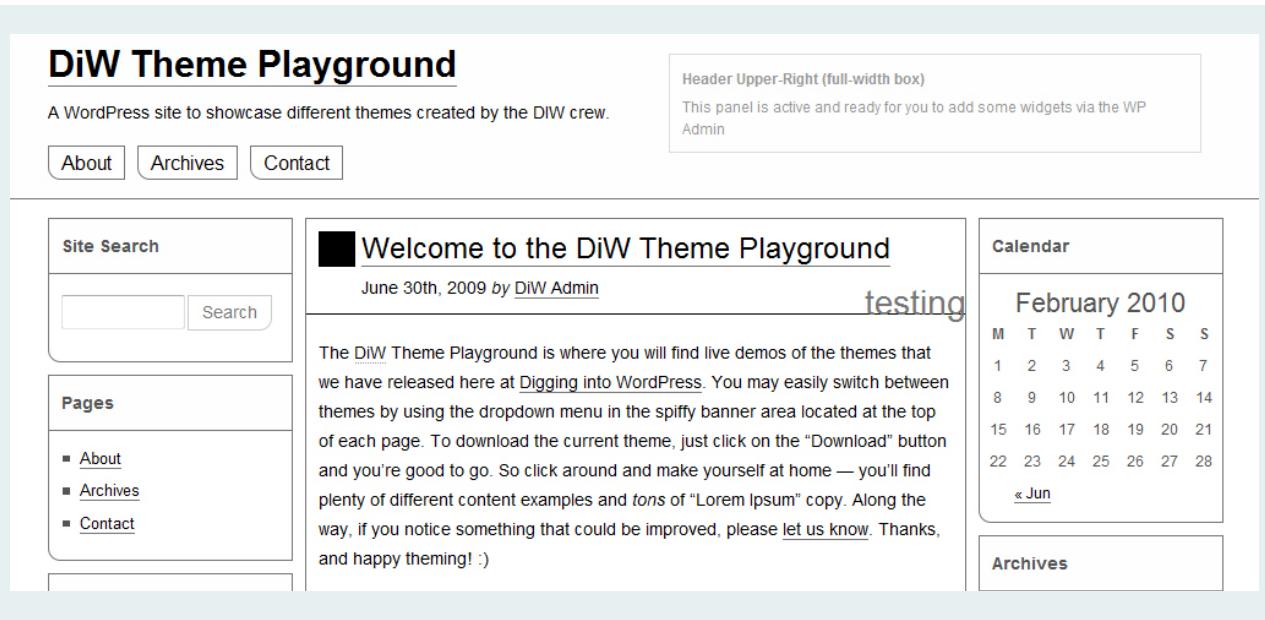
1.4.2 Child Themes and Category Styles

Out of the box, Plastique includes category-specific post styles, including unique "mini themes" for each different category of post. These post-specific styles feature the category name displayed in the post header and color coordinated link, border, and background styles.

Of course, the default pastel color-scheme may not be for everyone, so we've made it relatively easy to change things up with the included Plastique child theme, B&W.

Plastique's B&W child theme includes everything you need to customize the appearance by specifying your own CSS styles. B&W may be either used as is, or as a template for implementing your own custom styles.

The Plastique theme also features some slick comment styles, which by default includes support for "two-level" deep threaded comments. This is one of our favorite parts of Plastique's design, and is demonstrated live at our **Theme Clubhouse**: <http://digwp.com/u/437>



The screenshot shows a WordPress site with a light gray header and footer. The main content area has a white background with a grid layout. The top-left grid cell contains a "Site Search" form with a "Search" button. The top-right grid cell is a "Header Upper-Right (full-width box)" containing a note about adding widgets via WP Admin. The middle-left grid cell is a sidebar with a "Pages" menu and links to "About", "Archives", and "Contact". The main content area features a large "Welcome to the DiW Theme Playground" section with a "testing" watermark, a "Calendar" section for February 2010, and an "Archives" section.

DiW Theme Playground

A WordPress site to showcase different themes created by the DiW crew.

About Archives Contact

Header Upper-Right (full-width box)

This panel is active and ready for you to add some widgets via the WP Admin

Site Search

Pages

- [About](#)
- [Archives](#)
- [Contact](#)

Welcome to the DiW Theme Playground

June 30th, 2009 by DiW Admin

The DiW Theme Playground is where you will find live demos of the themes that we have released here at [Digging into WordPress](#). You may easily switch between themes by using the dropdown menu in the spiffy banner area located at the top of each page. To download the current theme, just click on the "Download" button and you're good to go. So click around and make yourself at home — you'll find plenty of different content examples and tons of "Lorem ipsum" copy. Along the way, if you notice something that could be improved, please [let us know](#). Thanks, and happy theming! :)

testing

February 2010

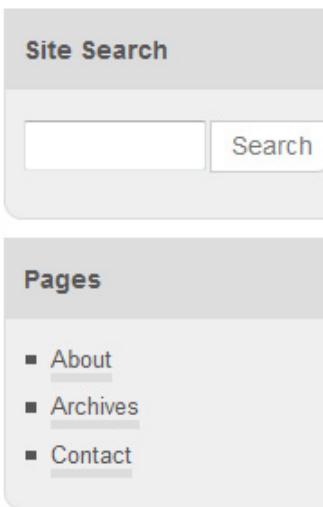
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

« Jun

Archives

Child's Play

Customize the Plastique theme by dropping in the B&W child theme and then rocking your own styles.



1.4.3 Other Plastique Features

Wrapping up, here are a few more awesome features tucked into Plastique:

- Automatic inclusion of the category ID included for each post and page view via the `post_class` and `body_class` template tags
- Automatic inclusion of WordPress' various feed links in the `<head>` section
- Smart inclusion of jQuery via `wp_enqueue_script()`

Custom Widget Styles

WordPress' default widgets have been pre-styled with matching colors and fonts for the Plastique theme. These custom styles may be easily overridden using a child theme.

That's it for this round of Bonus Tricks — stay tuned to the book's companion website, [DigWP.com](http://digwp.com) for more bonus material, articles, tutorials, and much more. For more WordPress themes, check out the [DigWP Theme Clubhouse](http://digwp.com/theme-clubhouse).

Plastique Theme Options

Layout Options

Three Columns

Customize number of columns for your site

Header Layout

Two Rows, Two Columns

Customize the layout of your Header

Header Menu



Display a Page menu in the header area? (requires one of the 'Two-Row' layouts)

Plastique Options

Detail view of the Plastique Options page, where you have full control over number of layout columns, menu items, custom content, header & footer layout, and everything else.

Plastique's Category Posts Widget

In addition to the many default widgets included with WordPress, the Plastique theme includes the James Lao's excellent Category Posts Widget <http://digwp.com/u/434>. The Category Posts Widget makes it easy to display the most recent posts from a certain category anywhere within your theme.

The widget includes plenty of great features, including optional display of comment count, post date, post excerpt, post thumbnails, and more. To use the Category Posts Widget, simply drag & drop it into any of Plastique's widgetized areas and customize the options to suit your needs.

Do not go where the path may lead, go instead where there is no path and leave a trail.

— RALPH WALDO EMERSON

2

WordPress 2.9 Update

2.1 Like a River...

WordPress is likely the most coordinated, focused, and fast-moving open source project on the planet. When bugs in WordPress are found, the community and core development team are usually quickly on the case and push patches out to take care of it. But WordPress releases aren't just bug patches. The "point releases" (e.g., from 2.8 to 2.9) typically represent significant changes.

These changes could be new functions available for theme builders, changes to how those functions work or what they return, aesthetic or functionality improvements to the Admin area, etc. And it's not always "add, add, add."

New versions sometimes simplify pre-existing things, which is the mark of truly great software.

Frequent updates make writing books about this software challenging. Fortunately, you made the right choice as this book will be continually updated to document those changes.

2.2 New in WordPress 2.9

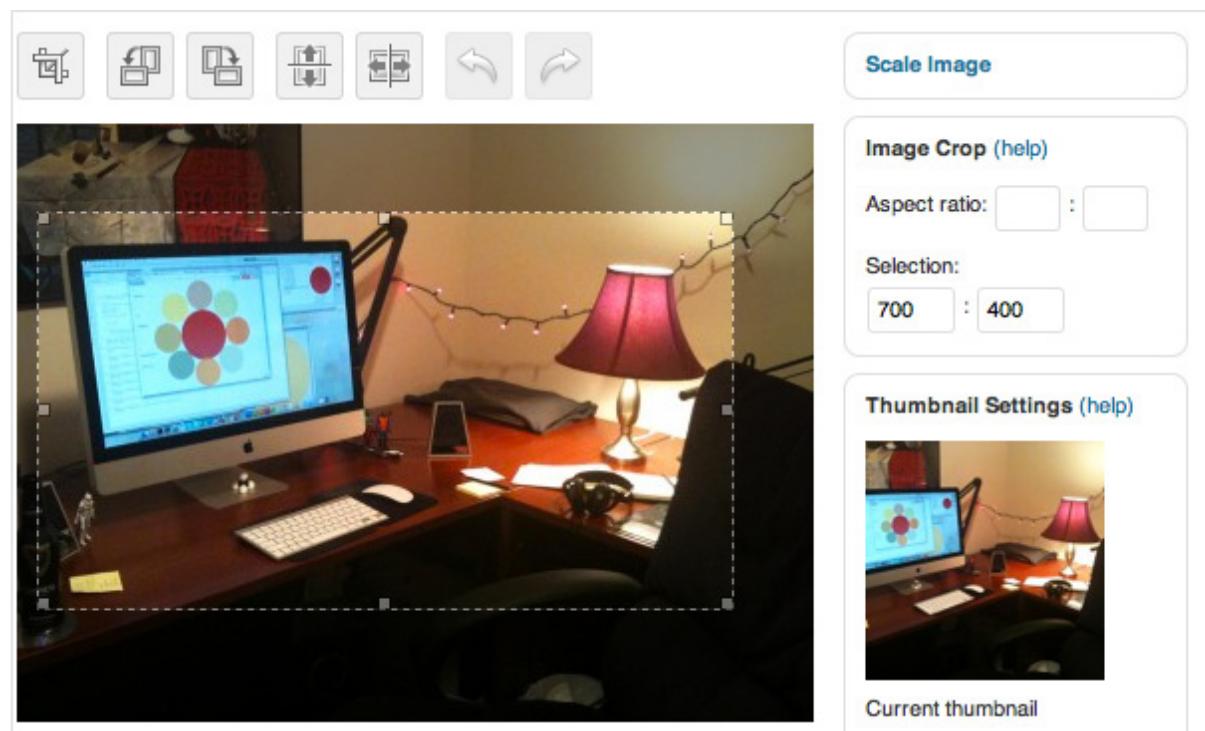
WordPress 2.9 was a fairly major release for WordPress, and brings with it a good number of new backend functions, as well as Admin area functions. WordPress users now enjoy a built-in image editor, undo/trash functionality, batch-updating of plugins, dead-simple video embedding, and tons more. Let's explore these great new features to help you get the most out of the latest version of WordPress.

2.2.1 Image Editor

WordPress' Media Library now features basic image editing. This is a real time-saver for users who need to make simple changes such as rotation, scaling, flipping, and cropping. To facilitate the editing process, the Media Library also includes "undo" and "redo" functionality, aspect-ratio adjustments, pixel-coordinate information, and optional bulk editing of all thumbnails.

How to use:

1. Go to the **Media Library** and click on "Add New" at the top of the screen
2. In the **Upload New Media** screen, select and upload your image
3. Next to the thumbnail of your image, click the "Edit image" button
4. Use the tools in the **Image Editor** to fix up your image



2.2.2 Trash Can

The new “Trash Can” is similar in functionality to Mac’s “Trash” and Window’s “Recycle Bin,” and serves as an intermediate safety net between your content and final deletion. Instead of simply deleting a post, page, draft – whatever – you now send it to the Trash Can, where it will remain until restoration or deletion, whichever happens first. This new “trash” functionality applies to anything that can be deleted, including posts, pages, attachments, comments, drafts, and so on.

By default, WordPress empties the Trash Can every 30 days. During this time, users may restore any trashed items or delete them permanently. If 30 days is not an ideal time period, modify it with the following code in your `wp-config.php` file:

```
define('EMPTY_TRASH_DAYS', 10); // empty trash every 10 days
```

Simply edit the “10” with the desired number of days and you’re good to go. If you would rather not have to deal with the whole Trash Can scene, use this instead:

```
define('EMPTY_TRASH_DAYS', 0); // disable trash can feature
```

How to use:

To view the contents of the Trash Can, go to the Edit Posts screen and click on the “Trash” link. If you would like to restore an item, click on its corresponding “Restore” link. Multiple items may be restored easily via the bulk restore options.

The screenshot shows the WordPress “Trash” screen. At the top, there are navigation links: “All (1)”, “Published (1)”, and “Trash (2)”. Below these are several buttons: “Delete Permanent”, “Apply”, “Show all dates”, “View all categories”, “Filter”, and “Empty Trash”. The “Empty Trash” button is highlighted with a red arrow and the text “empty all trash”. The “Apply” button is also highlighted with a red arrow and the text “restore/delete selected”. Below the buttons is a table listing items in the trash. The first item is “Title”, checked, with “More Trash”, “Restore”, and “Delete Permanently” options. The second item is “Deleted Post”, checked, with “More Trash”, “Restore”, and “Delete Permanently” options. The “Author” column shows “DigWP” for both items. The “Empty Trash” button is located at the top right of the table area, with a red arrow pointing to it and the text “empty all trash”. The “Restore” and “Delete Permanently” buttons for individual items are also highlighted with red arrows and the text “restore/delete individually”.

		Author
<input checked="" type="checkbox"/>	Title	DigWP
<input checked="" type="checkbox"/>	More Trash	DigWP
	Restore Delete Permanently	
<input checked="" type="checkbox"/>	Deleted Post	DigWP
	Restore Delete Permanently	

2.2.3 Embedding Videos with oEmbed

WordPress now makes it super-easy to embed video content in your posts. Before version 2.9, embedding video required users to format and include a customized slab of markup, which would require different parameters and settings depending on different types of video formats. Now, all that's needed is inclusion of the video URL in the post. Amazingly simple.

WordPress makes this possible by using the oEmbed API (oembed.com), which enables embedded video content on third-party sites. Of course, auto-embedding requires that the original video content is hosted on a site that supports oEmbed functionality.

Once you publish your video post, WordPress checks the video site for the required oEmbed auto-discovery code. If it's found, WordPress automatically converts the video URL into the required embed code.

By default, oEmbed is enabled. To disable oEmbed functionality, go to **Settings > Media** and uncheck "Attempt to automatically embed all plain text URLs".

How to use:

1. Create a new post
2. Get the URL of your video
3. Paste your video URL on its own line
4. Publish your post

If you need to include other content on the same line as the video URL, wrap the URL in an `[embed]` shortcode:

```
[embed]http://www.youtube.com/watch?v=Ms3QdGIzltU[/embed]
```

WordPress will automatically embed the video specified using this format.

Supported Video Sites

By default, for security purposes, WordPress will only embed URLs that match a built-in whitelist. Here are some currently supported sites:

- [YouTube](#)
- [Vimeo](#)
- [DailyMotion](#)
- [blip.tv](#)
- [Viddler](#)
- [Hulu](#)
- [Revision3](#)
- [Scribd](#)
- [Google Video](#)
- [Flickr](#) (both videos and images)
- [WordPress.tv](#) (currently works for only [VideoPress](#)-type videos)

Embeds

Auto-embeds

When possible, embed the media content from a URL directly onto the page.

Maximum embed size

Width Height

Enable/Disable Auto-Embedding for Videos

You have full control over whether or not you would like WordPress to automatically embed your videos. Just go to “Settings > Media” and check or uncheck the box.

For information on including additional websites to the oEmbed whitelist, see the official “Embeds” documentation at the WordPress Codex: <http://digwp.com/u/426>

2.2.4 Database Maintenance Tools

WordPress 2.9 makes it easier to optimize and repair your database from within the Admin area. This is especially useful for users who aren't already using some sort of a database-management plugin such as the excellent WP-DBManager: <http://digwp.com/u/443>

By including the following directive in your configuration file (wp-config.php), you



Canonical Tags for Previous Versions of WordPress

To implement canonical tags in older versions of WordPress, simply add the following slice of code to the `<head>` section of your theme's `header.php` file:

```
<?php if (is_singular()) echo '<link rel="canonical" href="'.get_permalink().'>'; ?>
```

This will add the `rel="canonical"` tag to all of your single posts and pages.

engines sort through the many different types of derivative and/or duplicate content and give proper credit where it's due.

How to use:

If the `wp_head()` hook is present in your theme's `header.php` template, WordPress automatically includes canonical tags on your single posts and pages. If you would like to disable the canonical-tag feature, simply add this code to your theme's `functions.php` file:

```
remove_action('wp_head', 'rel_canonical'); // disable canonical tags
```

2.2.6 Featured Images (Post Thumbnails)

A much-anticipated feature is WordPress' new post-thumbnail functionality. Adding post-specific thumbnails has always been possible using custom fields, but now the `the_post_thumbnail` template tag simplifies the process. Here is an example of how the tag is used in your theme files:

```
<?php while(have_posts()) : the_post(); ?>
  <h1><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h1>
  <?php the_post_thumbnail(); ?>
  <?php the_content(); ?>
<?php endif; ?>
```

[Learn More...](#)

For more information on `rel="canonical"` and duplicate content, see Chapter 8.2.5 in the main book.

[Name Changer](#)

When post thumbnails were added to WP 2.9, they were referred to as just that: "Post thumbnails." Somewhere along the way, the name was changed, and is now referred to as "Featured images."

Ahh, much better ;)

With this code in place, each post will display its post thumbnail (if available) as specified via the Admin. Once post-thumbnail functionality is activated on your site, the **Write/Edit-Post** screen will display a **Featured Image** panel that will enable you to assign a thumbnail for that particular post.

How to activate/use:

1. Activate post-thumbnail/featured-image functionality by adding the following code to your theme's functions.php file:

```
<?php if (function_exists('add_theme_support'))  
{ add_theme_support('post-thumbnails'); } ?>
```

2. Go to the **Write/Edit-Post** screen and click "Set featured image" in the "Featured Image" panel.
3. Once the media panel opens, go to the "Media Library" tab and select your thumbnail image.
4. After selecting the thumbnail image, click "Use as Thumbnail" next to the "Insert into Post" button.
5. That's it. Your selected image should now appear in the **Post Thumbnail** panel and may be displayed in your theme using this tag in the post loop:

```
<?php the_post_thumbnail(); ?>
```

You may also specify the size of the displayed thumbnail by adding a size parameter like so:

```
<?php the_post_thumbnail('thumbnail'); ?>  
<?php the_post_thumbnail('medium'); ?>  
<?php the_post_thumbnail('large'); ?>
```



All Types | **Image** (1)

Show all dates Filter »



IMG_1594-edit-web



File name: IMG_1594-edit-web.jpg

File type: image/jpeg

Upload date: January 13, 2012

Dimensions: 800 x 600

Edit Image

Title

*

IMG_1594-edit-web

Insert into Post

Use as featured

Post Thumbnail Tags

In addition to the_post_thumbnail() template tag, the following may also be used in your theme files:

has_post_thumbnail()
get_post_thumbnail_id()
get_the_post_thumbnail()

If no size parameter is specified, your default thumbnail size is used. You may also specify sizes explicitly via your functions.php file:

```
set_post_thumbnail_size(50, 50); // 50px resized thumbnails  
set_post_thumbnail_size(50, 50, true); // 50px cropped thumbnails
```

You can also specify custom thumbnail sizes as follows:

```
add_image_size('custom-thumbnail', 500, 500); // custom 500px thumbnail
```

...which would then be included in your theme loop like so:

```
<?php the_post_thumbnail('custom-thumbnail'); ?>
```

To style your thumbnail images, you may either use the provided CSS class:

```
.wp-post-image {}
```

...or else add your own:

```
<?php the_post_thumbnail(array('class'=>'my-custom-class')); ?>
```

In addition to these customizations, there is much more that may be done with post thumbnail functionality. Here are some examples..

Override default thumbnail dimensions (in pixels):

```
<?php the_post_thumbnail(array(250,250)); ?>
```

Specify additional image attributes (class, title, alt):

```
<?php the_post_thumbnail('medium', array('class'=>'custom-class', 'alt'=>'custom-alt',  
'title'=>'custom-title')); ?>
```

We can also ensure backwards compatibility for older versions of WordPress by wrapping the functions.php code in a conditional check:

```
if (function_exists('add_theme_support')) { // as of WP 2.9
    add_theme_support('post-thumbnails'); // activate thumbnails
    set_post_thumbnail_size(50, 50, true); // specify thumbnail size
    add_image_size('custom-thumbnail', 500, 500); // custom thumbnail size
}
```

...and then we'll also want to wrap the template code in a conditional check:

```
if ((function_exists('has_post_thumbnail')) && (has_post_thumbnail())) {
    the_post_thumbnail();
} else {
    $postimageURL = get_post_meta($post->ID, 'post-image', true);
    if ($postimageURL) {
        echo '';
    }
}
```

More information on post thumbnails

Check out these excellent resources for more about post thumbnails:

- New in WordPress 2.9: Post Thumbnail Images: <http://digwp.com/u/438>
- The Ultimative Guide For the_post_thumbnail: <http://digwp.com/u/494>
- Everything you need to know about the post image feature: <http://digwp.com/u/440>

2.2.7 Metadata API

WordPress 2.9 features a new Metadata API that eliminates redundancy and makes it easier to retrieve data from meta tables in the database. The `add_metadata()` function accepts the following parameters:

```
add_metadata($meta_type, $object_id, $meta_key, $meta_value, $unique);
```

This enables us to access just about any piece of data in the database and display it within our posts and comments. For example, displaying a specific user's email address is as easy as this:

```
add_metadata('user', 123, 'email', 'user@domain.tld');
```

As awesome as this is, the Codex recommends these more specific functions when creating or customizing themes and plugins:

- `add_post_meta()` – for posts, pages, and custom post types
- `add_user_meta()` – for user information
- `add_comment_meta()` – for comments

Collect 'em all

In addition to `add_metadata()`, the Metadata API includes these fine functions:

`update_metadata()`
`delete_metadata()`
`get_metadata()`
`update_meta_cache()`

Learn More...

The new Metadata API functionality is fully documented in the core at: /wp-includes/meta.php

..and also at the Codex:

<http://digwp.com/u/631>

2.2.8 Widgetized Sidebar Descriptions

This is a great new feature for theme designers that enables the addition of a custom description for each widgetized area. This makes it much easier for users to determine the location and purpose of the different widgetized areas present within a theme.

All that's needed is an additional "description" argument placed within the `register_sidebar()` function:

```
if (function_exists('register_sidebar')) {  
    register_sidebar(array(  
        'name'          => 'Custom Sidebar',  
        'id'            => 'custom',  
        'description'   => 'Custom Sidebar Description',  
        'before_widget' => '<li id="%1$s" class="widget %2$s">',  
        'after_widget'  => '</li>',  
        'before_title'  => '<h2 class="widgettitle">',  
        'after_title'   => '</h2>',  
    ));  
}
```

Just change the “description” argument to whatever you would like. It’s *that* easy.

2.2.9 Custom Post Types

WordPress now makes it easier for designers and developers to create custom post types. Up until WordPress 2.9, using anything other than the default four post types — “post”, “page”, “revision”, and “attachment” — was quite a chore requiring custom code and much understanding.

With the ability to register custom post types, WordPress opens the doors to easier post formatting, better content structuring, and greater design flexibility. While there is no user-interface for custom post types in WordPress 2.9, there are plans for better integration in version 3.0.

Even without an Admin interface, designers and developers may create custom post types and integrate them into themes using the `get_post_type()` function, which returns the “type” of post when used within the loop. There is also the `get_post_types()` function, which returns a list of all “types” of posts.

To create a custom post type, we have the `register_post_type()` function, which defines the name of the custom post type in the first argument. The second parameter is an array that specifies additional parameters. See the Codex for more information: <http://digwp.com/u/441>

[Learn More...](#)

See Chapter 3.2.7 for more information on WordPress' new custom-post (aka custom-content) types.

2.2.10 New Theme Templates

As avid theme designers, one of our favorite new WordPress features is the enhanced custom-template functionality. WordPress now makes it much easier to create custom theme templates based on slug information. Now we can do custom templates using files named with the following syntax:

- category-slug
- page-slug
- tag-slug

These are in addition to these existing formats:

- category-id
- page-id
- tag-id

To use, simply append the slug name to the end of the tag, page, or category filename and that template will be used to display the output for that particular set of posts. So for example, if a custom template named "category-hamsters.php" will be used to display all of the posts from the category with the "hamsters" slug.

2.2.11 Register Feature Support

This new feature is good news for plugin developers, who may now declare support for an existing feature by adding the `add_theme_support()` function to the

theme's functions.php file. Here it is used for the post_thumbnail function:

```
add_theme_support('post-thumbnails');
```

Additional functionality may be included via the require_if_theme_supports() tag:

```
require_if_theme_supports('custom-feature', '/path/to/custom.php');
```

Once a new feature has been declared, theme support is implemented with the following code via functions.php:

```
if (function_exists('add_theme_support')) { add_theme_support('custom-feature'); }
```

Notice that we provide additional protection by checking (via function_exists()) for support of the add_theme_support() function. This is good practice for backwards-compatibility.

2.2.12 Custom Theme Directories

WordPress 2.9 also features the ability to register custom theme directories. By default, the theme directory is located at "/wp-content/themes/". To specify an alternate theme directory, such as one at "/wp-content/alt-themes/", we use the new register_theme_directory function and specify the custom theme path (relative to the wp-content directory):

```
<?php register_theme_directory('alt-themes'); ?>
```

With this code in your functions.php file, WordPress will register your custom theme directory and scan it for any available themes. Any themes that are discovered will be presented in the WordPress Admin's "Presentation" screen.

This useful new feature enables plugin developers to automatically add themes without any action by the user. Another potential use for custom theme directories involves sharing themes among different installations of WordPress.

2.2.13 Other Cool Changes in Version 2.9

Most of this stuff is good news especially for developers, but it's also good for users to understand some of WordPress' new potential and possibilities.

- **Application Upgrades** - TinyMCE and SimplePie have been upgraded for better performance.
- **Batch Plugin Updates** - You can now check compatibility and update multiple plugins at the same time.
- **New Excerpt Filters** - Two new filters enable you to change the default text and length of excerpts.
- **User Registration** - User Profiles and Registration now may be filtered to collect additional user information.
- **Widgets Anywhere** - Widgets may now be called anywhere thanks to the new widget functionality.
- **New sanitization API** - New tools available for cleaning and sanitizing code using functions such as `esc_html()`.
- **Custom taxonomies** - Custom taxonomies are now included in the WXR export file and imported correctly.
- **Increased MySQL Requirements** - The minimum version of MySQL required for WordPress is increased from 4.0 to 4.1.2.
- **Better Hooks and Filters** - Better hooks and filters for excerpts, smilies, HTTP requests, user profiles, author links, taxonomies, SSL support, tag clouds, query_posts and `WP_Query`.

More Information

Of course, WordPress 2.9 included many more updates, changes, fixes and patches. For a more complete list, check out the WordPress Codex:

<http://digwp.com/u/442>

Don't try and reinvent the wheel –
just work on making it better than
anyone else.

– DAVID A. STUEBE

3

WordPress 3.0 Update

3.1 Giant Leap Forward...

One thing that people love about WordPress are all of the awesome new features rolled out with each new version, and WordPress 3.0 is no exception. WordPress has come a long, long way since it was first introduced way back in May 2003, and the newly released version 3.0 takes WordPress' powerful functionality further than ever before. A giant leap forward, without a doubt.

WordPress 3.0's new features are all about customization and better Content Management System (CMS) capabilities, including awesome stuff like new default themes, MultiSite options, custom post-types, and custom everything else. Whether you need a simple blog or an elaborate CMS, WordPress is the perfect solution. Let's check out some of the latest and greatest new features, and learn how to use them to improve your site's quality, appearance and functionality.

3.2 New in WordPress 3.0

WordPress 3.0 was a major release for WordPress, bringing with it some major behind-the-scenes functionality that improves its content-management capabilities. WordPress users now enjoy custom *everything*, including the ability to control and customize content types, menus, taxonomies, and even multiple sites from the comfort of the WordPress Admin. While everyone will appreciate the new default theme, many designers may miss some of the underlying changes that make WP 3.0 an awesome blogging platform and an even more powerful CMS.

[Themes](#)[Widgets](#)[Menus](#)[Background](#)[Header](#)[Editor](#)

[Customize It!](#)

Twenty Ten is easily customizable using the Widgets, Menus, Background, and Header options.

[Twenty Whatever](#)

The WP team says they will be releasing new default themes every year, and so far they have delivered through “Twenty Thirteen”, which is the current default WP theme.

[Twenty Ten](#)

The Twenty Ten theme (shown here) served as the default WordPress theme through at least version 3.3.

3.2.1 Goodbye Kubrick, Hello Twenty Ten

With version 3.0, WordPress comes bundled with a new default theme. Dubbed “Twenty Ten” (think: 2010), the new WordPress theme is inspired by Ian Stewart of Thematic and is packed with features:

- Two-column layout with widgetized sidebar and footer
- Fresh horizontal dropdown menu system
- Clean typography via Georgia/sans-serif fonts
- Custom background images (with tiling support)
- Custom header image via post-thumbnail functionality
- Built-in Support for microformats
- Strong(er) focus on SEO

Overall, Twenty Ten looks like a solid, well-built theme. The design may not appeal to everyone’s tastes, but it’s definitely a step up from ol’ Kubrick. Check out the new Twenty Ten theme in action after installing WordPress 3.0 (click on the **Appearance** tab in the left sidebar menu). To customize, visit the links shown in the upper-left screenshot (Widgets, Menus, Backgrounds, and Header options).

Digging into WordPress

Learn how to take your skills to the next level.



3.2.2 Goodbye “admin”, Hello Custom Username

Much has been said concerning the default admin username, “admin”, that automatically is chosen for you when installing WordPress. Now with WordPress 3.0, users are no longer required to jump through hoops to specify their own Admin username. A welcome change that benefits everyone.

This new custom-username feature is a *huge* timesaver for WordPress administrators. When setting up WordPress, you can now choose your own username during the installation process, and then change the randomly generated password to something both secure and memorable.

Be unique..

Choose your own custom admin username during the WordPress installation process.

Site Title	Digging into WordPress
Username	CustomUsername
Password, twice	*****

3.2.3 Custom Background Support

WordPress 3.0 also features support for custom background images. Any theme that includes the `wp_head` template-tag will work, but you need to actually enable the custom-background functionality by including the following line of code anywhere within your theme's `functions.php` file:

```
add_theme_support('custom-background');
```

Theme Support

Check out the Codex for a complete list of features that may be added to your theme:

<http://digwp.com/u/632>

Custom Backgrounds

*New in WordPress 3 –
Customize the default theme
and other supportive themes
with a background image.*

[Remove Image](#)

[Remove Background Image](#)

This will remove the background image. You will not be able to restore any customizations.

[Upload Image](#)

Choose an image from your computer:

[Choose File](#) No file chosen

[Upload](#)

Display Options

Position

Left Center Right

Repeat

No Repeat Tile Tile Horizontally Tile Vertically

Attachment

Scroll Fixed

Background Color

#

[Select a Color](#)

Once this code is in place, navigate to your Admin's **Appearance** menu and click on **Custom Background**. There you will find options for positioning, repeating, and if needed, deleting your custom background image.

How does it work? After you have specified your background options in the WP Admin, WordPress generates the CSS rules required to display the background image and outputs the code into your theme's `<head>` section. Here is the CSS code that was generated for our custom background-image on our test WP installation:

```
<style type="text/css">
  body {
    background-image: url('http://digwp.com/bg.jpg');
    background-repeat: no-repeat;
    background-position: top center;
    background-attachment: fixed;
  }
</style>
```

Although they are *not required* for custom-background functionality to work on your site, the following parameters enable you to customize functionality:

```
$defaults = array(
  'default-color'      => '',
  'default-image'      => '',
  'wp-head-callback'  => '_custom_background_cb',
  'admin-head-callback' => '',
  'admin-preview-callback' => ''
);
add_theme_support( 'custom-background', $defaults );
```

For more information on using these custom callback functions, check out Otto on WordPress for a great post: <http://digwp.com/u/454>.

The take-home message for custom-background support is that WordPress now makes it easy to use your favorite background image for your site. *Free and easy!*

Search not Scan

When viewing your source code, this CSS snippet (and most of the other WP-generated code discussed in this book) won't have this idealized formatting. Auto-generated source code usually ends up splattered all over the place. So instead of wasting time scanning through scrambled markup, use your browser's Find feature to search for identifiable portions of code.

Learn more..

Official Codex page for adding the custom background feature:
<http://digwp.com/u/633>

WordPress MU

WPMU enables the running of “hundreds of thousands of blogs with a single install of WordPress.” Indeed, WordPress.com uses WPMU to serve “tens of millions of hits on millions of blogs each day.” Awesomely enough, this same MultiSite functionality is now built into WordPress 3.0.

<http://mu.wordpress.org/>

Network Link

After defining MultiSite in the wp-config.php file, you will see a new “Network Setup” link under the Tools Menu:



3.2.4 WordPress MultiSite: The Merging of WordPress with WPMU

Up until version 3.0, WordPress was a *single-site* platform. Users desiring to setup and run multiple sites were required to use WordPress MU, which enables multi-user (multi-tenant) installations with multiple sites all under one roof. WordPress MU has never been as extensible or widely supported as WordPress itself, so the news that it is merging with the WP core is excellent news for users managing multiple WordPress sites.

WordPress’ new MultiSite functionality is not enabled by default, so single-site users will experience the same 5-minute installation process as before. Whenever you decide to create your own network of sites, or even just add a second site, just walk through the installation steps and you’re basically good to go.

To enable MultiSite, you need to add the following line of code to your site’s wp-config.php file (just above the line that says, “That’s all, stop editing!”):

```
define('WP_ALLOW_MULTISITE', true);
```

After uploading your updated config.php file to the server, a new link will appear in the WP Admin under the Tools menu. Clicking on that link will take you to the “Create a Network” page, where you’ll define a few options and create a network.

Clicking on the Network link in the Tools menu loads the network setup/installation page. It should look like this:

A screenshot of the 'Create a Network of WordPress Sites' page. The page title is 'Create a Network of WordPress Sites'. It features a 'Welcome to the Network installation process!' message and a note: 'Fill in the information below and you'll be on your way to creating a network of WordPress sites. We will create configuration files in the next step.' The page includes a sidebar with icons for Home, Tools, Available Tools, Import, Export, Network Setup (selected), and Settings, and a 'Help' button.

Complete Details

If you’re planning on using MultiSite, this section of the book will show the basics, but you should also read through the Network page at the Codex for a more comprehensive walkthrough.

<http://digwp.com/u/485>

On the network setup page, first decide if you want your network sites to use sub-domains or sub-directories. Note that sub-domain networks are not possible if your WP installation is in a subdirectory. In this case, the choice is made for you:



Sub-directory Install

Because your install is in a directory, the sites in your WordPress network must use sub-directories.

If WP is installed in the root directory, you will have the option of setting up your network using either sub-domains or sub-directories. In general, we advise to go with sub-directory networks whenever possible. It's just easier. After a choice has been made, proceed to the next section to fill in your Network Details:

The screenshot shows the 'Create a Network of WordPress Sites' page. The left sidebar has icons for Home, Site, Network, Plugins, Themes, and Tools. The main content area has a title 'Create a Network of WordPress Sites' and a sub-section 'Welcome to the Network installation process!'. It says 'Fill in the information below and you'll be on your way to creating a network of WordPress sites. We will create configuration files in the next step.' Below this, there's a 'Network Details' section with the following fields:

Sub-directory Install	Because your install is in a directory, the sites in your WordPress network must use sub-directories.
Server Address	The internet address of your network will be <code>redfeed.net</code> .
Network Title	<input type="text" value="Digging into WordPress Sites"/>
Admin E-mail Address	<input type="text" value="email@email.com"/>

At the bottom is a blue 'Install' button.

Using Subdomains...

...is quite a bit trickier than using subdirectories. Visit the WordPress Codex to chew through all the gory details:

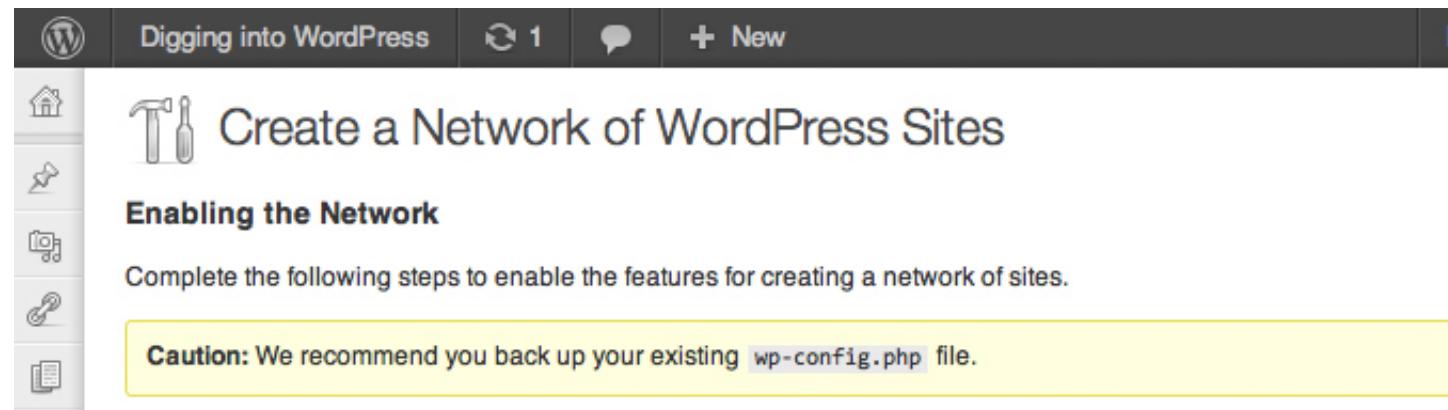
<http://digwp.com/u/484>

Top-Level Domains

Instead of using subdomains or subdirectories for your multiple sites, it's possible to use top-level domains:

<http://digwp.com/u/482>

Very straightforward so far. Once you enter your Network Title and Admin Email Address, click the “Install” button to create the network! As soon as it’s complete, you’ll see the “Enabling the Network” page:



Digging into WordPress

1

+ New

Create a Network of WordPress Sites

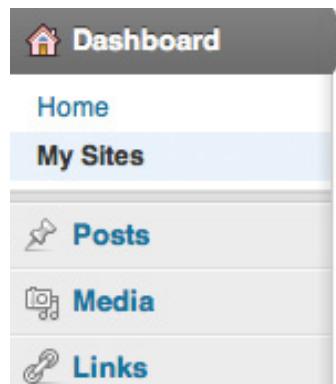
Enabling the Network

Complete the following steps to enable the features for creating a network of sites.

Caution: We recommend you back up your existing `wp-config.php` file.

Your Network Sites

The ‘My Sites’ link appears after updating your `wp-config.php` and `.htaccess` files. It’s a quick link to view and manage all of your WP network sites.



Dashboard

Home

My Sites

Posts

Media

Links

This page displays two chunks of code: one goes into your site’s `wp-config.php` file, and the other goes into your root `.htaccess` file. Remember to backup your files before making any changes. Also note that these two chunks of code will always be available under **Tools > Network** for future reference.

Welcome, Super Admin!

After updating your `wp-config.php` and `.htaccess` files, everything is setup and ready to rock. You are now “Super Admin,” and have ultimate control over every site in your network. The first thing you will notice is your shiny new “Super Admin” menu panel, which contains options for virtually everything in your network.

Adding a New Site to the Network

To add a new site to the network, click on the **Sites** link in the Super-Admin menu. At the top of the **Sites** page there is a list of all your network sites. Beneath the site list is the “Add Site” form, through which you will enter the address (sub-directory) of your new site, as well as a site title and admin email. It looks similar to this:

The Admin toolbar on the left includes links for Home, My Site, Posts, Media, Links, and Pages. The 'My Sites' link is highlighted in blue, indicating it is the active section.

The 'Network Admin' page has a sidebar with 'Dashboard', 'Sites', 'Users', and 'Visit Network'. The 'Dashboard' tab is active. A large blue box with a white 'W' logo is centered on the page.

The beauty of using WordPress MultiSite for your network is that everything runs from a **single installation** of WordPress: **one** database and **one** set of files. Just pick a Site Address when adding a site and WordPress takes care of everything else. No need to create a directory or edit any files.

Each new site added to the network includes its own unique Admin area and Default User, which is named according to the site address. In our example, we created a new site in a sub-directory called "forum," so the default username for that site is also "forum."

To manage and add users for your new site (or any site in the network), click on the "Users" link in the Super Admin menu (see screenshot to the right).

Network Admin

To manage your network, hover over the 'My Sites' link in the Admin toolbar (left).

Add New Sites

The 'Add New Site' section of the 'Sites' page enables you to easily create new sites in your network (right).



Add New Site

Site Address
redfeed.net/wordpress/
forum

Only the characters a-z and 0-9 recommended.

Site Title
DiW Forum

Admin Email
email@email.com

A new user will be created if the above email address is not in the database. The username and password will be mailed to this email address.

[Add Site](#)

Add New Users

The 'Users' page displays all network users and enables you to easily add new users.

The 'Users' page shows a list of users. A user named 'Perishable' is highlighted in red as a 'Super Admin'. Another user, 'forum', is also highlighted in red as the 'Default Admin for new site'.

Username	Name	Registered	Sites
Perishable	Jeff Starr	2012/01/13	/wordpress/
forum	DIW	2012/01/13	/wordpress/
Username	Name	Registered	Sites

Full Control Over Everything

On the surface, using WordPress to create and manage a network is relatively straightforward. But as the Super Admin there are many, *many* options and settings available to you. The default MultiSite settings should work well for most setups, and you can rest easy knowing that you can modify things as needed.

Next Steps

WordPress Tavern has a great post on “What To Do Or Consider After You Enable Multisite In WordPress”:

<http://digwp.com/u/483>

The good news is that **all of your network settings** are available through the Super Admin menu panel located at the top of the left sidebar (see screenshot). Here is an overview of what you will find under each options page:

- **Admin** - Provides a quick summary of sites and users, and enables you to search either users or sites. Also provides shortcuts to create new sites and users.
- **Sites** - Displays all sites in your network along with basic details such as default user, site paths, site IDs, and other tidbits. Also provides shortcuts for editing site details and visiting each site's Admin area. Below the site listing is a form for adding new sites to the network.
- **Users** - Displays all users in your network along with basic details such as name, email, registered sites, IDs and more. Also includes shortcuts for editing and deleting different users. Below the user listing is a form for adding new sites to the network.
- **Themes** - Provides theme options for individual sites and all sites in the network.
- **Options** - Provides general settings/options for your network. Similar to the “General Settings” page for individual sites, the “Network Options” page is where you specify things like dashboard and registration settings.

Path	Last Updated
/wordpress/	
Shortcut to site admin area	

Edit | Dashboard | Deactivate | Archive | Spam | Delete | Visit

Admin Areas for Individual Sites

Every site in your network includes its own fully functional Admin Area. When logged in as Super Admin, you can jump back-and-forth from one Admin area to the next without ever having to log out and log back in. Quick access to each of your site's Admin areas is available in the Sites page in the site listings.

3.2.5 Using Custom Taxonomies

Technically, custom taxonomies were available in WordPress 2.8, but they lacked an actual User-Interface (UI) and were not hierarchically structured. In WordPress 3.0, users now have a fully functional UI – for both posts and pages – enabling them to take advantage of hierarchical custom taxonomies. But before we can use the new taxonomy UI to manage our terms, we need to actually create our desired taxonomies via the functions.php file. Here is a basic example to get you started:

```
// create custom taxonomy
function digwp_custom_taxonomies() {
    register_taxonomy(
        'wordpress_books', // internal name = machine-readable taxonomy name
        'post',           // object type = post, page, link, or custom post-type
        array(
            'hierarchical' => true, // true for hierarchical like cats, false for flat like tags
            'label' => 'WP Books', // the human-readable taxonomy name
            'query_var' => true, // enable taxonomy-specific querying
            'rewrite' => true // pretty permalinks for your taxonomy?
        )
    );
}
add_action('init', 'digwp_custom_taxonomies', 0);
```

With this code, we're creating a custom taxonomy called "WP Books" that will enable us to further organize our collection of WordPress Books into whatever taxonomy we desire. Perhaps the *best* way to understand how this works is to add the code to your theme and then check out the new "WP Books" menu-item displayed in the Posts menu (see screenshot).

Custom Post Type UI

Essential plugin for working with custom post types and custom taxonomies, provides the much-needed Admin UI:

<http://digwp.com/u/585>

New Menu Item

For each custom taxonomy created via your theme's functions.php file, a menu item will appear under the Posts' menu in the Admin. Shown here is the menu item for our "WP Books" taxonomy.



The integration of custom taxonomies continues the transformation of WordPress from a simple blogging platform into a more robust and fully-featured CMS. Here are several excellent articles explaining all the juicy details about custom taxonomies, so be sure to check 'em out for more information:

- Custom taxonomies in WordPress 2.8 - <http://digwp.com/u/455>
- Introducing WordPress 3 Custom Taxonomies - <http://digwp.com/u/456>
- What are "custom taxonomies"? - <http://digwp.com/u/457>

3.2.6 Creating and Using Custom Menus

One of the most useful new features of WordPress 3.0 is the new menu-management system, which is developed by WooThemes to make it *super-easy* to create and manage multiple menus. Before custom menus, WordPress designers had to sort of "pick and choose" among various template tags and try to hack their way to a decent set of menus. But no longer!

To be fair, WordPress *does* have some powerful template tags for creating menus, but with so many different types of content, there is no "one-size-fits-all" template tag to suit each and every design. And as for enabling mere *users* to create their own custom menus – of any type – forget about it. It's just *too* painful to do using only template tags and functions.php trickery.

Thankfully, all this changes with WordPress 3.0's new menu management system. Now any admin-level user can easily and quickly fashion any type of custom menu: category menus with specific exclusions/inclusions, menus for external resources, specific posts, pages, and just about anything else you can think of.

Even better, version 3.0 enables users to create as many custom menus as needed. There is even a **default widget** that works *automagically* with any widgetized area. The power and flexibility that this new menu system brings to WordPress is extraordinary. Think about it: any combination of links may now be displayed anywhere in your theme with just a few simple mouse clicks. Awesome.

Custom Menu Widget

What if custom menus are not enabled in your theme? If you can add widgets, WordPress provides a "Custom Menu" widgets that can be used to display your menus. For further info on this (and much more), check out this awesome post:

<http://digwp.com/u/477>

To create and use your own custom menus, first register them by placing the following code in your theme's functions.php file. Let's say we want *three* menus:

```
// register three menus
register_nav_menus(
    array(
        'primary-menu'  => __('Primary Menu'),
        'secondary-menu' => __('Secondary Menu'),
        'tertiary-menu'  => __('Tertiary Menu'),
    )
);
```

This will register and enable three custom menus that can be displayed anywhere in your theme. Just place the following template tag in the desired location:

```
<?php // display nav menu
wp_nav_menu(
    array(
        'theme_location' => 'primary-menu',
    )
); ?>
```

In our example, the other two menus would have similar tags that also could be placed anywhere within your theme. Just replace "primary-menu" with "secondary-menu" and "tertiary-menu" for each tag. Then, with the required code in place, log into your Admin area and create a custom menu under **Appearance > Menus**.

Just specify the name of your custom menu where it says "My Custom Menu" and you're ready to create your own custom menus. Here's how to do it..



Menus Menu

After adding the required code to enable custom menus for your theme, visit the "Menus" link to create some custom menus!

Parameters Aplenty

The `wp_nav_menu` comes equipped with 15 parameters for customizing things like markup and attributes. Check out the [WordPress Codex](#) for a complete list:

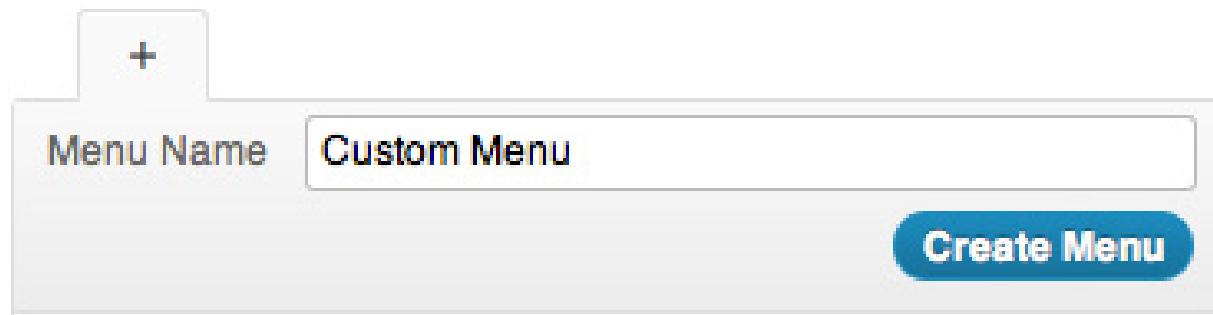
<http://digwp.com/u/458>

Create a Custom Menu

To create a custom menu, click on the “Menus” link in the “Appearance” menu panel in the Admin sidebar. Begin by picking a name for your custom menu:

Create a new menu..

Click on the tab with the plus-sign (+) to create a new custom menu.



After creating a menu, visit the inner-left sidebar and choose a Theme Location from the dropdown. Beneath that, you can add category, page, and even custom links to the menu.

Using Menus in WP 3.0

We don't always have room in the book to flesh out every topic completely. Fortunately, DigWP.com enables us to dig in to much more, such as this concise tutorial on “Using Menus in WordPress 3.0”:

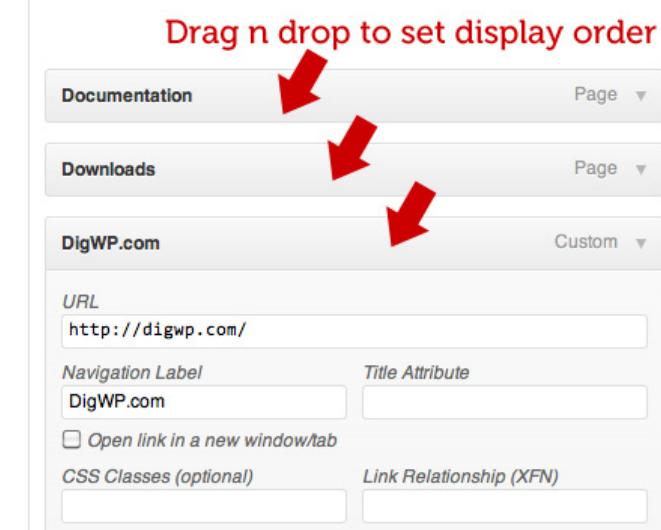
<http://digwp.com/u/502>

As you add items to your menu, they will appear as slide-open boxes in the right-hand column. There you can edit the details of any link and also drag-&-drop the link boxes to set their order of appearance in the menu.

There is much more that can be done with this template tag, so be sure to check out the WordPress Codex for more juicy details:
<http://digwp.com/u/458>

3.2.7 Custom Post Types

Our favorite new feature of WordPress 3.0 has got to be the ability to create custom-post templates. Up until now, setting up custom templates for different types of content required a bit of custom-field trickery and/or plugin shenanigans



to get the job done. But no longer! Now creating custom templates for different types of content is as easy as a few clicks in the WordPress Admin.

By default, WordPress 3.0 supports numerous post-types, including posts, pages, revisions, attachments, and nav-menus. But we aren't limited to these – WordPress' new custom-post functionality enables us to create *any* type of content imaginable. Everything you need is now well-integrated into the WordPress Admin – and it's all fully customizable according to your needs. Let's look at a basic example...

Basic Example of Custom Post Types

Let's say that you have a blog that features multimedia content. You want to keep the multimedia posts separate from the regular blog posts, such that they are displayed in separate loops and separate feeds (by default, custom types are *not* included with regular posts or the regular post feed).

To setup a custom post type, you need to create it via your theme's functions file. Here is a basic example whereby we create a custom post type for multimedia content using the `register_post_type` function:

```
// multimedia custom content type
function digwp_post_type_multimedia() {
    register_post_type('Multimedia', array('label' => __('Multimedia'),
    'public' => true, 'show_ui' => true));
    register_taxonomy_for_object_type('post_tag', 'Multimedia');
}
add_action('init', 'digwp_post_type_multimedia');
```

This code sets up a basic custom post type called "Multimedia", as seen in this screenshot. There are *tons* of additional parameters available for setting up and customizing your own content types.



Custom Post Menu

Similar to the Posts and Pages menu panels, a new menu panel will be displayed for each custom content type specified in your theme's functions.php file.

This is a Multimedia Post

Permalink: <http://redfeed.net/wordpress/blog/multimedia/post/> [Edit](#)

Upload/Insert 

Visual

HTML



This is a Multimedia post..

Custom ...What?

Around the Web, you'll see this new functionality referred to as "custom post types" in some places and "custom content types" in others. So which is it? I think the consensus is that "custom content types" makes more sense, but the WordPress Codex is calling them "custom post types," as seen here:

<http://digwp.com/u/480>

To go beyond the basics, head on over to the WordPress Codex – <http://digwp.com/u/475> – for the official scoop, and then don't miss this comprehensive article on custom-post types: <http://digwp.com/u/476>.

Displaying Custom Post Types

Once you have posted some custom posts, you can display them on your blog using the `WP_Query` class. In your theme file, add the following code snippet:

```
<?php global $wp_query;  
$wp_query = new WP_Query("post_type=multimedia&post_status=publish");  
while ($wp_query->have_posts()) : $wp_query->the_post(); ?>  
<h1><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h1>  
<?php the_content(); ?>  
<?php endwhile; ?>
```

This loop will display the title and content of the most recent custom posts (the multimedia posts in our example). See Chapter 4 in the main book to learn more about using and customizing the loop. Note that the `$wp_query` object (used in this loop) accepts the same parameters as `query_posts` (see Chapter 4 in the main book). See the Codex for details on the properties available to the `WP_Query` class, and its parameters: <http://digwp.com/u/478> and <http://digwp.com/u/479>, respectively.

3.2.8 Shortlinks

Social-media is bigger than ever. Popular services like Twitter restrict the number of characters allowed in a message, and you don't want to waste any of them on full

URLs. There are many great URL-shortening services such as *bit.ly* and *tinyurl.com*, and now WordPress makes it fast and easy to create and use your own. There are three main points to know about WordPress shortlinks:

Shortlink Format and Default Use

WordPress shortlinks are created for *posts*. The post-ID is used in the URL:

```
http://digwp.com/?p=123
```

By default, this information is included in each post's `<head>` section like so:

```
<link rel='shortlink' href='http://digwp.com/?p=123' />
```

To prevent the shortlink from appearing in the `<head>`, you must disable it:

```
remove_action('wp_head', 'wp_shortlink_wp_head', 10, 0);
```

Just add that code snippet to your active theme's `functions.php` file to disable it. Note that this technique merely disables the `<link>` element from appearing in the `<head>` section; the shortlink functionality itself will remain, enabling you to use (or not to use) the shortlinks elsewhere in your theme.

Permalinks Required!

To use `the_shortlink()` template tag in your theme, you must have permalinks enabled. See Chapter 2.3.1 in the main book to learn all about permalinks, and then visit Chapter 8.3.1 for some sweet SEO/optimization tips.

Using Shortlinks in Themes

Shortlinks may be used anywhere within the loop. Here is the template tag to use:

```
<?php the_shortlink(); ?>
```

By default, this will create a hyperlink for each post. The default code output will look like this:

```
<a href="http://digwp.com/?p=123" title="Title">The is the short link.</a>
```

We can tweak several aspects of the default markup using any the following parameters:

\$text – Link text to display. Default to: “This is the short link.”

\$title – Title attribute for the anchor tag, `<a>`. Defaults to the post title.

\$before – Text or HTML prefix added to the link. No default value.

\$after – Text or HTML suffix added to the link. No default value.

So if we want to display the shortlink in a paragraph, exclude the title attribute, and simplify the anchor text, we would include the following code in the loop:

```
<?php the_shortlink('shortlink', null, '<p>', '</p>'); ?>
```

The cool thing here is that the extra markup is only displayed if the shortlink is available for the post, leaving no empty HTML tags to soil your design.

There is also a `get_shortlink()` template tag that will merely return the shortlink without echoing it to the browser. For more information on the `the_shortlink()`, check out the WordPress Codex: <http://digwp.com/u/481>.



Grabbing the Shortlink from your post

Need to grab a quick copy of a post’s shortlink? Just pop into the **Edit Post** screen and click the “Get Shortlink” button. See screenshot at left.

This makes it super-easy to grab the shortlink for your latest post and share it immediately on Twitter, Facebook, or whatever social-media service you prefer. Just grab and go – nothing could be easier!

But wait, there's more..

Now that we've learned about the major new functionality available in WordPress 3.0, let's wrap things up by checking out some of the other improvements and features that help make version 3.0 the *best update* ever...

3.2.9 Other Awesome 3.0 Features

As if all that weren't enough, WordPress 3.0 also includes the following great features to make it better than ever before!

- **Specific Author Templates** – Now in addition to naming category-specific templates like "category-pancakes.php" and "category-20.php", we can also name author-specific templates like "author-fonzi.php" and "author-2.php", which would serve as templates for the author named "Fonzi" (case-insensitive) or the author-ID of "2" (respectively). This makes it super-easy to create author-specific page templates.
- **Bulk Updates for Themes and Plugins** – This means it will be even *easier* for you to manage your WordPress site. Bulk updating of themes and plugins is going to be a huge time saver. Just go to either Themes or Plugins in the Admin and follow the "Update" prompts. To *bulk update*, just check the box next to each item that you want to update and then click the submit button.
- **New "Super-Admin" Role** – The new "Super Admin" role has control over both individual and MultiSite content. This is setup during configuration of your own MultiSite network in the Admin area (under **Tools > Network**). See Chapter 12.2.5 for more information on MultiSite.
- **Easy Comment-Form Template Tag** – Before version 3.0, displaying your theme's comment form involved adding a crazy mess of PHP and markup to the comments.php file. Just scroll down to the bottom of just about any pre-3.0 theme to see how much code is used to create the form. In WordPress 3.0, you now have the option to include the entire comment form by simply adding the new `comment_form()` template tag in your comments.php file, like so:

```
<?php comment_form(); ?>
```

There are of course many ways to customize the default output of this tag, including parameters for just about every aspect of the comment form. For more information, check out Otto on WordPress: <http://digwp.com/u/470>

Just the Beginning...

Over the years, we've seen many WordPress updates, but of them all, version 3.0 seems to be the most **valuable and practical** in terms of advancing the **scope and usability** of WordPress.

The changes brought forth in this new version – from custom post types and taxonomies to MultiSite functionality and smarter menu management, WordPress 3.0 takes some huge leaps further ahead of the competition, becoming a much more powerful blogging engine and a robust and flexible CMS.

For complete details on changes made for the WP-3.0 update (there are many!), visit the WordPress Codex: <http://digwp.com/u/471>.



3.3 Welcome to WordPress 3.1

The long-awaited 3.1 update arrived February 23rd, 2011. Named “Reinhardt” in honor of jazz-guitarist Django Reinhardt, this latest WordPress update brings some great new features for the Admin Area and a wide variety of fixes and improvements under the hood.

As with any release, much of the WordPress 3.1 update is dedicated to cleanup, streamlining, security, and bug fixes. New features for users include **redesigned linking workflow, streamlined writing interface, and a handy Admin Toolbar**. All of this makes for a more enjoyable and efficient content publishing experience.

WordPress 3.1 brings tons of awesome to developers and theme designers as well, with new **post formats, advanced queries, a new Network Admin, revamped import/export system**, and much more. Let’s check out all the new goodness and see how the *Reinhardt* update makes WordPress better than ever.

Note: If you haven’t yet updated, you should do so automatically via the WordPress Admin (or manually if that’s your thing).

3.3.1 Custom Post Formats

Custom Post Formats enable you to publish different types of posts, like Asides, Links, and Galleries. While editing or writing a post, just select the Post Format from the menu, and the post will be displayed with a look that fits the content. For example, many blogs these days have an “Asides” category for short posts, links, and other random bits. Before WP 3.1, uniquely formatting Asides and other types of posts required custom fields, category-specific templates, or some CSS. But no more. WordPress 3.1 provides yet *another* way of making it happen.

Using Post Formats requires the addition of a small snippet to your theme file. After that, you will see a radio-button list in the Add New Post screen where you can choose the format for each post (see left screenshot). After publishing a specific type of post, it may be formatted and styled to suit your design. This is

WP 3.1: "lots of fun"

– *WordPress.org News*

<http://digwp.com/u/533>

Before Updating..

Just a reminder to always backup your current database and files before attempting any updates. Also, you may want to test your plugins for compatibility with WP 3.1 before upgrading. This plugin makes it easy to check ‘em:

<http://digwp.com/u/534>

Semantic Formatting

In short, with a theme that supports Post Formats, a blogger can change how each post looks by choosing a Post Format from a radio-button list. – WordPress Codex:

<http://digwp.com/u/528>

Formats for the Future

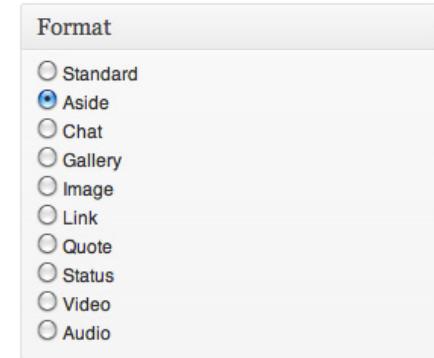
As a new feature, it will take time for developers to integrate this functionality into themes.

As that happens, formatting Asides, Galleries, and more will be a one-click process.

Until then, manually setting up Post Formats requires a bit of fiddling.

done at the theme level, where template tags and CSS may be used to make Asides look like Asides, Galleries to look like Galleries, and so on. By default, WordPress 3.1 includes 10 different Post Formats, each with its own purpose:

- **Standard** – A regular blog post (no format)
- **Aside** – Small blurbs, random thoughts, etc.
- **Chat** – Dialogue from instant messaging and chat
- **Gallery** – Groups of images or other media files
- **Image** – Any image
- **Link** – Cool links to awesome resources
- **Quote** – Any quote
- **Status** – Updates to 3rd-party services, like Twitter
- **Video** – Any video, screencast, slideshow, etc.
- **Audio** – Any audio clip, song, or podcast



For enabled themes, each post can be whatever you want. Just make your selection using the Format options, which should appear directly below the Publish/Update Panel in the “Add New Post” screen.

The idea behind Post Formats is to make the look and feel of each post match its content. Of course, it's unlikely that any site will use *all* of these formats, it's nice to have them available for custom styling if needed. As theme developers begin to implement Post Formats into their themes, users will be able to publish ready-formatted posts with the click of a button. To enable Post Formats in your theme, add this to your functions.php file:

```
// enable custom post formats
add_theme_support('post-formats',array('aside','chat','gallery','image',
'link','quote','status','video','audio'));
```

That includes options for all supported post formats. After adding this code, you should see the radio-button list of Post Formats in the Post Editor. To apply

a specific Post Format, just select from the list and click **Update**, **Save Draft**, or **Publish** (depending on the status of your post) to save your preference. From there, you have two basic ways of customizing your non-standard posts.

Method 1: Customizing Post Formats with Template Tags

The *easiest* way to customize Post Formats is to change the template tags and markup using some conditional code. Within the WordPress loop, we can customize any type of post format with something like this:

```
if (has_post_format('aside')) {  
    echo 'this post has an ASIDE post format;  
}  
  
if (has_post_format('chat')) {  
    echo 'this post has a CHAT post format;  
}  
  
if (has_post_format('gallery')) {  
    echo 'this post has a GALLERY post format;  
}
```

...and so on, with a similar conditional statement for each of your Post Formats. In this example, we're just echoing the post-format type. In your theme, you would maybe add a thumbnail to your Status posts, remove the title tag from your Aside posts, and display Gallery posts using a fancy image slider. After modifying your theme file(s), you will probably want to further customize with CSS.

Method 2: Customizing Post Formats with CSS

To style your posts with CSS, make sure the theme includes the `post_class()` tag in the outer `<div>` for each post. The `post_class()` function will output the format name as a class attribute for each post. This provides a nice hook upon which to hang your custom CSS. Here are a few examples to help illustrate:

Post Format Tags

Here are the new 3.1 template tags for working with Post Formats:

```
set_post_format()  
get_post_format()  
get_post_format_link()  
get_post_format_string()  
has_post_format()
```

For details, visit the Codex:

<http://digwp.com/u/540>

Not Displaying Stuff

*How do you *not* display the title tag for aside posts? Easy, just include an exclamation point in the if() condition, like so:*

```
if(!has_post_  
format('aside')) {  
    echo the_title();  
}
```

post_class()

When included in the loop, `post_class` will output a slew of class attributes such as `single`, `sticky`, and `post`. These styles are extremely useful for conditionally styling your posts. For more information, visit the [WordPress Codex](#):

<http://digwp.com/u/535>

```
.format-aside h3 {  
    display: none; /* hide the title on aside posts */  
}  
.format-image .meta {  
    display: none; /* hide meta info on image posts */  
}  
.format-quote blockquote {  
    color: #777; /* style quotes to look like quotes */  
    font: bold 48px/1.2 Georgia, serif;  
    margin-left: 20px;  
}
```

Post Format Classes

Here are the available CSS classes for Post Formats:

```
.format-aside  
.format-gallery  
.format-link  
.format-image  
.format-quote  
.format-status  
.format-video  
.format-audio  
.format-chat
```

3.3.2 Advanced Custom-Field Queries

WordPress 3.1 includes more powerful ways to query specific types of posts based on custom-field (aka, `postmeta`) data. In previous versions of WordPress, custom-field queries use two parameters: `meta_key` and `meta_value`. These parameters allow us to display, say, all posts with a `meta_key` value of “special”, or maybe all posts that have `meta_value` defined. With WordPress 3.1, we can drill down even further, to display all “special” posts that also include a custom-field value of “birthday”. This new functionality provides better granular control, enabling us to display virtually any set of posts.

To understand how advanced custom-field queries work, let’s consider some examples using our hypothetical “special birthdays” scenario. *Prior to WordPress 3.1, this is how we display all “special” posts (see caption for explanation):*

Custom Fields

More info on how to work with custom fields at [Perishable Press](#):

<http://digwp.com/u/538>

<http://digwp.com/u/539>

```
<?php // custom query for WP < 3.1  
$args = array(  
    'order' => 'ASC',  
    'orderby' => 'date',  
    'post_type' => 'post',  
    'post_status' => 'publish',  
    'posts_per_page' => 10,
```

The Old Way

The first five parameters are standard query variables that still work fine in WP 3.1. Note that we don’t need to declare these parameters for this custom-field query to work.

```

// old custom field stuff
'meta_key' => 'special',
'meta_value' => 'yes',
'meta_compare' => '='
);

$custom_query = new WP_Query($args);
while($custom_query->have_posts()) : $custom_query->the_post(); ?>
    <!-- template tags & markup go here -->
<?php endwhile; ?>

```

Notice the “old custom field stuff”: `meta_key`, `meta_value`, and `meta_compare`. These three parameters are saying “get only posts that include a `meta_key` of ‘special’ and a `meta_key` of ‘yes’”. This sort of query enables us to display a specific set of results, but there is no way to get posts that match more than one condition.

In WordPress 3.1, these old meta parameters are deprecated and replaced with the **new hotness**, `meta_query`. The `meta_query` parameter enables comparative queries on multiple fields. Here is the “special birthday” query again using `meta_query`:

```

$args = array(
    'order' => 'ASC',
    'orderby' => 'date',
    'post_type' => 'post',
    'post_status' => 'publish',
    'posts_per_page' => 10,

    // new custom field stuff
    'meta_query' => array(
        array(
            'key' => 'special',
            'value' => 'yes',

```

The Old Way (continued)

The final three parameters make the query happen in older versions of WordPress. They are deprecated in 3.1, and are included here to help illustrate the new things you can do with 3.1.

The last four lines use the query to create a custom loop. See Chapter 4 in the main book for more information on creating and working with custom loops.

Ahead of the Game

Post Formats will help you style & format your posts according to their content, but for true Tumblr-style Link posts with external title links, check out these two tutorials at DigWP.com:

<http://digwp.com/u/529>

<http://digwp.com/u/530>

The New Way

The first five parameters are standard query variables that still work fine in WP 3.1. Note that we don’t need to declare these parameters for this custom-field query to work.

The final `meta_query` parameter makes the query happen in WordPress 3.1 and beyond. `meta_query` is an array that contains the required terms: `meta key`, `value`, `compare`, and `type`. The `compare` parameter accepts a wide range of comparative operators (e.g., `IN`, `NOT IN`, `BETWEEN`, etc.).

```

        'compare' => '=',
        'type' => 'CHAR'
    )
)
);

```

This new query will return the same results, but the new `meta_query` parameters enable much more control over exactly which posts are displayed. Using the `compare` and `type` parameters (see `popout`), we can specify additional criteria and drill down to more specific results.

Continuing with our current example, let's refine our query to display only those posts that are both "special" AND "birthday". Thanks to `meta_query` and its ability to handle multiple arrays, we can display our "special birthday" posts like this:

```

$args = array(
    'order' => 'ASC',
    'orderby' => 'date',
    'post_type' => 'post',
    'post_status' => 'publish',
    'posts_per_page' => 10,

    // new custom field stuff
    'meta_query' => array(
        array(
            'key' => 'special',
            'value' => 'yes',
            'compare' => '=',
            'type' => 'CHAR'
        ),
        array(
            'key' => 'birthday',
            'value' => 'yes',

```

The New Way (continued)

Using the `compare` parameter, we may add multiple arrays to query posts that match on multiple terms, excluded terms, and just about any other advanced query you can imagine.

Advanced Custom-Field Query

The new `meta_query` parameter accepts an array of arrays, enabling us to query posts that match on virtually any set of terms.

We've seen plenty of support requests looking for help with `meta_query` not working, and almost always the reason was people forgetting to use an array of arrays for the parameters.

```

        'compare' => '=',
        'type' => 'CHAR'
    )
)
);

```

Notice that the `meta_query` parameter accepts an **array of arrays**. This is what enables as much granular control as needed. For example, we could further refine our “special birthday” query with a third array that requires a custom-field value of “designer”, resulting in posts associated with “special”, “birthday”, and “designer”. With this new functionality, we can query virtually anything.

meta_query Syntax

A good way to understand the required syntax is to remove the parameter values and just look at the array structure itself:

```
'meta_query' => array(array(),array());
```

meta_query Parameters

For a list of all parameters and options available when working with custom loops and custom fields, visit the WordPress Codex:

<http://digwp.com/u/537>

3.3.3 Advanced Taxonomy Queries

Similar to `meta_query` is the new `tax_query` parameter, which enables developers to query multiple taxonomies. As with `meta_query`, `tax_query` accepts an **array of arrays**, with each array refining the query with another condition. WordPress 3.1 is the first version to include advanced taxonomy queries, so official documentation doesn’t yet exist for `tax_query`, but Otto on WordPress (see sidebar) provides some good examples showing how to use `tax_query` with `query_posts`.

Query posts in category “athletes” AND tagged with “basketball”

```

$custom_query = wp_parse_args($query_string);
$custom_query['tax_query'] = array(
    array(
        'taxonomy' => 'category',
        'terms' => array('athletes'),
        'field' => 'slug',
    ),array(
        'taxonomy' => 'post_tag',
        'terms' => array('basketball'),
        'field' => 'slug',
    ),
);
query_posts($custom_query);

```

tax_query Examples

Otto provides some helpful examples of advanced taxonomy queries at his website, Otto on WordPress:

<http://digwp.com/u/531>

This type of query only works in version 3.1 (and better). It uses two arrays – one for the **category** and another for the **tag** – and will return only those posts that satisfy *both* conditions. Notice that the `terms` parameter accepts an array of terms. That comes into play with our next example.

Query posts NOT IN category “athletes” OR “coaches”

```
$custom_query = wp_parse_args($query_string);
$custom_query['tax_query'] = array(
    array(
        'taxonomy' => 'category',
        'terms' => array('athletes', 'coaches'),
        'field' => 'slug',
        'operator' => 'NOT IN',
    ),
);
query_posts($custom_query);
```

The new thing here is the inclusion of the `operator` parameter. By specifying “NOT IN” for this query, we are asking for all posts that aren’t categorized as either “athletes” OR “coaches”.

Query posts either categorized as “athletes” OR tagged as “basketball”

```
$custom_query = wp_parse_args($query_string);
$custom_query['tax_query'] = array(
    'relation' => 'OR',
    array(
        'taxonomy' => 'category',
        'terms' => array('athletes'),
        'field' => 'slug',
    ),
    array(
        'taxonomy' => 'post_tag',
        'terms' => array('basketball'),
        'field' => 'slug',
    ),
);
query_posts($custom_query);
```

The only difference between this “OR” example and the first “AND” example is the inclusion of the global relation parameter. For this query to work, the relation parameter is included *within* the tax_query, but not within either of the term arrays. It is recommended to always put this parameter first for the sake of clarity.

As mentioned, this new tax_query stuff isn’t yet documented in the WordPress Codex, but you may be able to find additional information via search engine. As soon as new information becomes available, we’ll be sure to update the book. Until then, just know that you’re way ahead of the curve!

3.3.4 Streamlined User Queries

Instead of fighting your way through the maze of odd, incongruent functions like get_userdata, get_userdatabylogin, and get_currentuserinfo to display user data, you can now just use “one function to query them all.” WordPress’ new get_users function simplifies the process and uses the powerful WP_User_Query class to do it.

The new get_users function is implemented in typical template-tag fashion. Simply specify your parameters in array format and the function will return an object comprised of the matching data. The parameters are similar to those used for query_posts, and enable you to do useful stuff like this:

```
<?php // display a list of subscriber emails
$blogusers = get_users('blog_id=1&orderby=nicename&role=subscriber');
echo '<ul>';
foreach ($blogusers as $user) {
    echo '<li>' . $user->user_email . '</li>';
}
echo '</ul>'; ?>
```

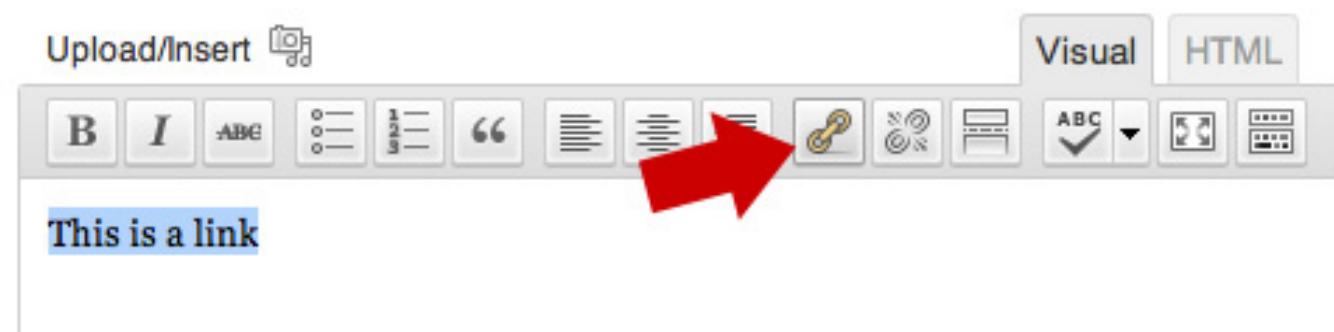
The new get_users function provides a convenient, consistent way to get at the user information. Visit the WordPress Codex for a complete list of parameters:
<http://digwp.com/u/532>

3.3.5 Awesome New Insert-Link Tool

Deep linking your internal content is now SO easy thanks to WordPress' new Insert-Link feature. When writing posts using the visual editor, just type your link text, highlight, and click the link icon in the toolbar. Here's a screenshot:

Insert-Link Button

Using the Visual Editor to compose your posts, just highlight your link text and click the Link button (looks like a little chain link). Likewise, you can use the Unlink button to change a link back to text.



Clicking the link icon brings up the **Insert/edit link** popup box. There you can specify any URL and title, or use the **insert-link** tool to find existing internal content on your blog. Using the **Search** field or selecting from the list, simply select your post, edit the title if needed, and click "Add Link" to seal the deal.

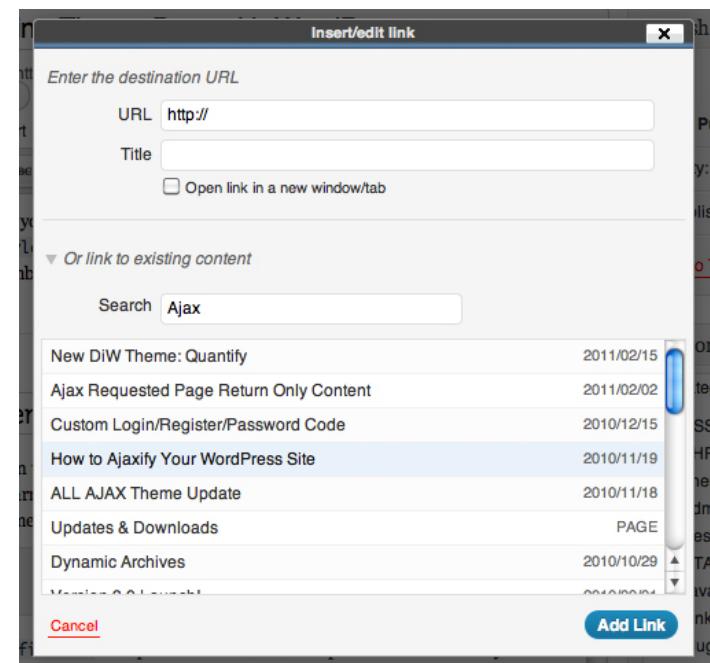
Enable the Visual Editor

Enabling the Visual Editor happens in your User Profile settings page. There you will find an option to "Disable the visual editor when writing". Make sure that's not checked – there's generally no reason to disable it because you can toggle between either Visual or Code Editor when writing or editing your posts.

The new Link tool is an excellent way to find related content to share with your visitors. As mentioned, this new tool is only available when using the visual editor, but it does a fine job of inserting links quickly and easily with the magic of Ajax & just a few clicks.

No more excuses for not deep-linking your posts!

The awesome new Link Tool, replete with both power and simplicity. A job well done.



3.3.6 Admin Area for MultiSite Networks

In WordPress 3.1, the Admin pages for MultiSite Networks have been relocated from the regular site Admin to their own exclusive area. Users with “super-admin” privileges can access the network-admin area by hovering over the **My Sites** link in the toolbar and clicking on “Network Admin.”

Once within the **Network Admin** area, you can return to the main site dashboard by hovering over **My Sites** link and clicking on your site name. In WordPress 3.1, all of the network-admin settings and options are available within the new **Network Admin** area. See Chapter 12.2.5 for more information on setting up WordPress MultiSite.

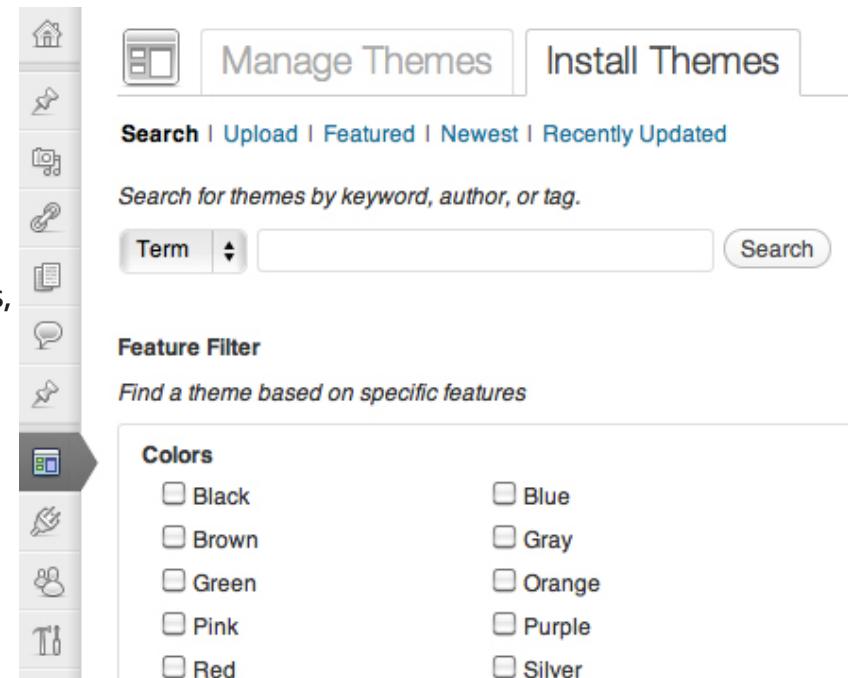


3.3.7 Theme Feature Filter

WordPress.com users have enjoyed the theme “feature-filter” for awhile, and now self-hosted WordPress users enjoy the same fancy functionality. Just visit the **Themes** settings page and click the big **Install Themes** tab at the top, as seen in the screenshot at right. There you may specify the desired attributes for your next theme. Choose colors, columns, widths, features, and more. After checking your selection, click the “Find Themes” button to view the results.

It Takes a Village...

The new Theme Filter uses tags to find matching themes. If you’re a theme developer, remember to include plenty of accurate, descriptive tags to describe your theme.



3.3.8 The Admin Toolbar

For the first time ever, the Admin area of WordPress peaks out into the front-end. With WordPress 3.1, logged-in users will notice a grey bar fixed to the top of the browser window when they visit their site. Perhaps a bit shocking for long-time WordPress users, but never fear, regular site visitors will never see the Admin Bar.

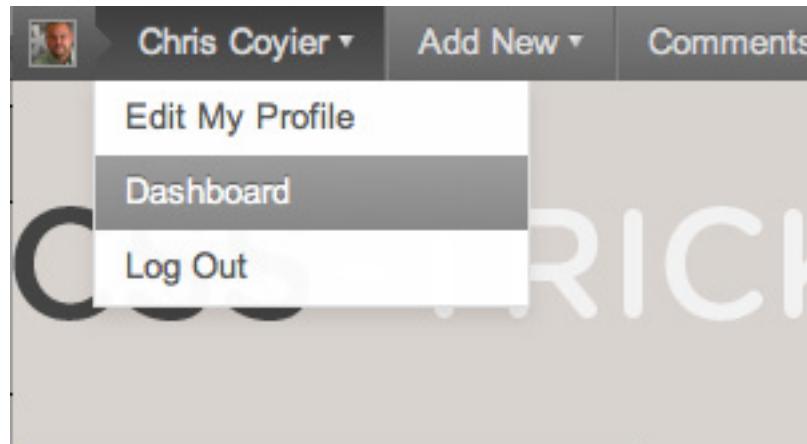
Update..

As of WP version 3.3, the much-debated Admin Bar has been replaced with the much-improved Toolbar. See Chapter 3.5.2 to discover the joy.

The new toolbar includes quick links for jumping directly to key pages in the Admin area. For instance, editing the very page you're looking at, editing comments for the page you're looking at, drafting a new post or page, and so on.

Described by Matt Mullenweg as the "first step toward a front-end editor," the Admin Toolbar helps registered users quickly access commonly used Admin pages. Used at WordPress.com and with BuddyPress, the Admin Toolbar stretches across the top of your site's pages and is designed to connect the front-end of your site with the back-end. Here's a bullet list of the highlights:

- **Regular (non-MultiSite)** – Admin Bar is displayed on the front-end only
- **MultiSite WordPress** – Admin Bar displayed on both front-end and back-end
- **Multisite WordPress** – Admin Bar features a "My Sites" dropdown menu, enabling quick access to network sites
- **Easy configuration** – you can set your Admin-Bar preferences in your User Profile settings page (see screenshot)
- **Built to customize** – the Admin Bar has plenty of hooks and filters for developers to add new features.



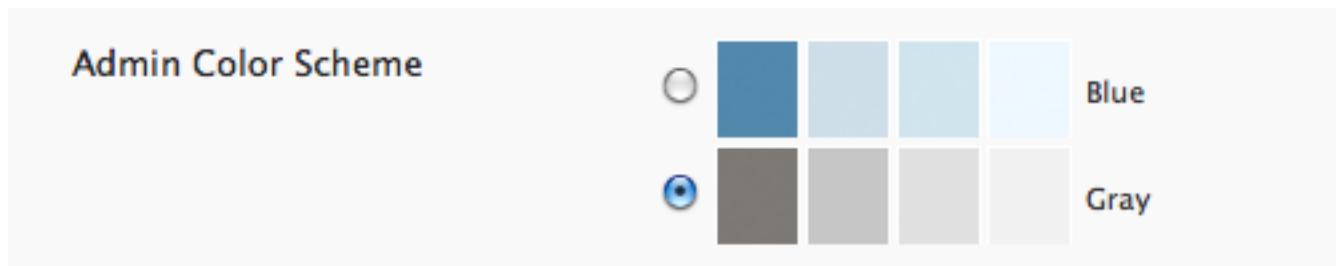
Behold the Bar

The Admin Bar is in effect at CSS-Tricks!

3.3.9 Other Awesome Features & Improvements

As if all that weren't enough to make this one of the best updates ever, WordPress 3.1 has even tons more good stuff:

- Permalinks for hierarchical taxonomies now include parent slugs in the URL, for example: <http://example.com/parent-slug/child-slug/>.
- Resetting passwords is easier than ever: one click, one email, new password.
- The "Blue" Admin color-scheme has been refreshed with a new color palette:



- Import/Export functionality vastly improved with resolved issues and better support for custom taxonomies.
- Streamlined "Add New Post" screen, thanks to new default Screen Options:
- Improved template system enables filtering of the template hierarchy using the `{$type}_template_hierarchy` filter. More info: <http://digwp.com/u/552>
- Improved admin-menu handling for custom post types: <http://digwp.com/u/553>
- New custom post-type archive templates: <http://digwp.com/u/554>

And of course tons of bug fixes and core improvements were included in the 3.1 update. For a *complete* list of features and fixes, visit the WordPress Codex: <http://digwp.com/u/558>

Gone?

After updating to 3.1, you may find the Add/Edit Post screen a lot emptier than before. Don't worry, your custom-fields, post excerpts, and other meta boxes are perfectly safe. By default, WordPress 3.1 hides most of these panels in an effort to streamline the writing process. Visit the Screen Options to customize your settings.

Problems with 3.1?

You're not alone. Here are two good resources for finding help:

Bugs and Fixes:

<http://digwp.com/u/556>

Troubleshooting WP 3.1:

<http://digwp.com/u/557>



Old Requirements

Minimum requirements for WordPress 3.1:

- PHP 4.3
- MySQL 4.1.2

For more info visit:

<http://digwp.com/u/569>

Good Riddance

WP 3.2 drops support for IE6 and begins phase-out of IE7. This is a GOOD thing.

Suddenly Refreshed

WordPress 3.2 takes the admin-area interface to the next level, which is further refreshed in WP 3.3.

3.4 WordPress 3.2 Update

WordPress 3.2 was released on July 4th, 2011. Dubbed “Gershwin” after jazz-legend George Gershwin, the WordPress 3.2 update improves performance, refreshes the administrative user-interface, and introduces a new default theme. Let’s dig a little deeper and explore these awesome new features.

3.4.1 Requirement Changes

As of version 3.2, WordPress now requires the following server software:

- **PHP version 5.2.4 or greater**
- **MySQL version 5.0 or greater**

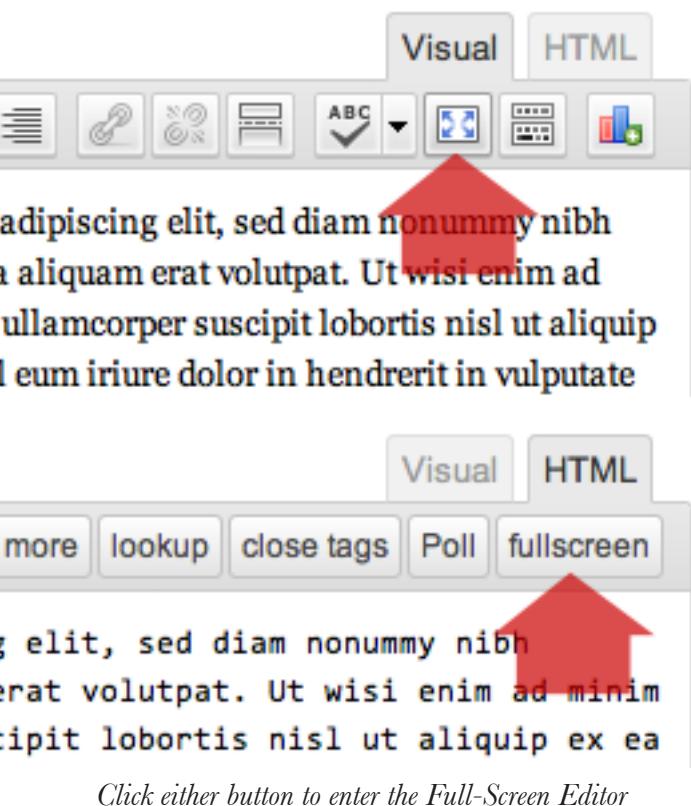
So if your host isn’t running *at least* these versions of PHP and MySQL, attempting to upgrade from a previous version of WordPress returns an error, for example:

You cannot update because WordPress 3.2 requires MySQL version 5.0 or higher. You are running version 4.1.22

These new requirements may be a pain for those hosted on older server setups, but they are for the *better*, enabling developers to further streamline WordPress and take advantage of newer software technology. Indeed, WordPress 3.2 brings faster page loads, better upgrades, and many other performance improvements.

3.4.2 Redesigned Admin Area

In addition to the awesome performance boost in WordPress 3.2, the Admin area has been completely redesigned, with tighter typography, cleaner layout, improved administrative tools, and much more.



One of our favorite new features in version 3.2 is the **Full Screen Editor**. When writing a post using the **Visual Editor**, click on the “full-screen” button to enter “zen-mode” and create your post in a clean, “distraction-free” environment.

While in zen mode, the admin-interface fades into a plain white background, leaving only your post content as the main focus.

After finishing your work, simply move your mouse toward the top of the screen, where you can exit zen mode and return to the WP Admin.

For those of us who create a lot of content, it's a welcome addition to WordPress' post-editing features.



Writing in Zen Mode

`Lorem ipsum dolor sit amet`

Hovering Menu

While in zen mode, move your mouse toward the top of the screen and the menu will magically appear.



Writing in Zen Mode

`Lorem ipsum dolor sit amet`

Full Zen Mode

Why stop at the browser window? Kick your zen-mode writing up a notch by pressing F11 to go full-screen with it.

3.4.3 New Default Theme: “Twenty Eleven”

As planned, WordPress 3.2 delivers a new default theme, dubbed “Twenty Eleven” (as in the year, 2011). It's stuffed with features and shows off the awesome new stuff you can do with WordPress: Post Formats, custom header images, custom colors, and tons more. Here are the highlights:

- **Semantic HTML5 markup**, including tags such as `<article>`, `<footer>`, `<header>`, `<aside>` and `<time>`
- **Responsive design via CSS3 @media queries** enables *Twenty Eleven* to display optimally for any screen size
- **Three different default layouts** including single-column layout or two-column layout with left or right sidebar

Honorable Mention

In WP 3.2, the HTML Editor uses a monospace font, a simple change that improves the post-editing process 100%, especially when working with code-based content. Gotta give mad props to the WP Team.

Faster Updates

After upgrading to WordPress 3.2, all future auto-upgrades will update only the files that have actually changed, rather than wasting time FTP-ing all of them. For example, updating DigWP.com to WP 3.3 was a blazing-fast experience.

WP 3.2 Update

For more complete details on the WordPress 3.2 update, check out WordPress.org:

<http://digwp.com/u/570>

<http://digwp.com/u/571>

Those are the big ones, but there's quite a bit more, including these great features:

- Rotating header images
- Support for Post Formats
- Light or dark color schemes
- Custom link colors
- Extensible, adaptable code
- 4 widgetized areas
- + tons more

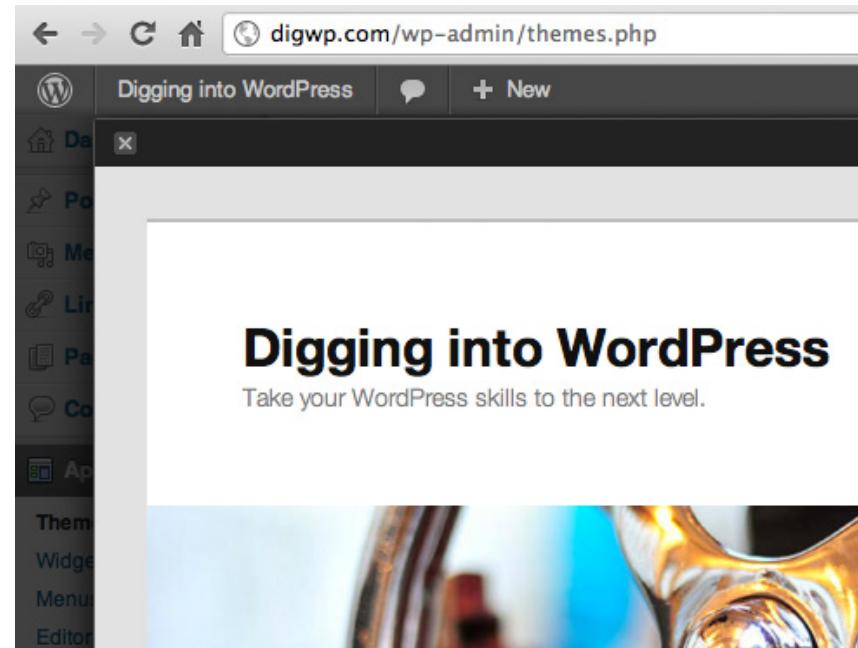
Of course, the best way to check it out is to, well, *go check it out*.

The **Twenty Eleven** theme is *enabled by default* for new WP installations. So if you're upgrading, or if the new theme isn't enabled for some other reason, you may preview it at any time from the comfort of the WordPress Admin area. Just go to **Appearance > Themes**, and then click on the "Preview" button beneath the Twenty Eleven description. As suggested by the above screenshot, it's an easy, convenient way to check out any of your installed themes.

For further information about Twenty Eleven, visit the WordPress Free Themes Directory: <http://digwp.com/u/572>

Moving on..

As awesome as it is, WordPress 3.2 is but another step toward a faster, easier, and more transparent piece of software. But like Lionel Richie we can't slow down, because the next update is already available — read on to see how WordPress gets even better with version 3.3...



3.5 WordPress 3.3 Update

Released on December 12, 2011, WordPress 3.3 brings the shine with a serious refreshing of the administrative user-interface. Dubbed “Sonny” after jazz-saxophonist Sonny Stitt, WordPress 3.3 continues the evolution of the Admin area with fast **flyout menus**, revamped **Toolbar**, and sleek new **drag-n-drop media uploader**. WordPress 3.3 delivers a cleaner, more efficient user-experience, with improvements in workflow, performance, and responsiveness. We think you’ll agree, the more you work in the WordPress Admin area, the more you’ll appreciate the 3.3 update. Let’s log in and check out the new hotness...

3.5.1 Sleek New Admin Area

WordPress 3.2 was a vast improvement over the old design, and version 3.3 makes it even better with a cleaner, tighter design, streamlined navigation, and responsive layouts that look great on a wider variety of devices and screen sizes. If you’re upgrading or installing, the first thing you’ll see is the new **Welcome Screen**, which provides helpful information about WordPress and how to do various tasks.

Basic Settings

Here are a few easy things you can do to get your feet wet. Make sure to click Save on each Settings screen.

- [Choose your privacy setting](#)
- [Select your tagline and time zone](#)
- [Turn comments on or off](#)
- [Fill in your profile](#)

Add Real Content

Check out the sample page & post editors to see how it all works, then delete the default content and write your own!

- View the [sample page](#) and [post](#)
- Delete the [sample page](#) and [post](#)
- [Create an About Me page](#)
- [Write your first post](#)

Customize Your Site

Use the current theme — Lines and Boxes — or [choose a new one](#). If you stick with Lines and Boxes, here are a few ways to make your site look unique.

- [Add some widgets](#)

After dismissing the Welcome Screen, hover your mouse over the **Dashboard Menu** to reveal the next wave of administrative efficiency: flyout navigational menus that put you *one click* away from any screen in the Admin area. Flyout menus have their own pros and cons, but once you start using them, you’ll see how much faster they make your Admin experience.

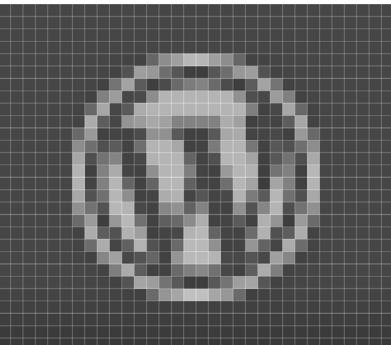
Welcome

The new **Welcome Screen** helps new users jump right in and start using WordPress immediately. It’s also the “About” page for WordPress, available anytime by hovering over the “W” icon in the Toolbar. Or visit directly @ [/wp-admin/about.php](#)



Welcome

If you need help, documentation rather dive right in, do first when the



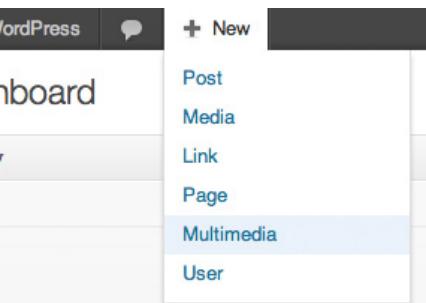
Move it..

To move the Toolbar to a less conspicuous location, check out the plugin, 'Stick Admin Bar To Bottom' by Coen Jacobs. Works great on both sides of WordPress:

<http://digwp.com/u/575>

Flyout, Dropdown

Hovering over various items in the Toolbar reveals more options via dropdown menus.



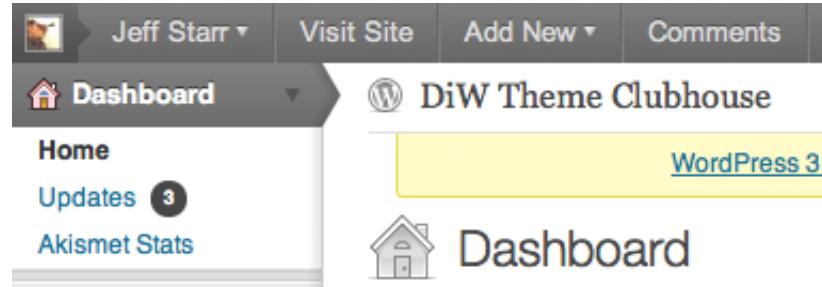
3.5.2 New Toolbar

To further improve efficiency, WP 3.3 sports a sleek new "Toolbar" that replaces the old "Admin Bar." Combining the admin header with the old Admin Bar, the new **Toolbar** takes up less vertical space and provides streamlined navigation to key pages throughout your site.

Although most will agree that the Toolbar is an improvement over the Admin Bar, not everyone loves it. For those who would like to customize or even *remove* the Toolbar, there are many plugins, scripts, and techniques for making it happen. Check out these two DiW articles for details and links to some awesome resources:

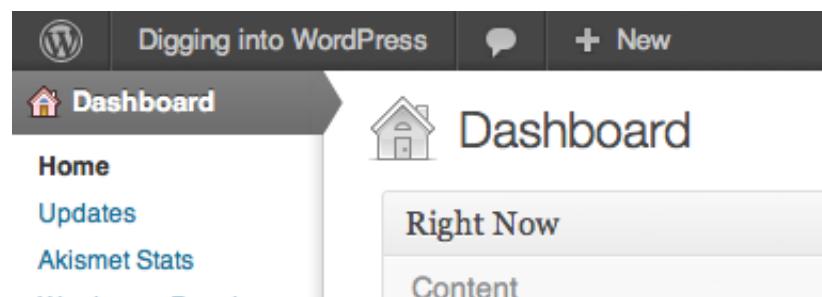
- **WordPress Admin Bar Tricks** – <http://digwp.com/u/573>
- **Goodbye Admin Bar, Hello Toolbar** – <http://digwp.com/u/566>

There you'll find an extensive list of Toolbar plugins, as well as some awesome tips, tricks, and code snippets for removing, relocating, hiding, and customizing the Toolbar. If you feel strongly one way or another about the Toolbar, check out our ongoing DiW Poll and cast your vote: <http://digwp.com/u/574>



Before & After

Screenshots comparing the old "Admin Bar" (above) vs. the fresh new "Toolbar" (below) in WordPress 3.3.



How to disable the Toolbar on the front-end

Disabling or removing the Toolbar from the *Admin area* is **not recommended** because it's integral to page design and contains links not found elsewhere. For the *front-end*, however, disabling the Toolbar is totally fine and easily done with the

following code added to your functions.php file:

```
// disable the admin bar (front end only)
add_filter('show_admin_bar', '__return_false');
```

Alternately, to disable the Toolbar for **non-admins** only, this snippet should do the trick:

```
// show admin bar only for admins
if (!current_user_can('manage_options')) {
    add_filter('show_admin_bar', '__return_false');
}
```

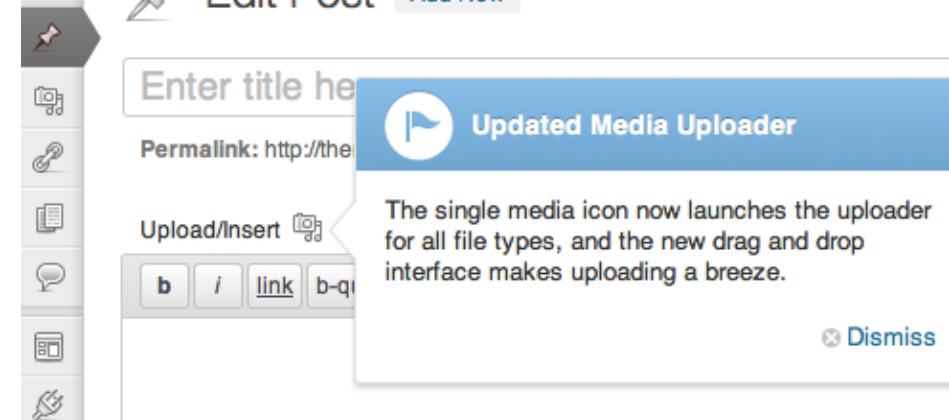
Just place in your theme's functions.php file and enjoy. Note that this also only affects the Toolbar on the **front-end** of your site.

3.5.3 Feature Pointers

As seen in the above screenshot, WordPress 3.3 also displays **Feature Pointers** that highlight new features like the drag-n-drop Media Uploader. With each subsequent release, WordPress will notify users of any relevant changes affecting the Admin area. This will help new users get acquainted and experienced users keep up with the latest. We're sure to see this new feature evolve along with WordPress.

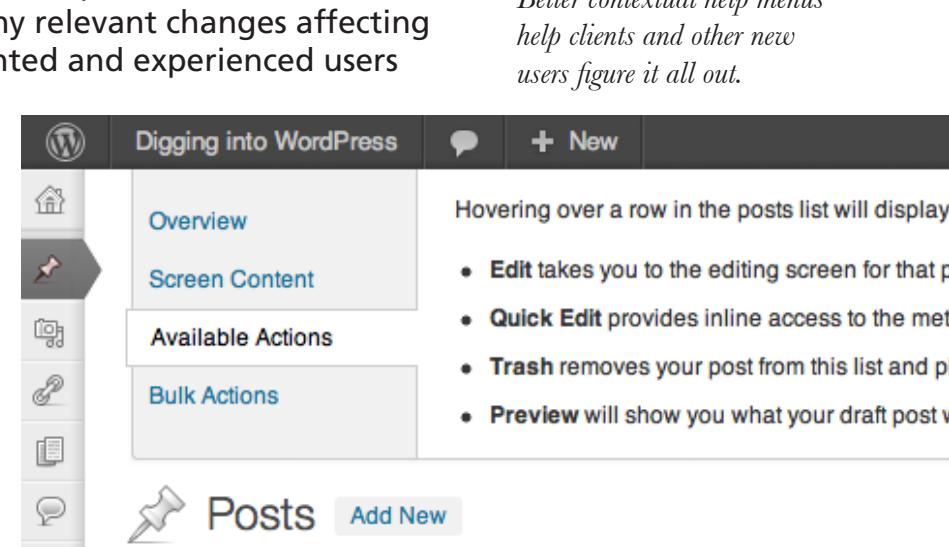
3.5.4 Better Contextual Help

Version 3.3 brings vastly improved **Help screens** throughout the Admin area. And they're *contextual*, to provide information that's relevant to the current location. For example, the screenshot to the right shows the help options for



Letting you know

Feature Pointers like this appear after new features have been added to WordPress, and they'll stay there too, until you dismiss them with a click.



Thanks, WordPress

Better contextual help menus help clients and other new users figure it all out.

The Help Button..

..is located in the upper right-hand corner of the Admin area, beneath the Toolbox.

the Posts screen. Other pages likewise display pertinent information for those who need it. If you've never browsed through the WordPress Help screens, or if it's been awhile, we encourage you to take a few moments to see it in action as you move around in the Admin area.

For developers, WordPress 3.3 makes it easier to hook into the Help screens using a new **Screen API**. Check out the following links for more information:

- **Plugin API/Admin Screen Reference** – <http://digwp.com/u/576>
- **Help and screen API changes in 3.3** – <http://digwp.com/u/577>

3.5.5 Drag-n-Drop Media Uploader

As if all the shiny new Admin chrome wasn't cool enough, WordPress 3.3 hits us with the incredibly awesome drag-n-drop **Media Uploader**. If you want to *dazzle* your clients with WordPress, the Media Uploader is sure to please.

Drag-n-Done

Drop multiple files into the Media Uploader and done. Or just one file, whatever works.

The old Browser Uploader is still there for those who use it, but the drag-n-drop uploader is much easier from start to finish. What can we say, try it once and you're hooked.



All-in-One

WP 3.3's file-type detection reduces clutter with a single 'Add Media' button.

Digging into WordPress

Upload New Media

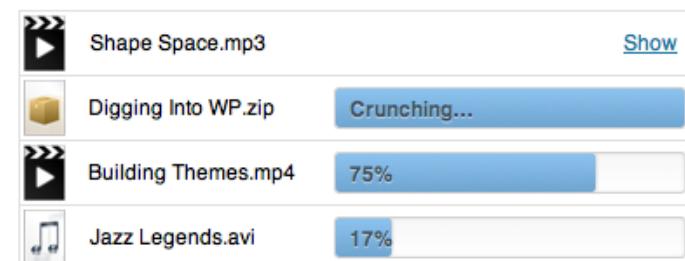
Drop files here

or

Select Files

Anything, Anywhere

From the 'Post Editor' or 'Add Media' screen, the Media Uploader handles anything you can throw at it, as seen here with some common media formats.



As shown in the screenshots, the Media Uploader can handle just about anything you can throw at it. Single JPG image? Done. Multiple MP3 files? Done. Random collection of diverse media types? No problem, just drag them from your desktop, drop them into the uploader, and watch WordPress work its magic.

Issues with 3.3?

Start here: Troubleshooting WordPress 3.3 – Master List:

<http://digwp.com/u/584>

3.5.6 Other Awesome Features in WP 3.3

From a practical, everyday-use perspective, WordPress 3.3 was largely about improving the appearance and functionality of the Admin area. Beyond this, the WordPress team seemed to focus on the **details**, resolving many bugs, tightening security, and enhancing existing features. The result is a WordPress that performs better than anything before it, especially from the administrative side. Here's a look at the other *best features* of WordPress 3.3.

Better design responsiveness

If you do administrative tasks on smaller devices, you'll appreciate the improved responsiveness of the Admin area. The two-column Edit Post screen, for example, collapses into a spacious single-column layout when viewed on the iPad.

WordPress on iPad

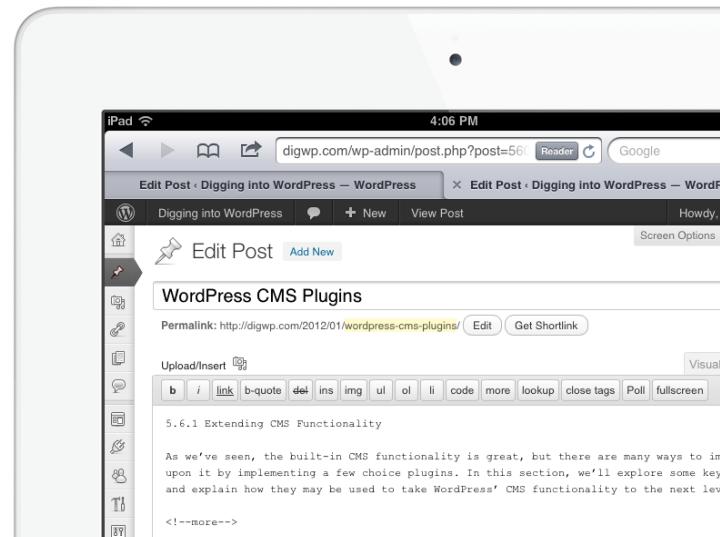
Example of responsive design on the iPad, as page layout collapses into a clean, single-column design, maximizing available space.

Improved co-editing support

No more emailing your co-authors asking them to "please log out so I can edit the post." Once you've left the Edit Post screen, it's now a matter of seconds before a different user can log in and work on the post. Huge time-saver for co-editors.

Tumblr importer

WordPress 3.3 adds Tumblr to the growing collection of Import Tools. It's a great way to supplement your site content, kick-start your own blog, or simply back up your Tumblr posts.



Patience Not Included

Tumblr servers can be slow, but WordPress does a good job of keeping track of everything and getting the job done.

Import Tumblr

[Refresh view](#)

Tumblr Blog	URL	Posts Imported	Author Selection	Action
Perishable Press	http://perishable.tumblr.com/	38	Jeff Starr	In Progress

To import your Tumblr posts, visit **Tools > Import**, and click “Tumblr.” After the “Install importer” screen pops up, click on the “Install Now” button to install the Tumblr importer plugin. After successful installation, click “Activate Plugin & Run Importer” to begin the Tumblr import process. Sounds complicated, but it’s not.

More on Postname..

Excellent write-up on 3.3’s post-name permalink fix:

<http://digwp.com/u/578>

From the **Import Tumblr** screen, enter your password, choose some basic options, and then, well... wait. Tumblr’s servers are “often overloaded,” so it can take some time for the import to complete, especially when importing many posts.



Permalink Settings

<input type="radio"/> Default	http://digwp.com/?p=123
<input type="radio"/> Day and name	http://digwp.com/2012/01/17/sample-post/
<input type="radio"/> Month and name	http://digwp.com/2012/01/sample-post/
<input type="radio"/> Numeric	http://digwp.com/archives/123
<input checked="" type="radio"/> Post name	http://digwp.com/sample-post/
<input type="radio"/> Custom Structure	<code>/%postname%/</code>

Post-Name Permalinks

In previous versions of WordPress, word on the streets was that using post-name only permalinks was a bad idea. Bad news too, as post-name permalinks happen to be optimal for SEO. People continued using them, apparently without issue...

Then in June 2011, **CSS-Tricks.com** went down. It was later determined that WordPress’ post-name permalinks were the issue. Chris explains the gory event in detail at DigWP.com:
<http://digwp.com/u/579>

Moral of the story? Don’t sweat it. As of WordPress 3.3, post-name-only is now a **recommended** permalink structure.

Smarter widgets

Another one of those “little things” that improves WordPress, widgets now remember their position in your theme. This makes it easier to restore your theme’s layout after working with alternate themes. Nothing to do, it just happens.

Even more awesomeness

Wrapping up the 3.3 update, it’s worth mentioning some additional **key changes** aimed mostly at WordPress developers.

- **New `wp_editor` API** – enables developers to run more than one instance of the visual WYSIWYG editor on any page in the WordPress Admin area.
- **jQuery upgrade** – now running jQuery version 1.7.1 and jQuery UI 1.8.16 (be sure to check any custom scripts).
- **New `is_main_query()` function** – lets you modify the main `WP_Query` object only, perfect for customizing multiple loops. Read more: <http://digwp.com/u/580>
- **Better post-format editing** – lets you bulk-edit and quick-edit based on post-format from the Posts screen (/wp-admin/edit.php).

As with any WordPress update, there are many minor improvements, bug fixes, and security enhancements that we just don’t have room for in the book. So for the truly dedicated WP enthusiast who wants all the gory details, the following resources are an absolute must:

- **WordPress 3.3 “Sonny”** – <http://digwp.com/u/581>
- **WordPress Codex: Version 3.3** – <http://digwp.com/u/582>

We hope you enjoy the 3.3 update – Happy WordPress’ing! :)

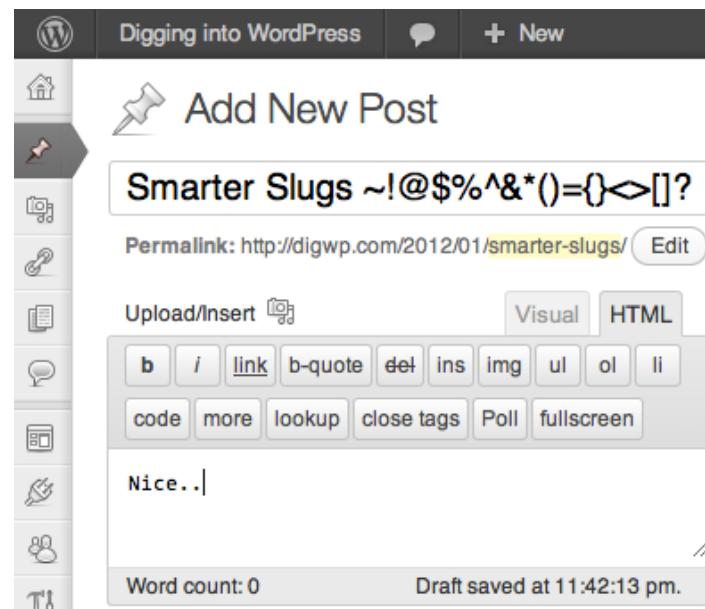
Better Performance?

Some report that the revamped Admin feels slower than in previous versions. Is it true? The dev4press.com team ran some interesting tests. Check out their WordPress 3.1/3.2/3.3 Benchmark:

<http://digwp.com/u/583>

Smarter Slugs

Automatic post-slugs get a little smarter in WP 3.3. Certain characters such as ~!@#\$%^&()={}<>[]? are no longer automatically included in auto-created post slugs, as demonstrated below.*



More Secure

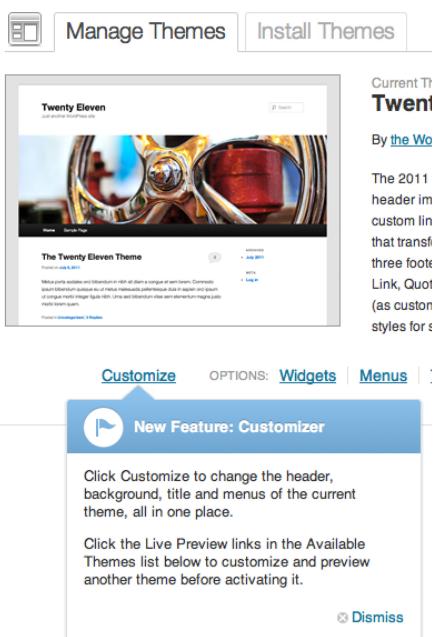
At the time of this writing, WP 3.4 has been updated twice (versions 3.4.1 and 3.4.2) with security fixes and further security hardening.

<http://digwp.com/u/604>

<http://digwp.com/u/605>

Theme Customizer

Visit Appearance > Themes

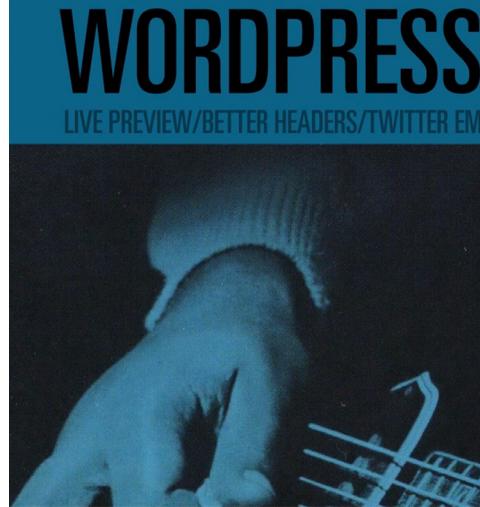


The screenshot shows the WordPress Admin interface for themes. At the top, there are buttons for 'Manage Themes' and 'Install Themes'. Below that, a list of themes is shown, with 'Twenty Eleven' highlighted. A 'Current Theme' section on the right shows a preview of the theme's design and some customization options. A prominent blue banner at the bottom left says 'New Feature: Customizer' with a 'Customize' button. Text below the banner explains that it allows users to change the header, background, title and menus of the current theme in one place. There are also links to 'Live Preview' and a 'Dismiss' button.

3.6 WordPress 3.4 Update

WordPress 3.4 is here! Named "Green" after guitarist Grant Green, version 3.4 hit the shelves on June 13th, 2012. In addition to a smorgasbord of bug-fixes and enhancements, the latest release features better theme customization, internationalization, core queries, and other good stuff.

WordPress 3.4 gives you more control over the *look and feel* of your WP-powered site, with a live Theme Customizer and custom backgrounds & headers to make it all good. And improvements in core WP-queries and the language-translation system enable a wider reach for your content, which is delivered faster and better than ever before.



WordPress 3.4 "Green"
<http://digwp.com/u/606>

3.6.1 Live Theme Customizer

The new Theme Customizer enables you to preview theme-changes in *real-time*, from within the comfort of the WP Admin. As you customize your options with the Theme Customizer, a live preview shows how the front-end design will look if you decide to save the changes. This live-preview, "WYSIWYG"-functionality makes it easy to change your theme's colors, fonts, layout, content, and much more. For supportive themes, the customizer simplifies and streamlines the customization process in a more intuitive, user-friendly way.

Using the Theme Customizer

To use the Theme Customizer, visit the **Themes** page and click either "Customize" (for the active theme), or "Live Preview" (for other available themes). If you're customizing the *active* theme, changes will not be applied until you click the "Save & Publish" button. For other themes, changes will not be applied until you click the "Save & Activate" button. In either case, you can cancel changes and return to the **Themes** page by clicking the "Cancel" button.



3.4 GREEN

BEDS/HTML CAPTIONS/UNDER THE HOOD

Adding the Theme Customizer to your theme

Implementing support for the Theme Customizer is a great way to enhance the usability and appeal of your theme. And the WordPress team says that there's "more planned for the customizer down the road," so now is a good time to get in on the action by adding support for the Theme Customizer in your own themes.

In-depth coverage of the Theme Customizer is beyond the scope of this book, but a concise tutorial on the basics will help bring you up to speed. Before we get into the tutorial, let's look at an **overview** of how to implement the Theme Customizer in your own WordPress theme:

- **Include the customizer** via the `customize_register` hook.
- Add a customization **section** via the `add_section()` function.
- Add a customization **setting** via the `add_setting()` function.
- Add a customization **control** via the using `add_control()` function.
- Include the customization variables in the `header.php` file.
- Include the required CSS in the `header.php` file.

To see an example of this process, let's enable users to customize the color and size of `<h1>` and `<h2>` tags. As we go, keep in mind that this is a basic example to get you up and running. Once you see how it works, we encourage you to explore the **Theme Customization API** and other resources mentioned in the sidebar.

Step 1: Include the customizer in `functions.php`

The first step is to enable the Theme Customizer by adding the following code to your theme's `functions.php` file:

Test Drive

The Theme Customizer provides a consistent method of customizing supportive themes. If your current theme doesn't support the Theme Customizer, WP's default Twenty Eleven theme should give you a good idea of how it works and a sense of the possibilities.

Theme Customizer

Theme Customization API

<http://digwp.com/u/607>

How to Leverage the Theme Customizer in your themes

<http://digwp.com/u/608>

Making a custom control for the Theme Customizer

<http://digwp.com/u/609>

You are previewing
Twenty Eleven
Site Title & Tagline
Navigation
Static Front Page

Step 1: Registering our function via customize_register enables the customizer and its default settings.

```
add_action('customize_register', 'digwp_customize_register');
function digwp_customize_register($wp_customize) {
    // stuff's gonna go here
}
```

This snippet sets up the customizer by hooking into `customize_register`. Once in place, it's ready to fill with the required code for your customization settings.

Step 2: Add setting and control functions for colors

Next, replace "stuff's gonna go here" from Step 1 with the following code:

```
$colors = array();
// h1 color
$colors[] = array(
    'slug'    => 'digwp_h1_color',
    'default' => '#111',
    'label'   => __('h1 color', 'digwp')
);
// h2 color
$colors[] = array(
    'slug'    => 'digwp_h2_color',
    'default' => '#777',
    'label'   => __('h2 color', 'digwp')
);

foreach($colors as $color) {
    // color settings
    $wp_customize->add_setting($color['slug'], array(
        'default'    => $color['default'],
        'type'       => 'option',
        'capability' => 'edit_theme_options'
    ));
    // color controls
    $wp_customize->add_control(new WP_Customize_Color_Control(
```

Note...

Code continues on next page.

```

$wp_customize, $color['slug'], array(
    'label'      => $color['label'],
    'section'   => 'colors', // using WP's built-in color section
    'settings'  => $color['slug']
));
}

```

This code sets up custom colors for both `<h1>` and `<h2>` tags. In the first section of code (highlighted in blue), we define each of the color arrays with a slug, default color, and label. In the next section of code (highlighted in red), we then apply the required settings and controls using a `foreach` loop. The default values here should work fine, but also may be further customized according to your needs.

In this step, we're using WP's native function, `WP_Customize_Control`, which uses its own `add_section()` configuration to include our color-settings in the "Colors" menu of the Theme Customizer. Thus, `add_section()` is not needed for this step, but will be required a bit later when we implement settings for font-sizes. In addition to colors, WordPress provides the following built-in sections:

- `background_image` – for **Background Image** settings
- `header_image` – for **Header Image** settings
- `nav` – for **Navigation** settings
- `static_front_page` – for **Static Front Page** settings
- `title_tagline` – for **Site Title & Tagline** settings

When using any of these built-in sections, declaring `add_section()` is not necessary. Instead, refer to them by name, as seen in the code above. In the next step, we'll need to use `add_section()` to create a new section for our font-size settings.

Step 3: Add section, setting, and controls for font-sizes

Just beneath the snippet from Step 2, add the following code:

The screenshot shows the WordPress Theme Customizer interface. At the top right, there are 'Close' and 'Saved' buttons. Below that, a message says 'You are previewing Twenty Eleven'. On the left, a sidebar has 'Site Title & Tagline' and 'Colors' sections. The 'Colors' section is expanded, showing 'h1 color' and 'h2 color' with their respective color swatches.

Step 2: After adding the required settings and controls to our registration function, color options appear in the Theme Customizer. Note that the color options won't affect the theme until adding the code from Step 4.

You are previewing Twenty Eleven

Headings

`<h1> font-size`
24px

`<h2> font-size`
18px

Site Title & Tagline

Colors

Step 3: After adding the required settings, controls, and section, our font-size options appear in the Theme Customizer. Note that, like before, these settings won't affect the theme until adding the code from Step 4. It all comes together in the next step!

```
// h1 font-size
$wp_customize->add_setting('digwp_h1_size', array('default' => '24px'));
$wp_customize->add_control('digwp_h1_size', array(
    'label'      => __('<h1> font-size', 'digwp'),
    'section'   => 'digwp_custom_headings',
    'settings'  => 'digwp_h1_size',
    'type'       => 'text'
));
// h2 font-size
$wp_customize->add_setting('digwp_h2_size', array('default' => '18px'));
$wp_customize->add_control('digwp_h2_size', array(
    'label'      => __('<h2> font-size', 'digwp'),
    'section'   => 'digwp_custom_headings',
    'settings'  => 'digwp_h2_size',
    'type'       => 'text'
));
$wp_customize->add_section('digwp_custom_headings', array('title' => __
('Headings', 'digwp')));
```

The first two blocks of code (highlighted in blue) define the settings and controls for the custom font-sizes. Note that WP doesn't have a built-in section for headings, so it's necessary to define it in the last line (highlighted in red), and also declare it for each control by including the line, "'section' => 'digwp_custom_headings'".

Step 4: Include the required code in header.php

At this point, our theme's functions.php file is ready to go, but it won't do anything until we've added the required code to the header.php file. In order for the changes to be applied to the theme, the customizer needs a way to update the web-pages with the user's custom settings. In our current example, we're changing CSS properties, so we want to add the following code to the header.php file, placed just before the closing <head> tag:

```

<?php // Theme Customizer
$digwp_h1_color = get_option('digwp_h1_color');
$digwp_h2_color = get_option('digwp_h2_color');
$digwp_h1_size = get_theme_mod('digwp_h1_size');
$digwp_h2_size = get_theme_mod('digwp_h2_size');

?>

<style>
h1 {
    color: <?php echo $digwp_h1_color; ?>;
    font-size: <?php echo $digwp_h1_size; ?>;
}
h2 {
    color: <?php echo $digwp_h2_color; ?>;
    font-size: <?php echo $digwp_h2_size; ?>;
}
</style>

```

With this code in place, our theme uses the customizer to change the color and size of our `<h1>` and `<h2>` heading tags. In the first code block (highlighted in blue), the custom variables are defined using either `get_option()` or `get_theme_mod()`.

The second block of code (highlighted in red) then uses the custom variables to generate the CSS required to customize our theme. At this point, everything's in place and we're ready to see the customizer in action by choosing the color and font-size of our theme's `<h1>` and `<h2>` tags (see screenshot).

Going further

Once you see how the Theme Customizer works and how it's implemented, the possibilities become endless. In addition to customizing any of the CSS, you can add text-fields, radio-buttons, custom content and more to any location in your theme. There are emerging some great articles online covering different aspects of the customizer, and the Codex contains some good information as well. To keep going, visit the links provided at the beginning of this section.

Details

Note: when getting your variables, use `get_option()` for built-in sections such as `color`, and use `get_theme_mod()` for everything else.

Selective Targeting

If we want to customize only those `h1` tags found in, say, the header, we can prefix our CSS selectors with the appropriate class name:

```
.header h1 { ... }
.header h2 { ... }
```

The screenshot shows the WordPress Theme Customizer interface. At the top, there are 'Close' and 'Saved' buttons. Below that, a message says 'You are previewing Twenty Eleven'. The main area has a dark header with 'Headings' in white. Under 'Headings', there are two sections: '<h1> font-size' with a value of '24px' and '<h2> font-size' with a value of '18px'. Below this, there are sections for 'Site Title & Tagline' and 'Colors'.

Custom Background

Background Image

Preview



Remove Image

Remove Background

This will remove the b

Once enabled, custom-background options are available in the Admin via Appearance > Background.

Theme Paths

`get_template_directory_uri()` is used to get the theme's directory path. So our default background-image is located in our theme's `/images/` folder.

Custom Backgrounds

For more information about custom backgrounds, check out section 3.2.3 and also the WordPress Codex:

<http://digwp.com/u/610>

3.6.2 Custom background and header images

As if the Theme Customizer weren't awesome enough, WordPress 3.4 also introduces new registration methods for custom backgrounds and headers, making it even easier to customize your theme. Custom backgrounds and headers have been a part of WordPress since version 3.0, but are much improved in WP 3.4, with streamlined functionality that works great with the new Theme Customizer. If your theme doesn't yet support them, check out WP's default *Twenty Eleven* theme to see how they work, and then read on to learn how to implement them.

Add theme support for custom background-images

The improved custom-background feature enables the user to specify display options for backgrounds, select images from the Media Library, and preview custom backgrounds from within the Theme Customizer. In WordPress 3.4, support for custom backgrounds is added via `add_theme_support()` in the `functions.php` file. Here is a simplified example to get you started:

```
// add theme support for custom backgrounds
$args = array(
    'default-image' => get_template_directory_uri() . '/images/bg-default.jpg',
    'default-color' => 'efefef', // no # symbol before the color!
);
add_theme_support('custom-background', $args);
```

In this snippet, we first specify an array of arguments that includes `default-image` and `default-color`. For this to work, the hash-symbol "#" prefix must be excluded and the image "bg-default.jpg" must be available on the server. Once everything is in place, users may specify custom backgrounds using the Theme Customizer (if enabled), or by visiting **Appearance > Background**.

Custom-background options via the Theme Customizer

Background Image

Background Image



Background Repeat

- No Repeat
- Tile
- Tile Horizontally
- Tile Vertically

Add theme support for custom header-images

The revamped custom headers feature enables the user to specify the width and height of custom images, select images from the Media Library, and preview custom headers from within the Theme Customizer. As with custom backgrounds, support for custom headers is added via `add_theme_support()` in the `functions.php` file. Here is the basic code required to implement custom headers:

```
// add theme support for custom headers
$args = array(
    'height' => 288,
    'width' => 1000,
    'flex-height' => true,
    'flex-width' => true,
    'default-image' => get_template_directory_uri() . '/images/header.jpg',
    'random-default' => false,
    'header-text' => true,
    'uploads' => true
);
add_theme_support('custom-header', $args);

// custom-header registration
register_default_headers(array(
    'header_image_newyork' => array(
        'url' => '%s/images/header-newyork.jpg',
        'thumbnail_url' => '%s/images/header-newyork.jpg',
        'description' => __('New York', 'digwp')
    )
));
```

In the first section of code (highlighted in blue), we pass an array of arguments via the `add_theme_support()` function. There are more arguments available (see sidebar on next page), and you may customize any of the ones used here to suit your needs. For example, you may want to change the width and height, enable flexible image-sizing, specify the default image, or even randomize the header-images.

Custom Header

Header Image

Preview



Once enabled, custom-header options are available via Appearance > Header.

Theme Paths

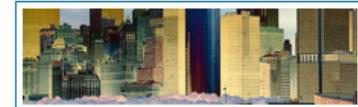
`get_template_directory_uri()` is used to get the theme's directory path. So our default header-image is located in our theme's `/images/` folder.

Header Image

Header Image

No Image

Default



Custom-header options are also available via the Theme Customizer.

Custom Headers

For more information on custom headers, check out these fine pages at the WP Codex:

<http://digwp.com/u/611>

<http://digwp.com/u/612>

<http://digwp.com/u/613>

In the second section of code (highlighted in red), we complete the implementation by registering our custom header-image. Note that the function we're using, `register_default_headers()`, accepts an **array of arrays** as the argument. This enables us to specify easily the URL, thumbnail-URL, and description for each of our custom header-images. In our example, we register a single image only, but it's simple to register as many as you'd like. Here's a simplified example showing the *pattern of arrays for multiple header-images*:

```
register_default_headers(array(  
    'header_image_01' => array(),  
    'header_image_02' => array(),  
    'header_image_03' => array(),  
) );
```

At this point, everything is in place for custom-headers to work, but there's one more step that's required for the images to be displayed on the front-end. Open your theme's `header.php` file and add the following slice of code:

```
width; ?>" height="<?php echo get_custom_header()->height; ?>" alt="" />
```

Even more..

For more details on custom backgrounds and headers, check out Chip Bennett's article:

<http://digwp.com/u/619>

And to go in-depth with flexible custom headers, check out this excellent tutorial by Amy Hendrix:

<http://digwp.com/u/618>

As you can see, this is just an `` tag that uses our theme's custom-header settings for the `src`, `width`, and `height` values. Easy peasy.

Once everything is in place, users may specify custom headers using the Theme Customizer or by visiting **Appearance > Header**.

As seen by visiting `Appearance > Header`, image-dimensions specified via `custom_header` are now suggested in bold text.

 **Custom Header**

Header Image

Preview



Select Image

You can upload a custom header image to be shown at the top of your site. On the next screen you will be able to crop the image. Images of exactly **1000 x 288 pixels** will be used as-is.

Backwards-compatibility for backgrounds & headers

When adding theme-support for custom backgrounds and headers, it's wise to make things backwards-compatible by checking if WordPress is at least version 3.4. For **custom backgrounds**, add the following code to your theme's `functions.php` file:

```
// backwards-compatible backgrounds
global $wp_version;
if (version_compare($wp_version, '3.4', '>=')) {
    add_theme_support('custom-background');
} else {
    add_custom_background($args);
}
```

For **custom headers**, we employ similar logic: use the new method for version 3.4, otherwise use the old method. Add the following to `functions.php`:

```
// backwards-compatible headers
global $wp_version;
if (version_compare($wp_version, '3.4', '>=')) {
    add_theme_support('custom-header');
} else {
    add_custom_image_header($args);
}
```

Especially for distributed themes, the two seconds it takes to make things backwards-compatible is worth the effort. Your users will thank you when things don't break if and when they forget to upgrade to the latest version of WordPress.

3.6.3 More Theme-related Goodness

As we've seen, the 3.4 update brings *much love* to theme-customization, and there's still more to mention. For example, Twenty Eleven has been updated to take advantage of the Theme Customizer. This helps new users easily customize WordPress to suit their needs right out of the box (see side note). And if the default theme doesn't get them there, improvements to the **Theme Installer** in version 3.4 enable a smoother "shopping" experience for themes available in the **WP Themes Directory**. Here are some highlights of the revamped Theme Installer:

- **Optimized theme browsing** – now possible to browse themes without paging
- **Better theme searching** – theme-installer now searches by keyword by default
- **Easier to install child-themes** – the theme-installer now supports child-themes
- **Special characters** – theme-installer now supports special characters in names
- **Better search results** – existing themes are now excluded from search results

Note..

The Theme Customizer works "right out of the box" for supportive themes. For example, default WP themes such as Twenty Thirteen support basic customization of the site title, tagline, and static front page. See your theme's `readme.txt` for information.

[Free WP Themes!](#)

[WP Themes Directory](#)
<http://digwp.com/u/614>

Simpler Login URLs

Another cool new feature in the latest version of WP: all of the following URLs are redirected to the Admin area:

`example.com/admin`
`example.com/dashboard`
`example.com/login`

Before 3.4, we did it this way:

<http://digwp.com/u/620>

Collectively, these enhancements make for a much-improved theme-searching experience. Try it for yourself next time in the WP Admin.

3.6.4 Other awesome improvements in WP 3.4

WordPress 3.4 also brings improvements in core-performance, XML-RPC functionality, and media-support. Some of these features may be noticeable immediately, while others will be appreciated when they're needed by the user. Here are some of our favorites...

Ticket #18536

An intriguing, insightful, and important ticket in the WP Trac: "Improve performance of WP_Query core".

<http://digwp.com/u/621>

Digging Deeper

XML-RPC WordPress API

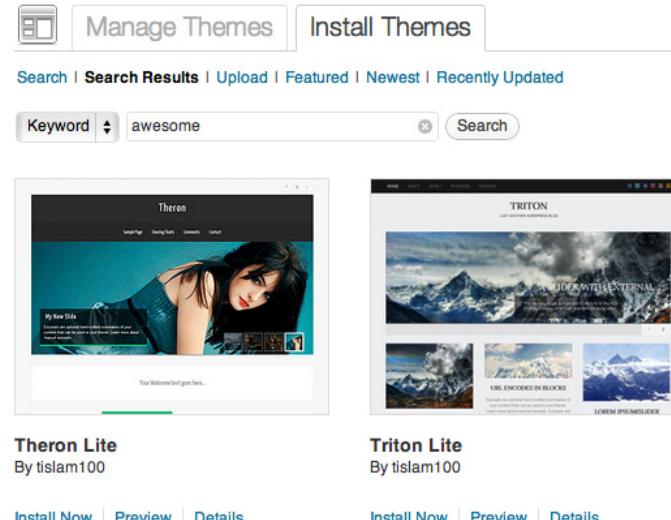
<http://digwp.com/u/617>

XML-RPC Support

<http://digwp.com/u/616>

Extended API Plugin

<http://digwp.com/u/615>



The screenshot shows the WordPress Theme Installer interface. At the top, there are buttons for 'Manage Themes' and 'Install Themes'. Below that is a search bar with the keyword 'awesome' and a 'Search' button. The main area displays two theme preview cards. The first card is for 'Theron Lite' by tislam100, featuring a woman in a blue dress. The second card is for 'Triton Lite' by tislam100, featuring a snowy mountain landscape. Each card has an 'Install Now' button, a 'Preview' button, and a 'Details' button.

The Theme Installer is better than ever in WP 3.4

Performance upgrade

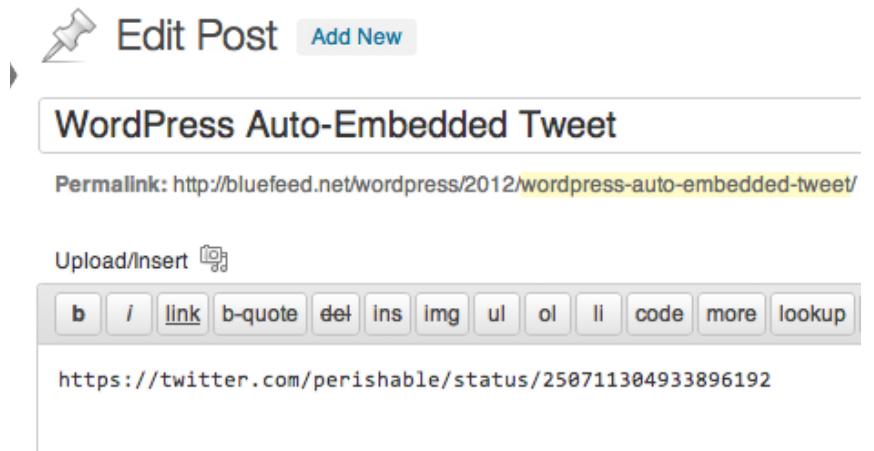
WordPress 3.4 improves the performance of its **core queries**, meaning that it's able to query the database faster and thereby decrease the load-time of your web-pages. Instead of selecting *all* posts from the database, `WP_Query` now grabs only what it needs to build post queries. Depending on the number of posts, this dramatically lightens the load on the database and results in "vastly improved performance." Smaller sites should notice somewhat snappier pages, while larger sites should notice *significantly faster* page loads.

Better XML-RPC support

Long-awaited by users requiring XML-RPC support, improvements to WP's XML-RPC functionality include many bug-fixes and support for featured thumbnails, custom post-types, taxonomies, and more. This means that it's easier than ever to use WordPress with other software such as Windows Live Writer and mobile apps. Check out the sidebar links for more information on XML-RPC.

Auto-embedded tweets

Since version 2.9 it's been possible to embed YouTube videos and other media, and now in version 3.4, tweets also are embedded easily. Just grab the URL for any Twitter tweet and place it on its own line in your post or page. WordPress then uses the oEmbed protocol to automatically convert the URL into a fully functional (and very authentic-looking) tweet. As seen in the screenshot, embedded tweets include permalinks, popup "Follow" screens, and even "Reply", "Retweet", and "Favorite" links. It's literally "click, click, awesome."



Edit Post [Add New](#)

WordPress Auto-Embedded Tweet

Permalink: <http://bluefeed.net/wordpress/2012/wordpress-auto-embedded-tweet/>

Upload/Insert 

[b](#) [i](#) [link](#) [b-quote](#) [del](#) [ins](#) [img](#) [ul](#) [ol](#) [li](#) [code](#) [more](#) [lookup](#)

<https://twitter.com/perishable/status/250711304933896192>

3.6.5 Bug-fixes and enhancements

WordPress "Green" also includes a plethora of bug-fixes and enhancements, by the numbers:

- **401** bugs fixed
- **116** enhancements
- **3** requested features
- **52+** miscellaneous tasks

oEmbed

Visit section 2.2.3 for the scoop on embedding media in WordPress with oEmbed.

The easiest tutorial we've ever done: To embed a tweet, enter its URL in a post or page, and then done. For example, the screenshot at left produces the screenshot at right (using the Twenty Eleven theme).

WordPress Auto-Embedded Tweet

Posted on **September 25, 2012**



Jeff Starr
@perishable



Checking out WP's new auto-embedded tweets feature :D

25 Sep 12

 [Reply](#)  [Retweet](#)  [Favorite](#)

Note..

If using a theme other than one of the WP defaults (e.g., Twenty Eleven, Twenty Twelve, Twenty Thirteen), it may be necessary to add some CSS to style the embedded tweets. Check your theme's `readme.txt` for information.

Improvements in WordPress internationalization and localization

Much improvement has been made in WordPress' internationalization and localization functionality. **Better language support is a new focus** for the WordPress core team, with the development of language packs, integration of locale-specific modifications, and much more now in progress.

So far, the list of improvements is impressive. From better character localization to faster translations,

WordPress 3.4 brings better language support and a promise "that localized builds are a major player in the ecosystem." More information:

Moving Locale-Specific Modifications into Core
<http://digwp.com/u/623>

Important Changes for WordPress 3.4
<http://digwp.com/u/624>

Fresh Libraries

In WP 3.4, the jQuery library is updated to version 1.7.2, and other external libraries such as jQuery UI, TinyMCE, PHPMailer, SimplePie, and hoverIntent are also updated to latest versions.

While the bigger features such as the Theme Customizer tend to receive the most attention, the cumulative effect of all the minor improvements is what really makes WordPress 3.4 shine. It just *feels* like a better piece of software, with everything working together *better than ever*. Here are a few more for the road:

- Improvements to the contextual Help Tabs
- Spam comments are now excluded from the Dashboard
- It's now possible to add new comments from the Post Edit screen
- Nav Menus now show custom post-formats when supported by the theme
- Links widget now includes "Sort by" and "Number of Links" options
- Better performance for the "Recent Comments" dashboard widget
- RSS feeds now show the correct language
- Icons added for Retina devices

Details on WP 3.4

WP Codex – Version 3.4
<http://digwp.com/u/622>

Trac Milestone 3.4
<http://digwp.com/u/625>

Plus much more! Visit the sidebar links for all the nitty-gritty on the 3.4 update.

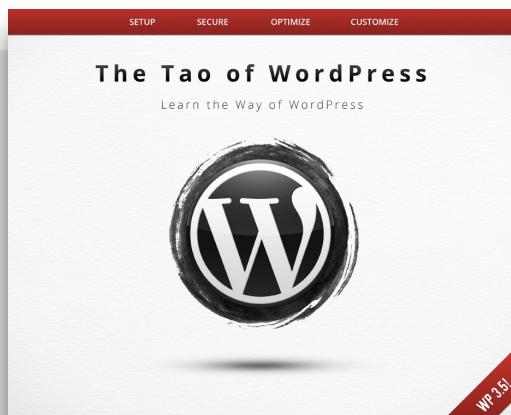
Until next time...

We hope you've enjoyed the **Bonus PDF** and found it useful in your WordPress adventures. It was a tough decision moving the last several chapters into their own supplemental book, but there are some great benefits to doing so:

- More focused and streamlined core content
- Plenty of room for new content in the supplemental book
- Fewer pages = less expensive to print the main book
- Fewer pages = lighter file size for mobile devices

You may be asking, "what about new sections for WP 3.5, 3.6, 3.7, and beyond?" Well that's the beauty of the new format. From now on, the core book will be updated as needed when awesome new features are rolled out. So instead of adding another ream of pages to the book for each new version of WordPress, the main content will be updated to reflect all of the latest and greatest improvements and changes — everything you need to know about WordPress in a concise, focused book. And even better, there is plenty of room in the supplemental book to further streamline and add content in the future.

If you have any questions about the content in either the main book or the supplemental PDF, feel free to drop us a line at sales@digwp.com. Thank you!



More WordPress!

Hungry for more? Check out my new book, **The Tao of WordPress** — the complete guide for beginners, students, admins, users, and anyone who wants a fresh, minimalist approach to rocking it with WordPress.

Check it out @ WP-Tao.com

DigWP.com

The companion site to the book, DigWP.com features news, updates, articles, and tutorials on WordPress.

<http://digwp.com/>

DigWP Downloads

Download the latest version of Digging Into WordPress and all the extras in the DigWP Member's Area:

<http://digwp.com/u/626>

WordPress + .htaccess

Shameless plug: Check out my new book on .htaccess! .htaccess made easy covers WordPress, site-performance, security, SEO & much more:

<http://htaccessbook.com/>

Thank you

Without a doubt, WordPress has revolutionized the way we build, operate, and even think about websites. We hope this book has given you a taste of the vast potential of WordPress and the endless possibilities that it provides.

What you do with WordPress is up to you, but we encourage you to use it in an intelligent and responsible way. With so much information available to WordPress users, it is easy to get lost in the details instead of focusing on what's important: sharing content with your readers.

One of the most exciting things about WordPress is that it makes creating and managing content so easy. Future versions of WordPress will continue to improve in terms of code, features, and potential. And we will be there to help you make the most of it all.

Thank you for sharing the journey with us :)



About the Authors



Chris Coyier



Chris Coyier is a real-world web designer who has been reaching for WordPress to power client sites for many years. He subscribes to the theory that not only is WordPress capable of powering any website it is almost always the right choice.

When he's not designing sites with WordPress, Chris Coyier shares his wisdom at [CSS-Tricks.com](https://css-tricks.com), a wildly popular design community focusing on CSS, web design, and everything in between.

Continue the journey...



Learn more about WordPress at the book's companion site, where we cover many more awesome tips, tricks, and techniques to help you get the most out of WordPress.

<http://digwp.com/>

Jeff Starr



Jeff Starr has been designing & developing WordPress-powered sites since 2005. He develops WordPress plugins, creates WordPress themes, and writes lots of articles about WordPress, web security, and designing with Web Standards.

In addition to writing about WordPress, Jeff also writes in-depth articles on web design and development at [PerishablePress.com](https://perishablepress.com), where he spends way too much time focusing on the details.

The Theme Clubhouse



Looking for a great theme for your WordPress site? Check out the growing collection of themes available at the DigWP Theme Clubhouse!

<http://themeclubhouse.digwp.com/>

W

RITTEN BY WORDPRESS VETERANS

Chris Coyier and Jeff Starr, *Digging Into WordPress* is packed with everything you need to get the most out of WordPress. Suitable for beginners, perfect for intermediate users, and a great comprehensive reference guide for advanced WordPress code wranglers.

WITH PLENTY OF JUICY CODE EXAMPLES, *Digging Into WordPress* teaches you how to improve your WordPress site using modern techniques, tips, and tricks. Using clear, easy to follow steps, *Digging Into WordPress* guides you with the tools and knowledge to build incredible sites that are powerful, secure, and awesome.

*Thanks for teaching
me to fish.*

-Eddie

<http://digwp.com>

Supplemental PDF
Included free with
book purchase.