



## **The Handbook**

### **Codex**

[http://codex.wordpress.org/Creating Admin Themes](http://codex.wordpress.org/Creating_Admin_Themes)

### **Version Date**

6 August 2005

# Creating Admin Themes

WordPress's flexible nature allows for almost every part of it to be easily changed. Creating a custom WordPress [Admin Panel](#) Theme is no different. There are essentially two ways of making a WordPress Admin theme: with a Plugin or by simply changing the CSS. The Plugin Method is the easier of the two methods, allowing you to install WordPress Admin Themes quickly and easily. You literally "plug it in" and it works.

If you are the creative type, we also have instructions to help you design your own style sheet for the Administration Panels, and even how to turn your [Admin Theme into a Plugin](#) for easy distribution to the public.

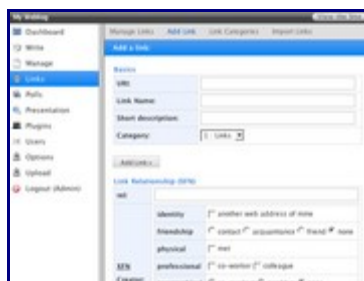
## Using Admin Theme Plugins

[WordPress Plugins](#) allow a user to easily add functionality to their blog without editing core WordPress files. There are several WordPress Plugins available that will allow you to use a WordPress Admin Theme with little or no fuss. Check the [List of Admin themes](#) and follow their instructions to change the look of your Administration Panels.

Typically, the instructions are to upload the Admin Theme Plugin to your plugins folder and then activate it on your [Plugin Panel](#). Simple and easy.

Plugin	Version	Author	Description	Action
<a href="#">SpotPress</a>	0.8b	<a href="#">Joe Kohlmann</a>	A cosmetic redesign for the WordPress admin pages that resembles the Spotlight search window in Mac OS X 10.4 Tiger.	<a href="#">Activate</a>
<a href="#">Blue Steel</a>	1.0	<a href="#">Mr. Wordpress</a>	A wacky custom Administration theme to replace the dull default style.	<a href="#">Deactivate</a>
<a href="#">Tiger Style Administration</a>	1.2	<a href="#">Steve Smith</a>	After working with WordPress for several months now, I felt the Administration area needed a little "freshening up". I wanted the utility to feel more like an application, and less like a traditional website.	<a href="#">Activate</a>

## Styling The Admin Theme



### Tiger Admin Theme

Whether you are designing your own Administration Panel Theme or creating one for public distribution as a WordPress Plugin, the process is basically the same. As a Plugin, the Admin Theme includes a function that will instruct WordPress to use a new stylesheet for displaying the Admin Panel. A distinct advantage to making a plugin is that if you ever want to revert back to the default style, all you have to do is deactivate the plugin. If you

are just making few changes and don't want to go through the hassle of creating your plugin, you might be better off just editing the CSS file directly.

## Direct Editing

Make a backup copy of the `wp-admin.css` file from the `wp-admin` folder on your WordPress site. Then you can go in and make changes in the new `wp-admin.css` file with a backup, just in case.

## The Admin Style Sheet



### Pink Administration Theme

The original Admin Theme style sheet is very complex, covering all aspects of the Administration Panels thoroughly. Not all the parts and pieces may need to be changed to recreate your Admin Theme.

Here is a partial list of the important style references used in the Administration Panels. Per CSS web standards, # denotes a style ID and . denotes a style class.

#wphead

The main title of the admin panel. Used to display the name of the blog and a link to **View Site**.

#adminmenu ul

The main level navigation bar, for links: Dashboard, Write, Manage, etc.

#adminmenu2 ul

The sub level navigation bar, for links (example: under Manage: Posts, Pages, Categories).

.wrap

The basic wrapper for all content in the admin panel, set in a `<div>`.

#zeitgeist

The sidebar on the Dashboard displaying Latest Activity and Blog Stats.

#footer

The footer at the bottom, with Wordpress logo, version number, and help links.

.wrap h2

Individual Page headers for the various subpanels, like **General Options**.

The changes you make in the `wp-admin.css` can be minor or extensive. You can just change the background color, add a background image to different sections, change the font, or even just do a minor color or design change to the [Quicktag buttons](#). It is up to you to use your imagination and create whatever effect you want.

## Creating an Admin Theme Plugin

To create an Admin Theme Plugin, it will need to be easily installed with little effort by the user, and easily uninstalled or deactivated, returning the [Administration Panels](#) to their original state.

We begin by telling WordPress to link to a new style sheet.

In a [text editor](#), in a new document, put the following:

```
<?php
/*
Plugin Name: Blue Steel Theme
Plugin URI: http://example.com/blue-steel-admin-theme
Description: Blue Steel WordPress Admin Theme - Upload and Activate.
Author: Mr. WordPress
Version: 1.0
Author URI: http://example.com
*/
?>
```

This is the "header" of the style sheet and provides information about the plugin to be viewed on the [Plugin Panel](#). It provides the name, [URI](#) of the plugin, the description, author, and version.

Create a folder on your computer, per this example, called `blue-steel`. Save this file inside of the folder and call it `blue-steel.php`. When you are ready, upload the entire folder to your plugin folder on your website at `/wp-content/plugins/`. This folder should be at `/wp-content/plugins/blue-steel/` and the plugin file should be at `/wp-content/plugins/blue-steel/blue-steel.php`.

To make this plugin call a new style sheet for the Administration Panels, we need to create a function that will add the style sheet to the head of the Admin Panel's header. This is no different than adding a link to a style sheet in all web pages. It will look something like this when the page is generated:

```
<link rel="stylesheet" type="text/css"
href="http://example.com/wp-content/plugins/blue-steel/wp-admin.css">
```

With your plugin, you will want to detect where the user has installed WordPress, so that you know where the rest of your Theme is located. You can use the `get_settings()` function for that. This makes our plugin flexible and portable. Here is how we would create the stylesheet link shown above:

```
<?php
/*
Plugin Name: Blue Steel Theme
Plugin URI: http://example.com/blue-steel-admin-theme
Description: Blue Steel WordPress Admin Theme - Upload and Activate.
Author: Mr. WordPress
Version: 1.0
Author URI: http://example.com
*/

function mr_blue_steel() {
    $url = get_settings('siteurl');
    $url = $url . '/wp-content/plugins/blue-steel/wp-admin.css';
    echo '<link rel="stylesheet" type="text/css" href="' . $url . '" />';
}

?>
```

Before we get to the actual styles, you need to add an action with the [Plugin API](#). Actions allow for plugins to "hook" into functions and features of the program. For Admin Themes, you want to hook into the `admin_head` (called in the `<head>`) of the Administration Panel with `add_action()`:

```
function mr_blue_steel() {
    $url = get_settings('siteurl');
    $url = $url . '/wp-content/plugins/blue-steel/wp-admin.css';
    echo '<link rel="stylesheet" type="text/css" href="' . $url . '" />';
}

add_action('admin_head', 'mr_blue_steel');

?>
```

In addition to the `admin_head` Plugin API hook, you can also optionally add a function to add content to the `admin_footer`. For example, you might want to put a notice about the theme in the footer. Here is how to add it to your plugin:

```
function blue_steel_footer() {
    echo 'This theme was made by <a href="http://example.com">Mr.
WordPress</a>.';
}

add_action('admin_footer', 'blue_steel_footer');
```

Save the plugin and upload it to your site. Select it from the Plugins Panel and see if anything about `blue-steel` appears. If it does, you are on the right track!

## Advanced CSS Styles



ID	Name	Description	# Posts	Actions
1	Articles		5	<a href="#">Edit</a> <a href="#">Delete</a>
2	Articles	For smaller articles	13	<a href="#">Edit</a> <a href="#">Delete</a>
3	General		48	<a href="#">Edit</a> <a href="#">Delete</a>
4	Science		0	<a href="#">Edit</a> <a href="#">Delete</a>
5	Code		1	<a href="#">Edit</a> <a href="#">Delete</a>
6	Political		0	<a href="#">Edit</a> <a href="#">Delete</a>
7	Real		0	<a href="#">Edit</a> <a href="#">Delete</a>
8	School		0	<a href="#">Edit</a> <a href="#">Delete</a>
9	Science		0	<a href="#">Edit</a> <a href="#">Delete</a>
10	Sports		1	<a href="#">Edit</a> <a href="#">Delete</a>
11	Technology		0	<a href="#">Edit</a> <a href="#">Delete</a>

### Rounded Corners

Sometimes there are places where CSS just can't achieve the look you want without making modifications to the HTML of the Administration Panels.

A popular CSS style is to create rounded corners on "boxes" of content. The technique involves adding divisions or *wrappers* to the HTML architecture in order to achieve the effect. Since we really do not want to get into the core Administration Panels to make these changes, which will disappear with the next upgrade, you can use the DOM (Document Object Model). The DOM is a way of dynamically accessing and updating content, structure, and style of documents.

In this example, using the [Transparent Rounded Corners](http://www.456bereastreet.com/archive/200505/transparent_custom_corners_and_borders/) ([http://www.456bereastreet.com/archive/200505/transparent\\_custom\\_corners\\_and\\_borders/](http://www.456bereastreet.com/archive/200505/transparent_custom_corners_and_borders/)) effect from 456 Berea Street, you can add the Javascript provided on the site to your Admin Theme Plugin, without editing the Wordpress source.

Download the script and save it to your `blue-steel` folder as `javascript.js`. Change

the `mr_blue_steel()` function to:

```
function mr_blue_steel() {  
    $url = get_settings('siteurl');  
    $dir = $url . '/wp-content/plugins/blue-steel/';  
    echo '<link rel="stylesheet" type="text/css" href="' . $dir . 'wp-admin.css"  
>';  
    echo '<script type="text/javascript" href="' . $dir  
<script>';  
}
```

This script uses a single "hook" (`cbb`) to create many divisions around the container. For this to work, open the Javascript file and change all references of `cbb` to `wrap` to match the `wp-admin.css` standard style references.

## ***Resources***

- [List of Admin themes](#)
- [Writing a Plugin](#)
- [Plugins](#)
- [Blog Design and Layout](#)
- [WordPress and CSS](#)