Advanced Guide to Branding cPanel

David Grega

Reminders

All trademarks within are the sole property of their respective owners.

Introduction

My name is David Grega. I have been handling Sales Support here at cPanel since 2006.

My job responsibilities include knowing the capabilities of our software and ensuring you know about the capabilities of our software. Therefore, if you need to know what our software can do, talk to me.

What Will be Discussed?

- Customizing the appearance of the cPanel interface
- How Plugins can assist with branding.
- Customizing the Webmail interface
- Customizing the Login and Logout screens

Deploying Your Customizations

What We Will Be Skipping Today

- Introduction to the Branding Editor
- Internationalization
- In-depth discussion of our APIs
- Detailed discussion of how to create a cPanel Plugin

Basic Terminology

Styles, YUI and API2

What is a Style?

A Style is your company's branding. This can include your logo, customized icons and even a fully customized appearance of the cPanel interface to convey your company's brand and re-enforce brand loyalty.

The branding is stored in what is known as a Style which can then be applied to other users.

Different styles may have different appearances, but will retain the same underlying functionality.

What is YUI?

YUI is the Yahoo User Interface library. This Javascript library is used to power the AJAX components of the graphical user interface.

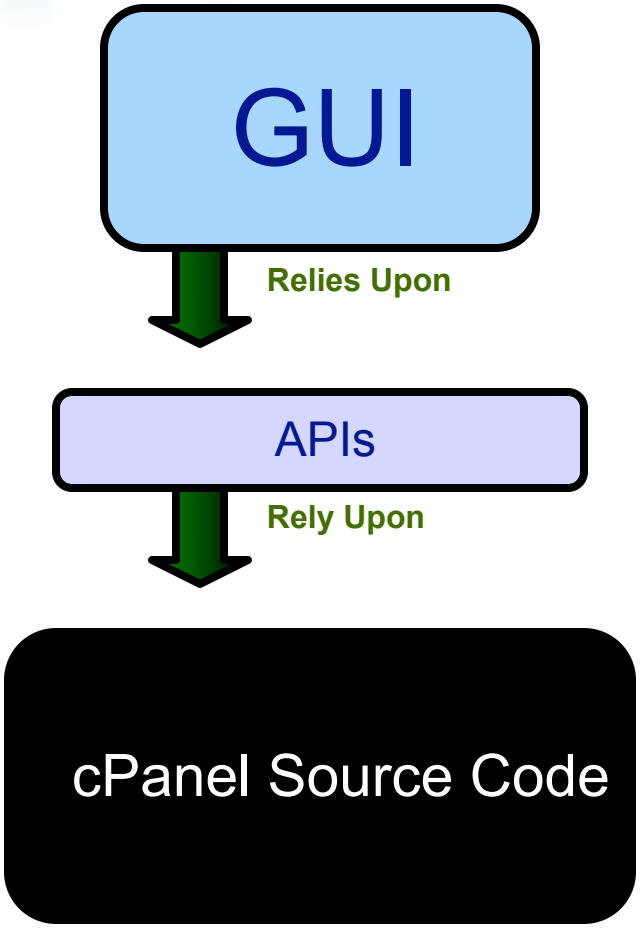
Knowing how to use YUI is useful if you wish to add additional AJAX functionality to the cPanel interface.

For more information about YUI, refer to their documentation which is available at: http://developer.yahoo.com/yui/

What is API2?

API2 allows components of the GUI (including Styles) to indirectly interact with the cPanel Source Code.

Customizing API2 calls from the Branding Editor can substantially modify the appearance of the cPanel interface.



Documentation for API2

Official API2 documentation is available on our website at: http://www.cpanel.net/plugins/api2.

Last year's “Customizing cPanel” presentation available at http://www.cpanel.net/support/docs has an introduction to API2.

What About API1?

Not all functions in the themes have been updated to API2, so there are some API1 functions remaining.

You will recognize API2 functions since they are typically (but not always) of the format Module::function rather than Module="function...

API1 code can be used wherever API2 code is permitted. However, API2 is significantly preferred.

About Styles

Styles Belong to Resellers

Styles belong to reseller users (“resellers”). Each reseller may have their own styles to reflect their company's brand.

Individual hosting account owners do not have the ability to edit their branding.

Information on how to deploy your styles to all resellers and your other cPanel/WHM servers is later in this presentation.

Where are Custom Styles Stored?

`/home/reseller/cpanelbranding/theme/style`

- `reseller` is your home directory as a reseller.
- `theme` is the theme you have created a style for, typically it will be `x3`.
- `style` is the name of your style.

Styles are Theme-Specific

Styles only work for the theme for which they are designed.

Today's presentation addresses creating a style for the x3 theme.

However, you cannot design a style for x3 then apply it to another theme such as a theme included with RVSkins.

Styles and Theme Managers

Theme Managers such as the Universal Theme Manager built into WHM do not have the ability to manage individual Styles.

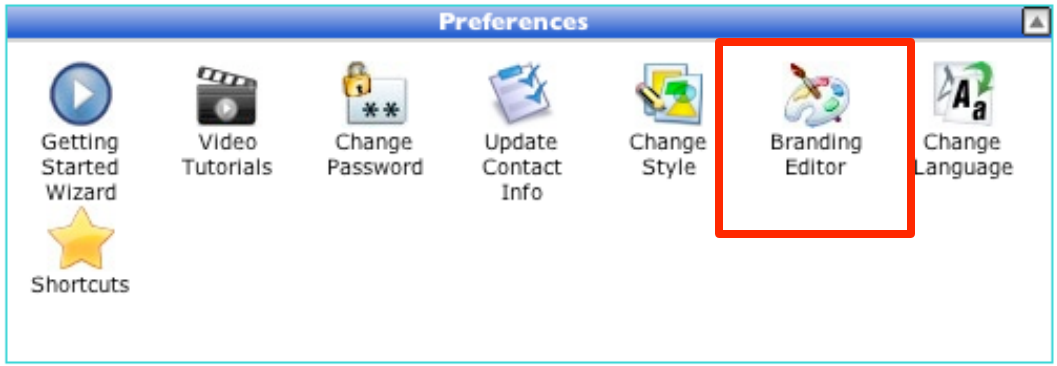
Theme Managers can only manage the Themes the Styles rely upon.

Advanced Functionality of the Branding Editor

What Can The Branding Editor Do?

The Branding Editor is a feature of the cPanel interface that allows resellers and root users to edit styles.

The Branding Editor is typically all you need to do for substantially customizing the appearance of cPanel.

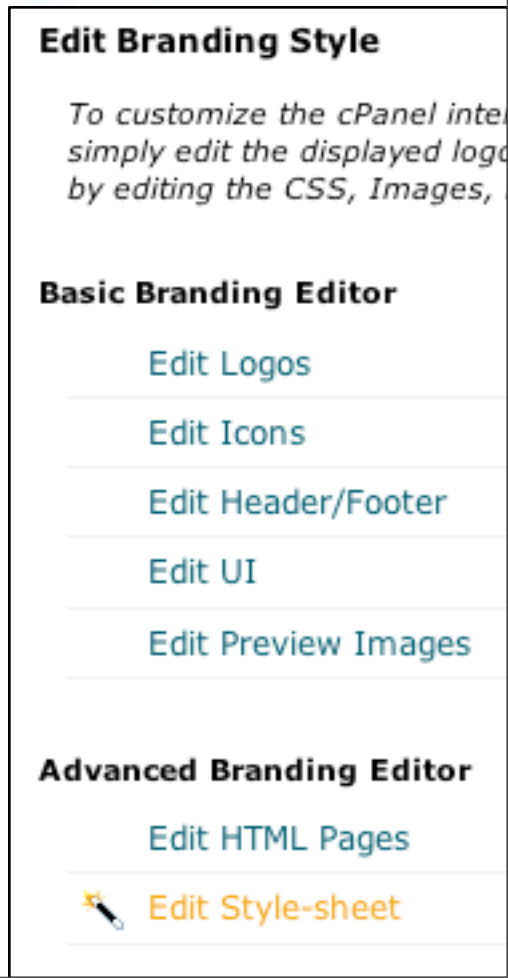


Style-Sheet Editor

CSS is a web design standard governing the appearance of web pages.

Since the cPanel interface is merely a series of web pages, you can use CSS on the cPanel interface as you would a typical web page.

This editor allows you to modify the CSS style-sheet of the style and override any other CSS style-sheets.



How Do We Know What CSS To Edit?

As with most contemporary web-based applications, the CSS code for cPanel is elaborate.

As a result, it may be difficult to determine what CSS code needs to be modified to achieve the effects you desire.

Tools such as Firebug allow you to easily know what CSS code needs editing.

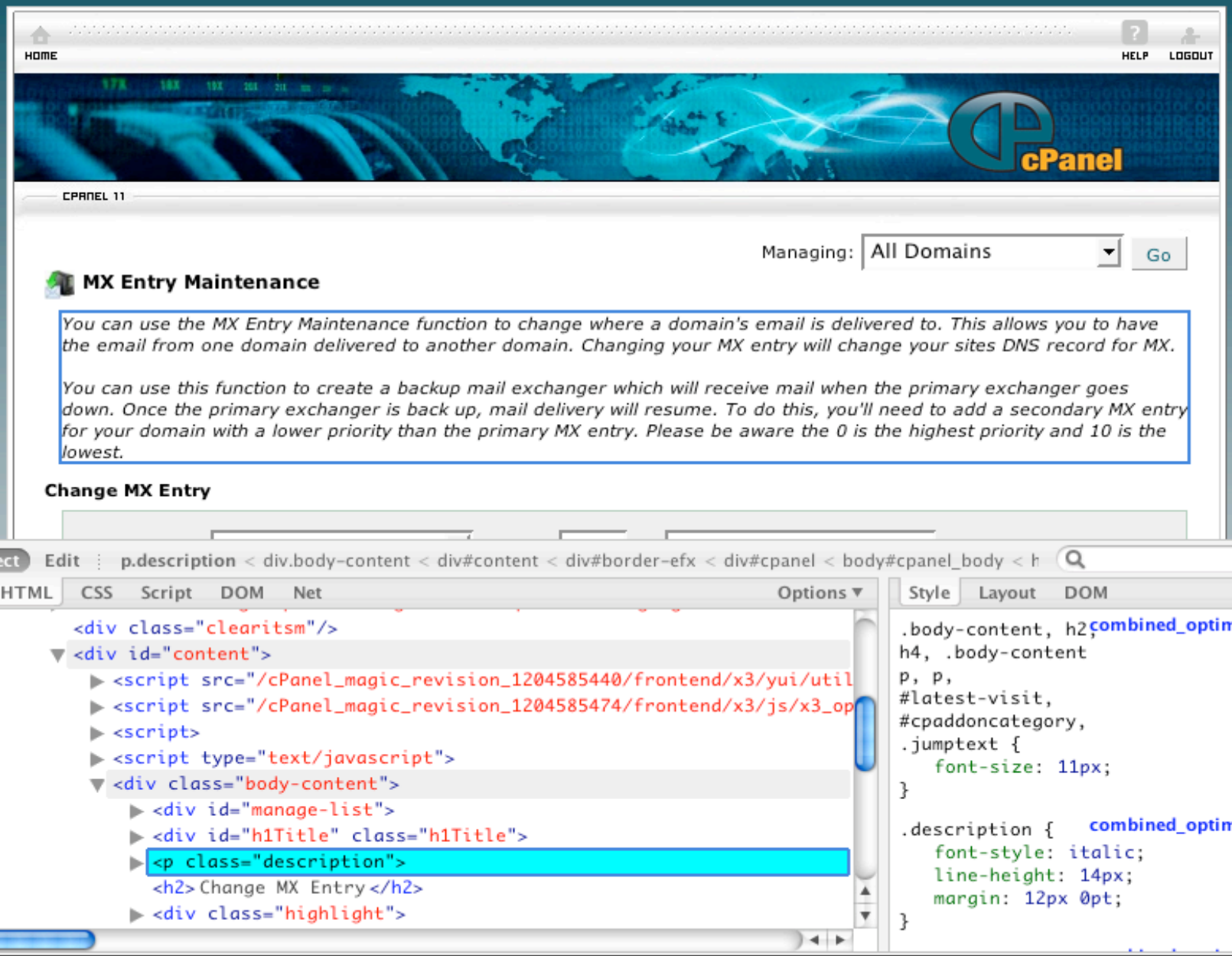
Firebug

Firebug is a Firefox extension that allows you to quickly determine what CSS code governs nearly any object you see on a web page.

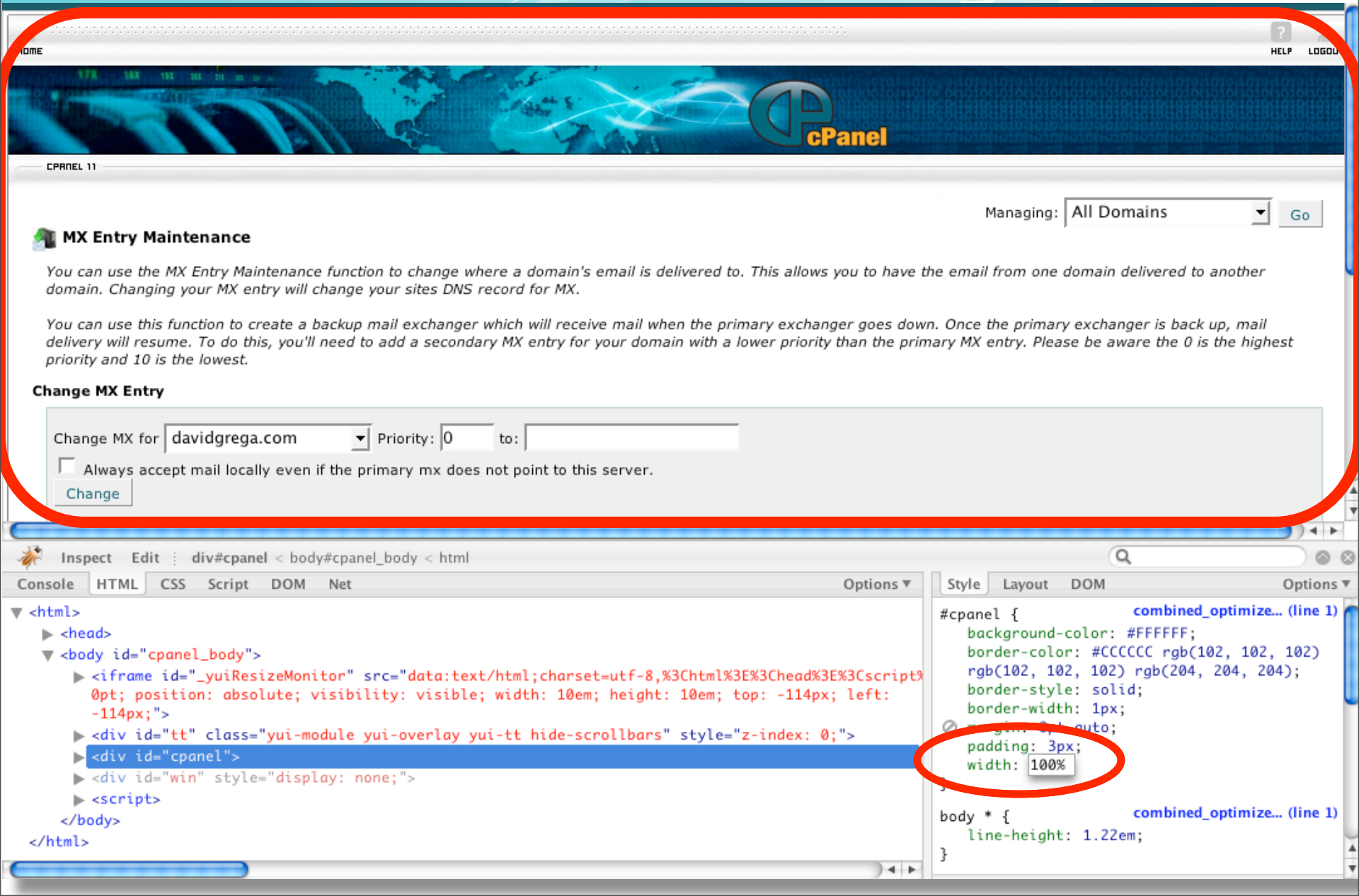
This extension will also allow you to experiment with new CSS code before modifying it on the server. As a result, it makes customizing a style's CSS much faster and easier.

This extension can be downloaded at: http://www.getfirebug.com/

Firebug in Action: Determining CSS Classes



Firebug in Action: Testing Modifications



Using Firebug with the Style-sheet Editor

It's easy to incorporate experimental changes you made to your branding with Firebug into your style!

Go to the Style-Sheet Editor in the Branding Editor and simply type in the CSS code you wish to include. For example, based off our last modification, you would enter:

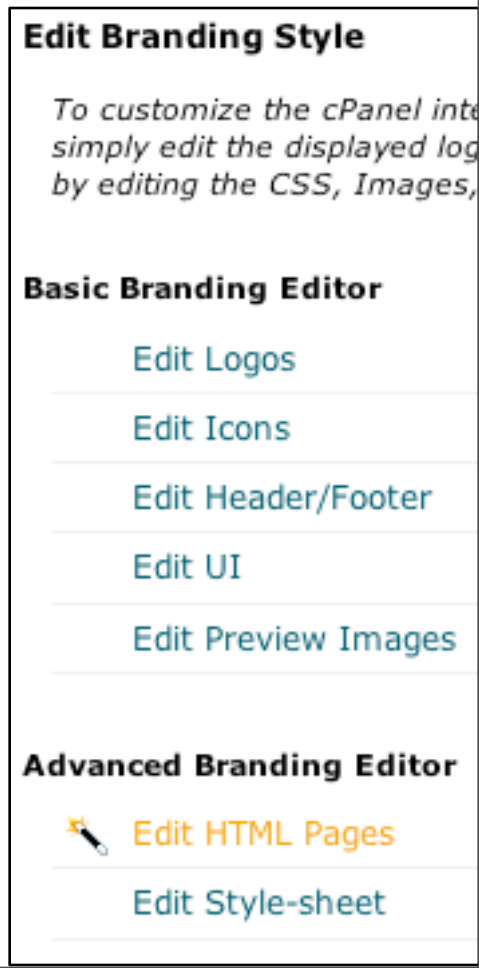
```
#cpanel {
    width: 100%;
}
```

Edit HTML Pages

This portion of the Branding Editor allows you to perform substantial customizations to the appearance of the cPanel interface.

By editing these HTML pages, you can transform the cPanel interface to mimic the appearance of your website.

All of these pages can contain API2 code.



Branding Editor

Edit HTML Pages

The HTML editor allows you to edit different areas of the cPanel interface. This can be used in combination with the CSS editor and changing the images to skin the interface into any look you desire.

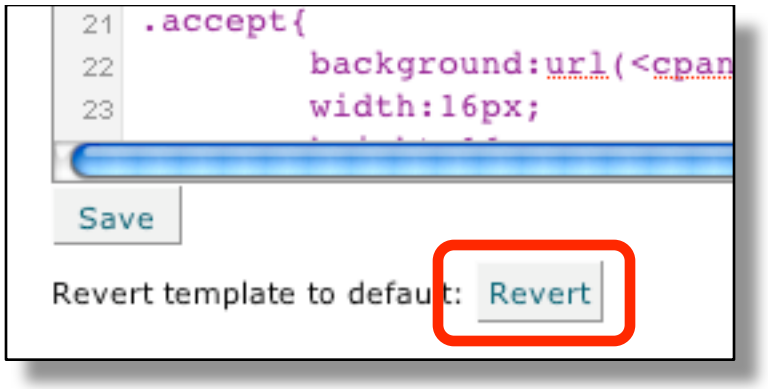
Current Style **[root]**

Description	File Name (Right Click and Save As...)	Edit	Upload	Reset to default
cPanel Parsed Inline CSS	cpanel_parsed_css.html	Edit	Browse...	Reset
cPanel Parsed Inline CSS (Local)	cpanel_parsed_css_local.html	Edit	Browse...	Reset
Getting Started popup Include	getstarted-pop.html	Edit	Browse...	Reset
Entry Page	index.html	Edit	Browse...	Reset
Option Selection popup Include	optionselect-pop.html	Edit	Browse...	Reset
All Sprites include	sprites.html	Edit	Browse...	Reset
UI Sprites include	sprites_ui.html	Edit	Browse...	Reset
YUI Sprites include	sprites_yui.html	Edit	Browse...	Reset
Footer	stdfooter.html	Edit	Browse...	Reset
Header Top Section	stdheader.html	Edit	Browse...	Reset
Header Middle Section	stdmheader.html	Edit	Browse...	Reset
Webmail Login Include	webmailloginclude.html	Edit	Browse...	Reset

Upload Pages

Don't Be Afraid To Make Mistakes!

In case you make a mistake and wish to revert back to the code our design team provided, click the **Revert** button on any of the screens where you can edit these HTML pages.



cPanel Branding Tags?

These API2 functions reference files stored **in the path of that particular style** and includes the appropriate HTML content into the page.

```
<cpanel Branding="image(cp)">
```

- Checks for image file cp that ends in .gif .png or .jpg

```
<cpanel Branding="text(supportinfo)">
```

- Includes the text stored within supportinfo.txt

Can insert HTML code here.

Why Use Branding Tags?

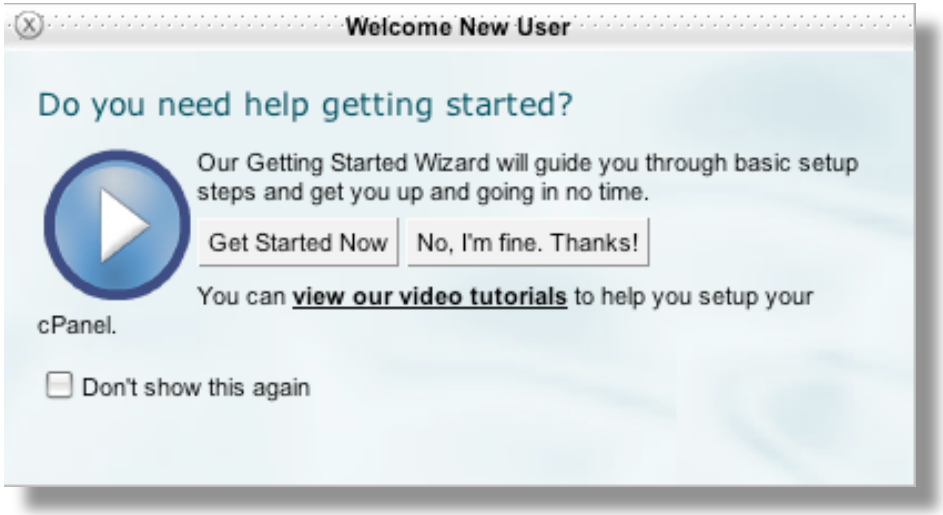
Better Performance!

By placing these files in the style's directory and calling them with cPanel Branding tags, cPanel can cache these into sprites and thus significantly improve the performance of the x3 interface.

As a result, it is recommended you use cPanel Branding Tags rather than directly link to existing images on your website.

Getting Started Popup Include

This is the pop-up your users see the first time they login to the cPanel interface.



One common reason for editing this page is to display the EULA the first time a customer logs into their cPanel interface.

Entry Page

The page users see when they first login.

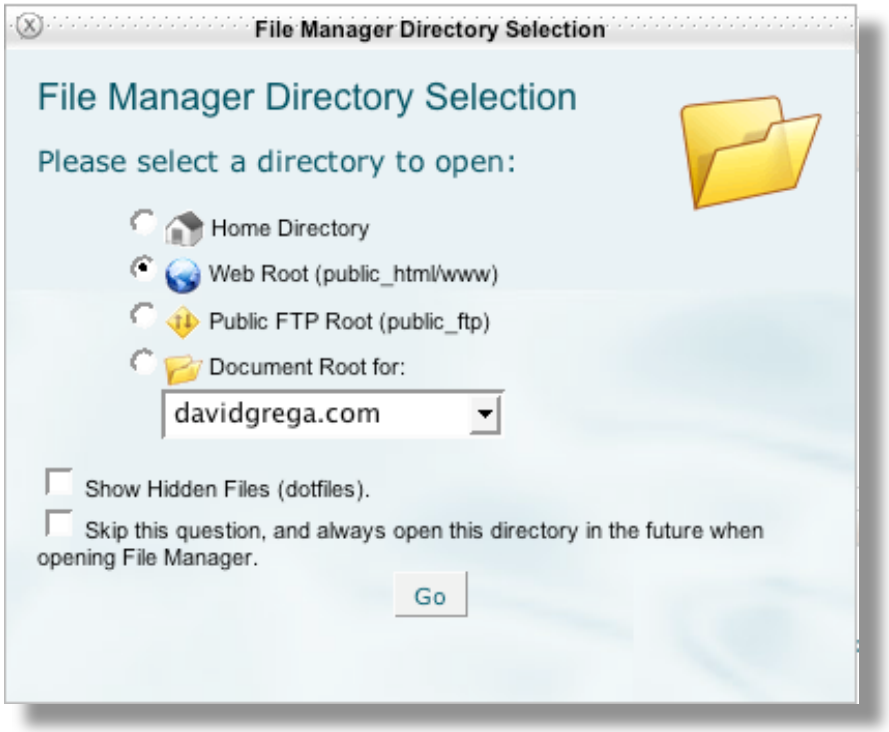
Entry Page – Custom Stats

To hide information from your users, simply remove it from the appropriate API call. Removing this red text will hide OS information.

```
<?cp StatsBar::stat([tr class="row-%"][td
width="40\{percent}" align="left"][span
class="stat-head"]%[/span][/td][td align="left"
class="truncate" ]%[/td][/
tr],rowtype,item,value) display=hostingpackage|
shorthostname|cpanelversion|cpanelrevision|
theme|apacheversion|phpversion|mysqlversion|
machinetype|operatingsystem|sharedip|
dedicatedip|sendmailpath|perlpath|
kernelversion,rowcounter=mainstats ?>
```

Option Selection Popup Include

This is the pop-up users see the first time they click on the File Manager icon.



Footer

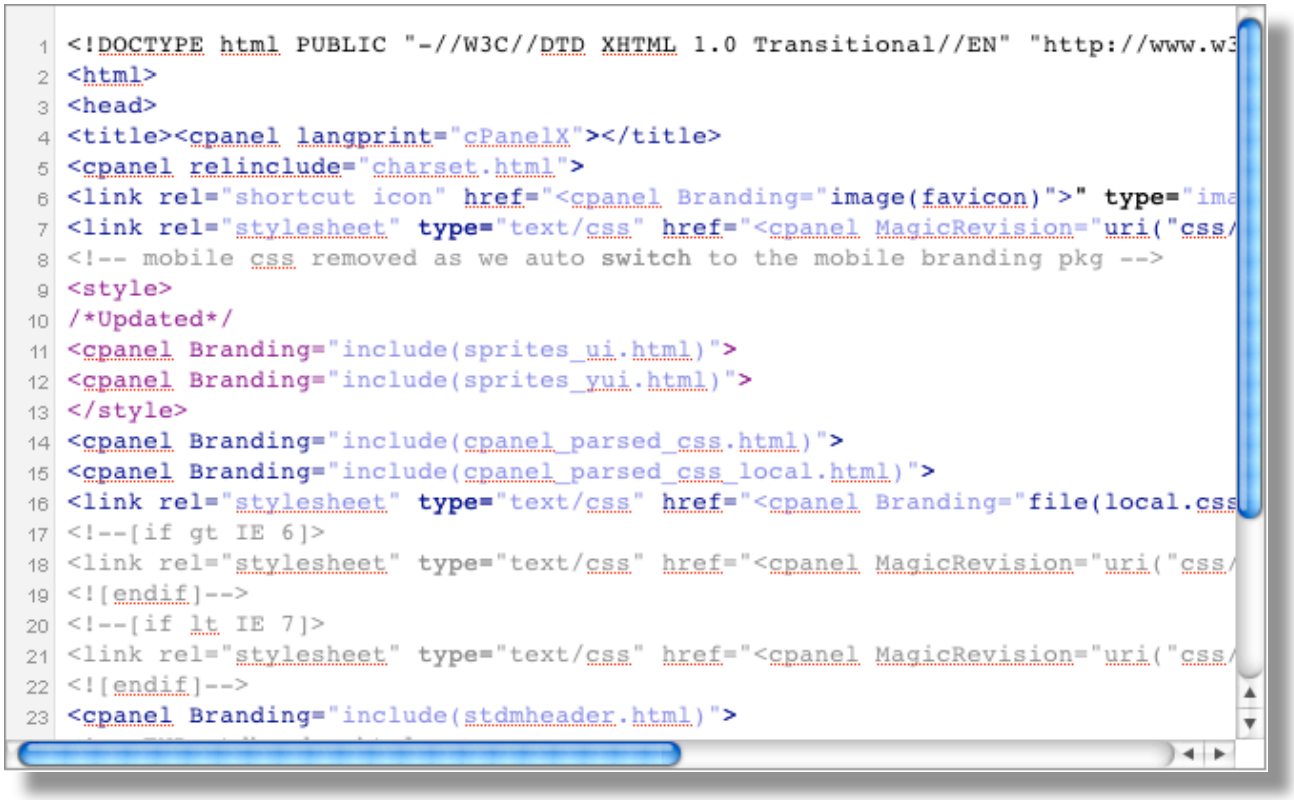
Stores the code for the footer of the page.

```
<!-- START stdfooter.html -->
<div class="clearitmed"> </div>
 </div><!--Close Content Div-->

<!--Start Footer-->
                <div  id="heading-bottom_bg" class="spriteheading_bg footer">
                        <ul class="links" >
                                <li><a href="<cpanel printvar="dprefix">index
                                <li><span  class="footer" style="font-size:18
                                <li><a href="<cpanel printvar="dprefix">trade
                                <li><span  class="footer" style="font-size:18
                                <cpanelif !$CPVAR{'hidehelp'}>
                <li><a onClick="show_help(); return false;" target="_blank" h
                </cpanelif>
                                <li><span style="font-size:18px;">•</span></l
                                <li><a href="http://www.cpanel.net/docs/cpane
                                <li><span style="font-size:18px;">•</span></l
                                <li><a href="/logout/" class="bottomlink"><cp
                        </ul>
                </div>
                <!--End Footer-->
        </div><!--Close border-efx div-->
</div><!--Close cpanel div-->
```

Header Top Section

Stores the code for the page up to the `</head>` tag.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3
<html>
<head>
<title><cpanel langprint="cPanelX"></title>
<cpanel relinclude="charset.html">
<link rel="shortcut icon" href="<cpanel Branding="image(favicon)">" type="ima
<link rel="stylesheet" type="text/css" href="<cpanel MagicRevision="uri("css/
<!-- mobile css removed as we auto switch to the mobile branding pkg -->
<style>
/*Updated*/
<cpanel Branding="include(sprites_ui.html)">
<cpanel Branding="include(sprites_yui.html)">
</style>
<cpanel Branding="include(cpanel_parsed_css.html)">
<cpanel Branding="include(cpanel_parsed_css_local.html)">
<link rel="stylesheet" type="text/css" href="<cpanel Branding="file(local.css
<!--[if gt IE 6]>
<link rel="stylesheet" type="text/css" href="<cpanel MagicRevision="uri("css/
<![endif]-->
<!--[if lt IE 7]>
<link rel="stylesheet" type="text/css" href="<cpanel MagicRevision="uri("css/
<![endif]-->
<cpanel Branding="include(stdmheader.html)">
```

stdmheader.html

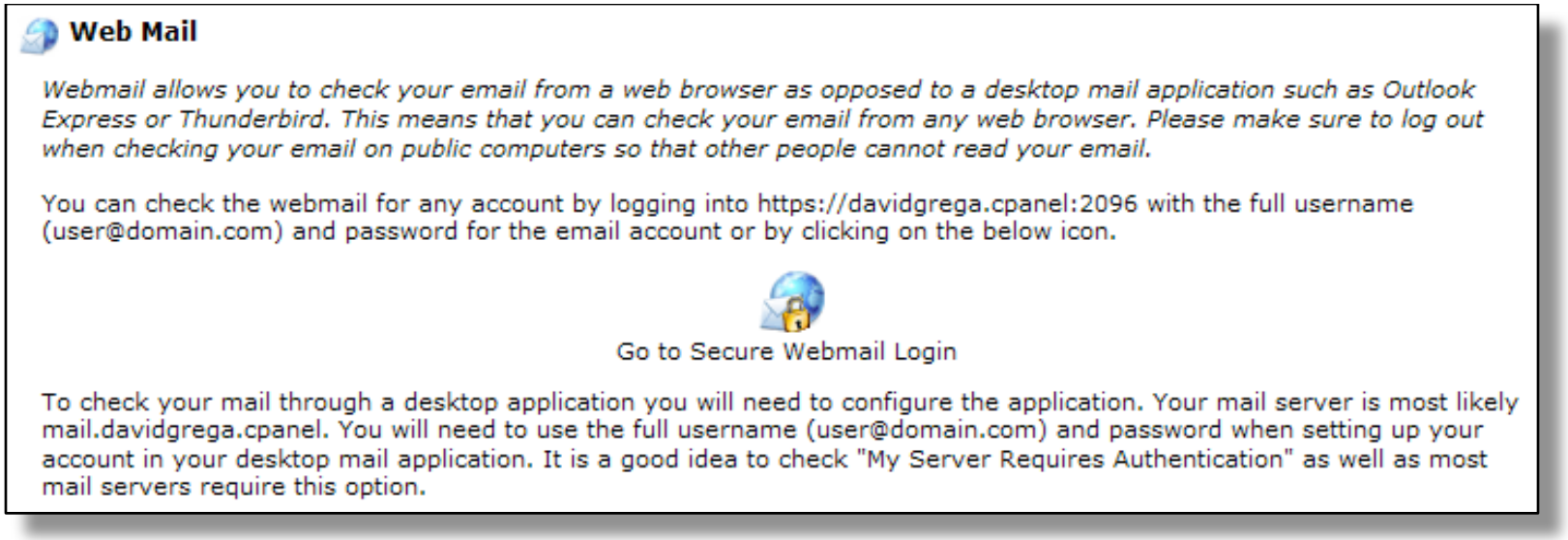
Header Middle Section

Controls the top visible portion of the page.

Webmail Login Include

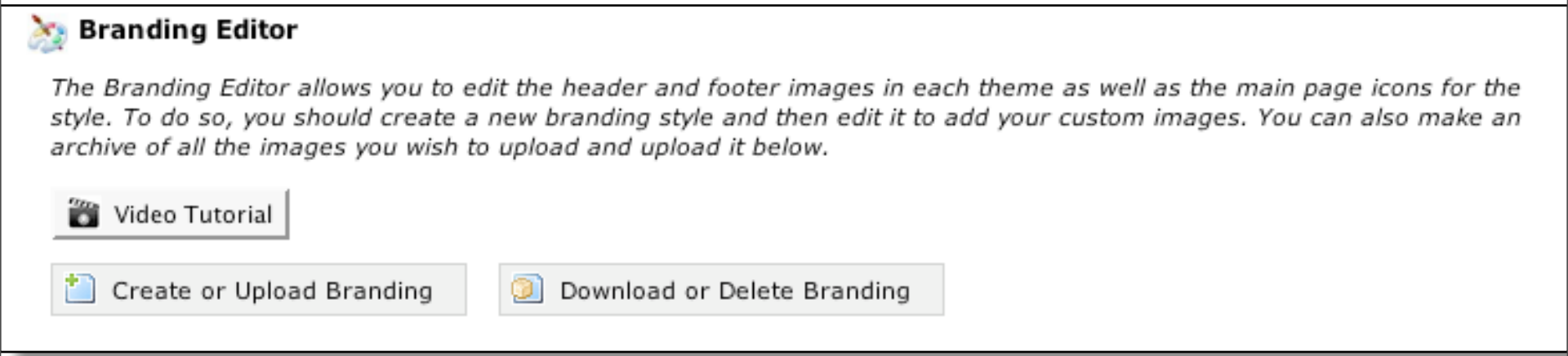
What a user sees when they click on “Webmail” via the cPanel interface.

Useful to modify if you are hosting mail on an unsupported external server.



Deploying Your Custom Styles

Copying Styles to Another Server



The Branding Editor provides utilities to download your style as a file and upload a style created on another server to your server.

Deploying Styles Server-Wide

`/usr/local/cpanel/base/frontend/branding/theme/style/`

- Copying your styles here will make the style available to all accounts on the server, where `theme` is the theme the style is designed for (typically x3) and `style` is the name of your style.

➤ This eliminates the need to download then upload the style to all of your resellers' accounts.

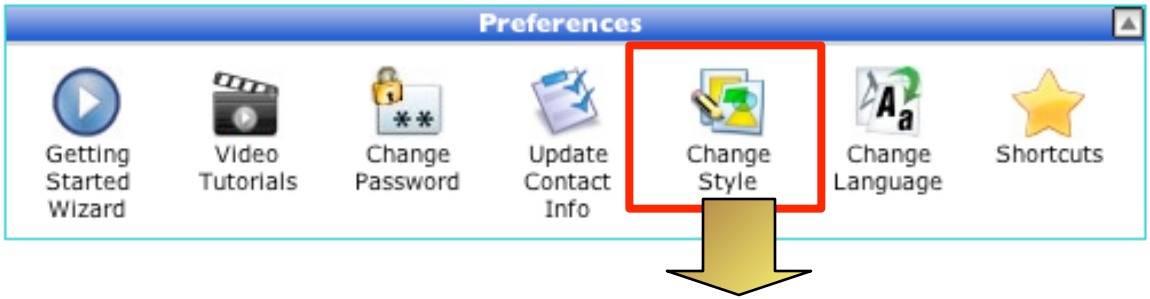
Edit Preview Images

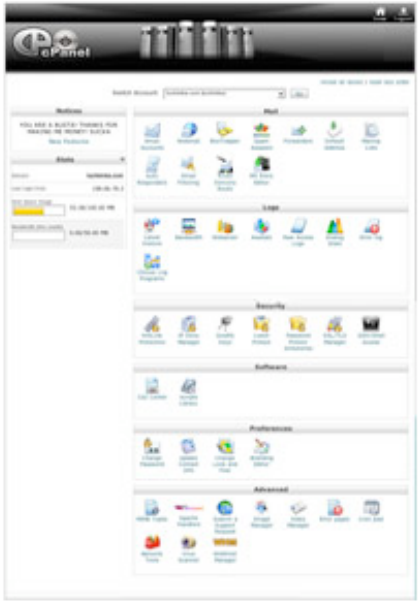
This is especially useful if you intend to distribute your styles.

Preview images are not automatically generated.

Here, you can upload screen captures of what your theme looks like. This will show users in the style selection screen what their interface will look like before they select your style.

Use of Preview Images





black_ice

falltheme

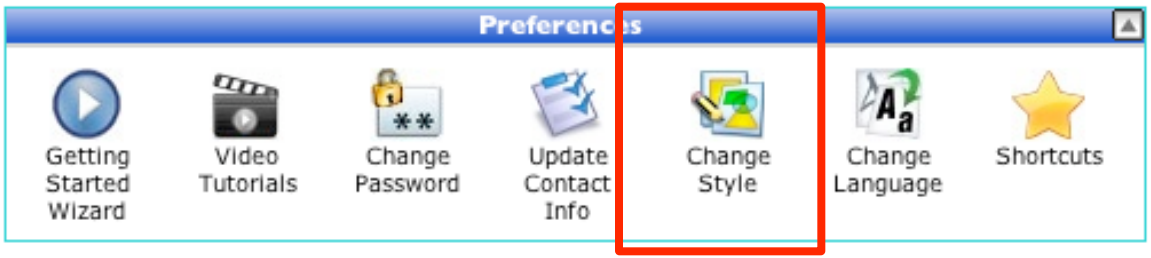
crimson_smoke

business2business

Forcing the Use of Your Style

Simply apply your style to all users, then remove the ability for them to use the “Change Style” feature from Feature Manager.

This is useful if your style reflects branding that would not be offered in other styles.



Plugins and Branding

Simply Adding a Hyperlink?

If you simply want to add a hyperlink with an icon to the cPanel interface, create a cPanel Plugin. It is very easy with our Plugin Generator!

You can find the Plugin Generator on our website at: http://www.cpanel.net/plugins/devel/installgen.html

cPanel Plugins Install File Generator

Plugin Name: my_icon

Name of the Plugin.

Module Description: Billing System Link

Description of the module (used for branding).

Feature Manager? ☑

If enabled, the feature will be an option in the feature manager.

Display Name: Pay Hosting Bill

Name displayed in the cPanel Interface.

Display Group: Preferences

What cPanel icon group the icon should appear in. (Currently only in x3).

Display Order: 999

Order within the group.

Access URL: my_icon/index.html

URL used to access the plugin in cPanel (relative to /usr/local/cpanel/base/frontend/<theme_name>/).

Anchor Content:

Data in this field will be included in the anchor tags, you can use something like target="_blank" etc.

On Click Event: pcation='http://example.com/billing';return false;

Data in this field will be included in the anchor tags, you can use something like —JavaScript_HERE_; return false;

Plugin Icon: Browse...

Icon to be displayed in cPanel (jpg or png only).

submit

Explanation of the Javascript Code

```
window.location='http://example.com/billing';return false;
```

This Javascript code goes in your On Click Event content.

The `window.location` indicates where you want the user to go when the icon is clicked.

The `return false` simply tells your web browser to go to that location instead of to the path we said was the plugin path.

Installing Your Plugin

Simply follow these instructions to install your plugin:
http://www.cpanel.net/plugins/devel/install.html

Be sure to follow the instructions carefully and execute those commands as root.

You can execute the commands as sudo, but be sure to execute every command as sudo if you do.

Adding Additional Functionality

Plug-ins can be used for purposes beyond adding simple hyperlinks.

You can add entirely new pages and functionality to the cPanel interface using Plug-ins.

Why Plug-ins are Preferred

Plug-ins are style and theme independent. This means the same functionality will be available if you use the themes bundled with cPanel/WHM or if you use a third party theme such as those bundled with RVSkins.

This theme independence also means your plug-ins are likely to continue functioning without maintenance for years to come.

Changing and Removing Functionality in the cPanel Interface

Why Remove Functionality?

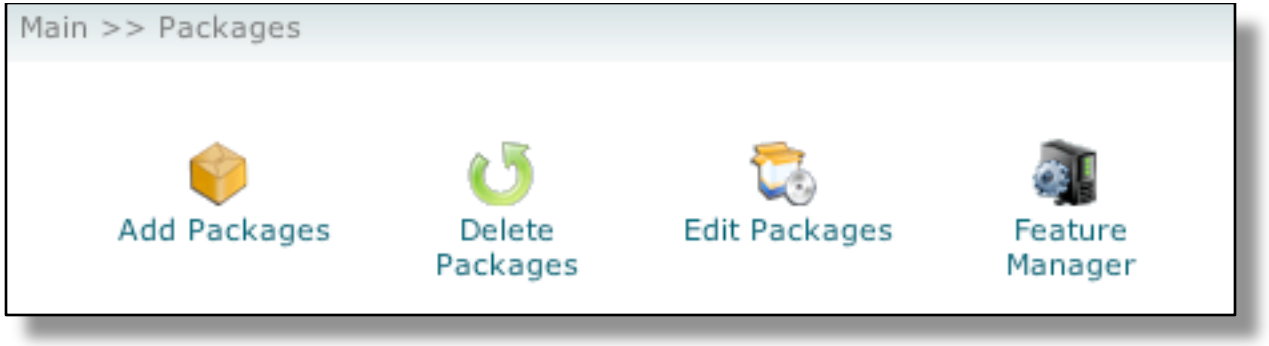
Some hosting providers prefer to charge for specific functionality. For example, offering very inexpensive hosting with limited features for those who are new to running their own website.

Sometimes technical limitations of one's unique setup requires the removal of certain functionality to prevent their users from becoming confused.

Removing Functionality

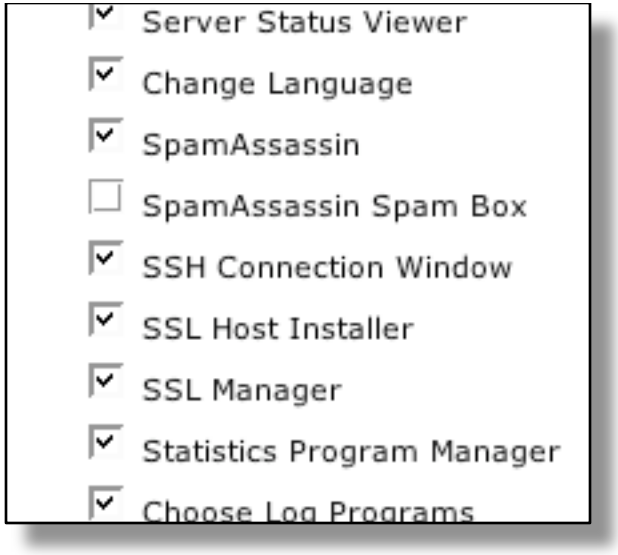
You can disable Plug-ins as well as native functionality of cPanel/WHM via Feature Manager by using Feature Lists.

Feature Lists can remove functionality on a per-package basis or on a server-wide basis.



Feature Lists?

Feature Lists allow you to turn any icon in the cPanel interface on or off individually.



Server-Wide Feature Lists

There are 2 special Feature Lists that control available functionality on a server-wide basis:

Disabled

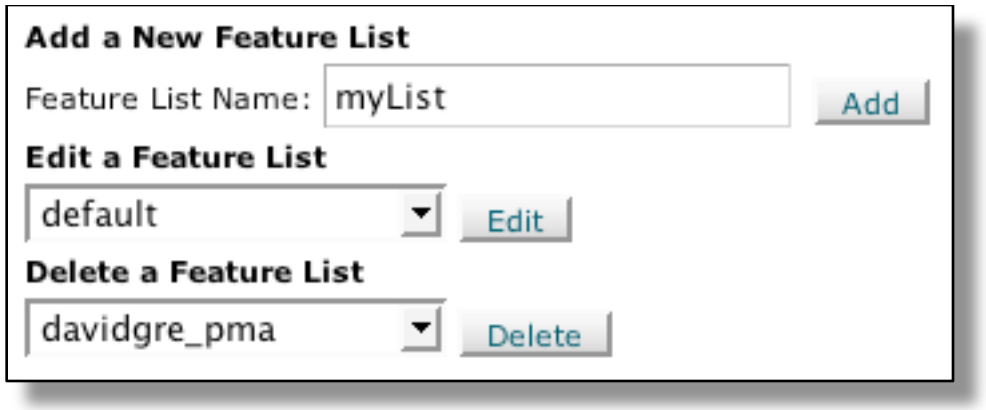
If something is disabled in the “Disabled” feature list, then it is disabled for all users.

Default

Governs the functionality available to users that do not have a Feature List assigned to the Package associated with their account.

Package-Specific Feature Lists

You can also create your own Feature Lists using Feature Manager.



Your own Feature Lists can then be associated with Packages.

Applying Feature Lists to Packages

Feature Lists cannot be applied to individual accounts, they must be applied to Packages.

You can do this when editing or creating a Package in the WHM interface:

Customizing the Webmail Interface

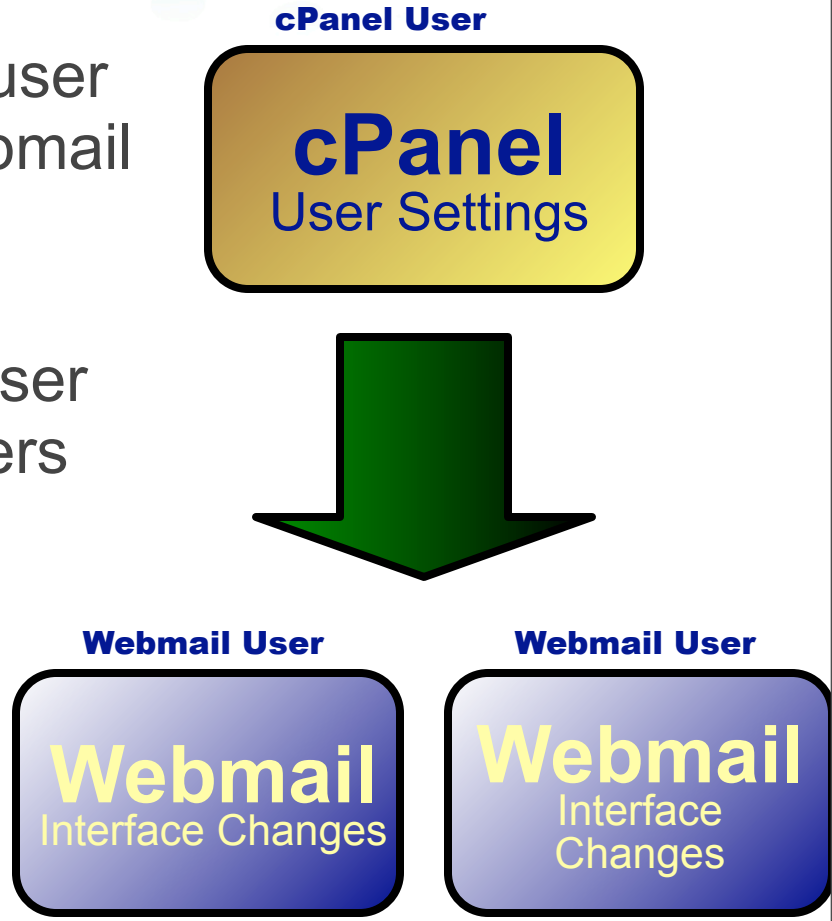
Changing Available Webmail Clients

If you wish to control which webmail clients you want your users to be able to use, simply uncheck the mail clients you do not wish for them to use in the WHM interface under Tweak Settings.

Webmail Interface Inheritance

- Features disabled for a cPanel user will also be disabled for any webmail users under that cPanel user.

- The style used by the cPanel user is also used by any webmail users owned by that cPanel user



Custom Login and Logout Themes

Not to be Confused With...

... embedding a cPanel, WHM or Webmail login onto your existing website.

If you desire to do that, please view the sample code available on our website at:

www.cpanel.net/cpanel-login.phps

Common Mistake

DO NOT

SIMPLY EDIT AN EXISTING THEME!

Your customizations will be overwritten during the next cPanel update if you do this.

Always create a new theme rather than modify a theme bundled with cPanel/WHM.

Login Themes

default theme: cpanel

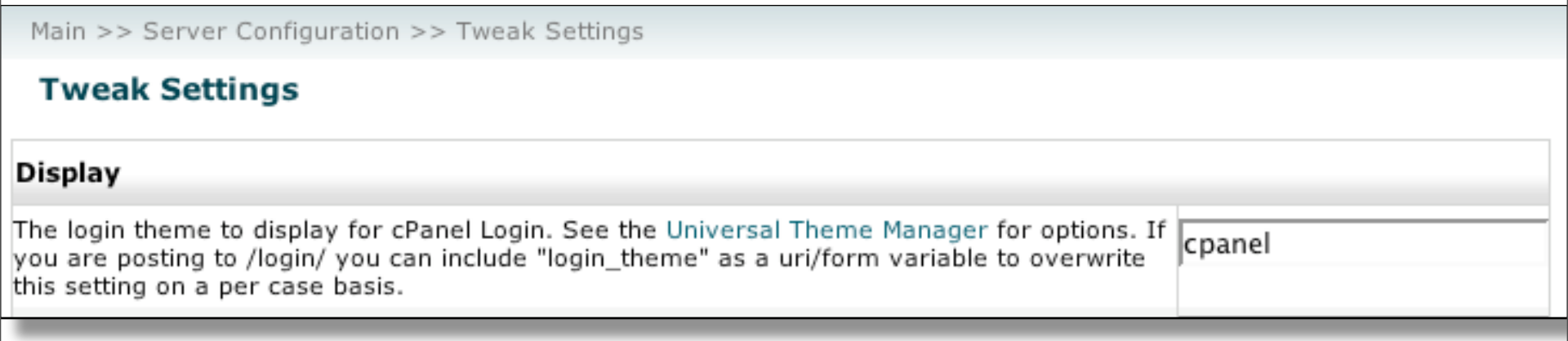
Stored in: `/usr/local/cpanel/base/unprotected/cpanel`

To create your own theme, simply copy our existing theme (cpanel) and paste it into a similar directory, for example:

`/usr/local/cpanel/base/unprotected/mycpanel`

Setting the Login Theme

You can set the login theme to use via Tweak Settings.

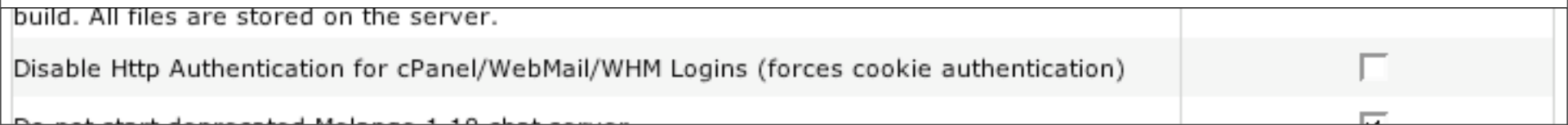


login Page

Users will only see the login page visible if you disable HTTP Authentication in Tweak Settings.

HTTP Authentication is more secure, hence it is the default setting.

At the theme level, there are different files that govern the login pages for the cPanel, WHM and Webmail interfaces.



Common File Suffixes

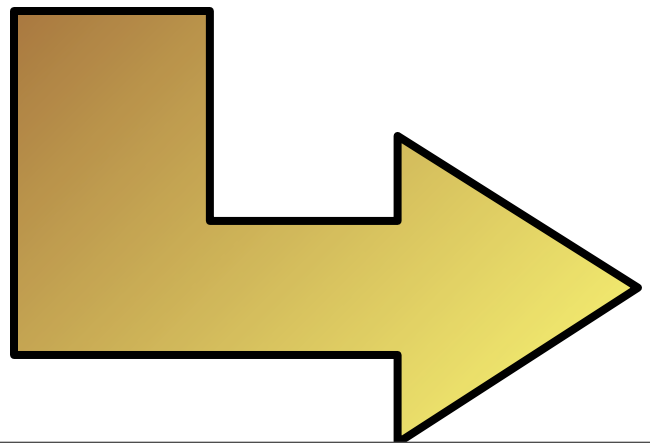
- _cpaneld
  Used when logging into cPanel.
- _whostmgrd
  Used when logging into WHM.
- _webmaild
  Used when logging into the webmail interface.

`login` files

The code is displayed within the context of the Login theme.

These pages should only contain the code for the actual login form.

```
<form action="/login/" method="POST">
<input type="hidden" name="login_theme" value="%login_theme%">
<table width="200" class="login" cellpadding="0" cellspacing="0">
<tr>
<td align="left"><b>Login</b></td>
<td> </td>
</tr>
<tr>
<td>Username</td>
<td><input type="text" name="user" autocomplete="off" size="16"></td>
</tr>
<tr class="row2">
<td>Password</td>
<td><input type="password" name="pass" size="16"></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="sub
</tr>
</table>
</form>
```



`logout` Page

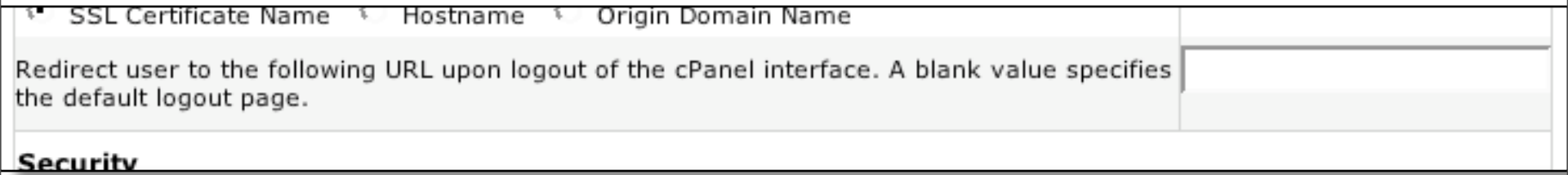
Content of logout messages are controlled by the cPanel/Webmail/WHM interfaces.

The layout of logout pages governed by the login theme.

Logout URL

If you simply wish to redirect users to one of your web pages upon logging out, a custom theme is not necessary.

Simply change the **Logout URL** setting in Tweak Settings to redirect users to your web page upon logging out.



`header` files and `footer` file

Essentially these files work to wrap code around whatever message may be displayed by an individual file.

These files provide the overall theming of the Login theme.

This is the core of what will make your login theme look unique.

Headers and Footers explained

Header

All the code that comes **before** the message being displayed by the Login interface.

Footer

All the code that comes **after** the message being displayed by the Login interface.

header files and footer file

Custom CSS and Images

You can edit the CSS for the Login theme by editing `style.css`

Custom images may be placed in the images folder of your Login theme.

When images are presented to the end-user, their browser will need to be instructed to look for the images in the relative directory of `unprotected/images` (not `images`).

Deploying Your Custom Login Theme Across cPanel/WHM Servers

This is a very easy process!

Simply copy the files to your other cPanel/WHM server(s) and set the login theme via Tweak Settings on each server appropriately then everything will work properly.

Closing Remarks

Summary

With Styles, Plugins and WHM settings, you have substantial control over the appearance of the cPanel and Webmail interfaces.

Coming Soon

Improved WHM Theming coming in version 12.

Questions?