CanIt Branding Guide

for Version 6.1.3

Roaring Penguin Software Inc.

9 October 2009

1 Introduction

CanIt has a facility called *branding* that lets you alter the look and feel of the web-based interface. You can brand CanIt to look like the rest of your web site, and make it fit seamlessly into the site. Generally, you can create your branding so that it survives CanIt upgrades.

1.1 Target Audience

CanIt's web interface is written in PHP. Before you tackle a branding project, you should:

- Be very familiar with HTML.
- Have considerable experience programming in PHP.

2 File Layout

All of CanIt's PHP files live under a single directory. On CanIt appliances, this directory is **`/var/www/canit`**. On Red Hat machines, it's likely to be **`/var/www/html/canit`**. In this manual, we will refer to the directory containing the files as ***`web_root`***.

The files are laid out as follows:

- **`.php`** files directly under ***`web_root`*** are small stubs that serve the actual CanIt web pages.
- ***`web_root`*****`/classes`** – object definitions for various CanIt objects such as users, domain rules, custom rules, etc.
- ***`web_root`*****`/images`** – static GIF, PNG and JPEG image files.
- ***`web_root`*****`/js`** – various JavaScript files.
- ***`web_root`*****`/manual`** – online HTML versions of the manuals.
- ***`web_root`*****`/pages`** – the actual implementation of CanIt web pages.
- ***`web_root`*****`/site`** – a directory for placing site-specific configuration files.
- ***`web_root`*****`/themes`** – a directory of *theme* files. A theme specifies the branding (or "look-and-feel") of CanIt.
- ***`web_root`*****`/themes/langs`** – language files for translating the CanIt web interface into another language.

As you customize the look of CanIt, you'll work mostly in the **`site`** and **`themes`** subdirectories of ***`web_root`***. But you may occasionally need to look at files in the other directories.

3 Configuration Files

When CanIt starts up, it reads the following configuration files, in order:

1. ***`web_root`*****`/config.php`**

2. ***web_root*/config-pro.php**
3. ***web_root*/config-domain-pro.php** (CanIt-Domain-PRO only.)
4. ***web_root*/config-appliance.php** (CanIt appliances only.)
5. ***web_root*/site/config.php** (if it exists.)
6. ***web_root*/site/*hostname*/config.php** where *hostname* is the server name sent by the Web browser.

CanIt then loads the appropriate language file and calls **post_lang_init()** (described later).

Next, CanIt looks for additional configuration files as follows:

1. If the directory ***web_root*/config.d** exists, then CanIt reads each file in that directory whose name matches ***.php**. The files are read in the same sorted order as output by **ls**.
2. If the directory ***web_root*/site/config.d** exists, then CanIt reads all the ***.php** files in that directory in sorted order.

4 Page Rendering

All CanIt pages are rendered by instantiating a **Page** object (actually, a subclass of **Page**.) **Page** is defined in ***web_root*/classes/page.php**.

The **Page** constructor instantiates a **CanItTheme** object (or an object of a class derived from **CanItTheme**.) We will refer to the theme object as **$theme**. The theme controls the entire look-and-feel of the Web page. CanIt ships with three theme classes:

- The base **CanItTheme** class, defined in ***web_root*/themes/theme_base.php**.
- The **ModernTheme** class, defined in ***web_root*/themes/theme_modern.php**.
- The default **PostmodernTheme** class, defined in ***web_root*/themes/theme_postmodern.php**.

You should familiarize yourself with all of these theme classes.

The entry point for rendering is **Page::render**. This function performs authentication checks (if necessary) and then calls **do_render**, a method that must be supplied by derived classes.

When a page is rendered, the calling sequence is typically:

1. **$theme->boilerplate_top()** This method outputs the preamble, the logo, the menus, etc.
2. Page contents are rendered.
3. **$theme->boilerplate_bottom()** This method outputs the footer and HTML postamble.

5 Creating a Theme

To create a theme, you need to write a PHP class that inherits from one of the three theme classes shipped with CanIt.

DO NOT simply **copy** one of the existing theme files and edit it. If you do that, your theme is likely to break when CanIt is upgraded. Instead, **inherit** from an existing theme class and only override the behavior you want changed.

For example, suppose you name your theme class **`MyTheme`**. Here's how you would go about creating the files:

1. Create a file called ***`web_root`*`/site/themes/theme_mytheme.php`**. It would look something like this:

   ```
   <?php
   /* Pull in the base class */
   require_once("themes/theme_base.php");

   /* Define our theme */
   class MyTheme extends CanItTheme {
      /* Theme definitions go here */
   }
   ?>
   ```

2. To tell CanIt to use your theme rather than the default theme, create ***`web_root`*`/site/config.php`** and make it look like this:

   ```
   <?php
   global $Config;
   $Config['Theme:Class'] = "MyTheme";
   $Config['Theme:Include'] = "site/themes/theme_mytheme.php";
   ?>
   ```

   The two assignments in the **`$Config`** array cause CanIt to load ***`web_root`*`/site/themes/theme_mytheme.php`** and to use an instance of **`MyTheme`** as **`$theme`**.

6 The boilerplate_top Method

The **`boilerplate_top`** method is responsible for rendering the opening HTML of the page. In the base **`CanItTheme`** class, **`boilerplate_top`** works like this:

1. It calls **`$theme->pre_boilerplate_top_hook()`**, which by default does nothing. You can override **`pre_boilerplate_top_hook`** in your theme class if required.
2. It calls **`$theme->print_header`**, which is responsible for printing the opening HTML code. **`print_header`** is passed three arguments:
   - **title**, the page title.
   - **head_extra**, which is extra code to go in the **`<HEAD>`** section.
   - **body_extra**, which are extra attributes to go in the **`<BODY>`** tag.

print_header in turn:

(a) Calls **$theme->print_stylesheet()** which emits HTML code to pull in the CSS stylesheet.

(b) Calls **$theme->javascript_boilerplate()** which emits HTML code to pull in required JavaScript libraries.

(c) Calls **$theme->top_of_page_hook($title)** which by default does nothing, but can be overridden.

3. When **print_header** returns, **boilerplate_top** calls **$theme->print_images_and_welcome_info** with two arguments: The login name of the logged-in user (if any) and some extra stream-related information. This function prints the logo and the "Logged in as: username" message.

4. Next, an instance of the **Menu** object is created and its methods are called to draw the menus.

5. Next, **$theme->print_page_intro($title)** is called with the page title as its only argument. This prints the page title.

6. Finally, **$theme->do_top_help()** is called to draw the "Online Documentation" and "Show Help" box.

The **boilerplate_top** function in **ModernTheme** and **PostmodernTheme** has been overridden; its flow is:

- Call **$theme->pre_boilerplate_top_hook**
- Call **$theme->print_header**
- Call **$theme->print_images_and_welcome_info**
- Call **$theme->draw_menus**
- Call **$theme->draw_content_area_start**
- Call **$theme->print_page_intro**
- Call **$theme->do_top_help**

See the files ***web_root*/themes/theme_modern.php** and ***web_root*/themes/theme_postmodern.php** for details.

7 The boilerplate_bottom Method

The **boilerplate_bottom** method is responsible for rendering the closing HTML of the page. It should balance any opening tags created by **boilerplate_top** with corresponding closing tags.

The standard **boilerplate_bottom** method closes out the content area and prints a copyright notice. Note that you are not allowed to remove the Roaring Penguin copyright notice, even if you re-brand CanIt.

See the files ***web_root*/themes/theme_base.php** and ***web_root*/themes/theme_modern.php** for examples of **boilerplate_bottom**.

8 Theme Methods

The **CanItTheme** class defines many methods which can usefully be overridden. The following sections are a summary of some of the theme methods. Note that built-in themes may define additional helper methods; before creating your own theme, you should study and understand the standard themes.

8.1 javascript_boilerplate

Arguments None.

Return Value Ignored.

Description Prints HTML code to load required JavaScript libraries. If you override this method, be sure to upcall to the **parent::javascript_boilerplate** method to pull in standard CanIt JavaScript libraries.

8.2 print_header

Arguments

- **$title**: The title of the page.
- **$head_extra**: Extra text (typically JavaScript) to add in the header section.
- **$body_extra**: Extra attrbutes to add in the **<BODY>** tag.

Return Value Ignored.

Description Print the HTML header code (starting with the **<HTML>** tag) and continuing to the **<BODY>** tag. Calls **top_of_page_hook** right after the **<BODY>** tag.

8.3 top_of_page_hook

Arguments None.

Return Value Ignored.

Description The method is called right after the **<BODY>** tag has been emitted. The base method does nothing, but derived classes may override the method.

8.4 print_stylesheet

Arguments None.

Return Value Ignored.

Description Prints HTML code to pull in the CSS stylesheet. Override this method if you create your own stylesheet.

8.5 print_images_and_welcome_info

Arguments

- **`$user`**: The user-ID of the logged-on user, or the empty string if no-one is logged on yet.
- **`$extra_stream_info`**: Extra stream-related information to display. (For example, "Viewing stream foo")

Return Value Ignored.

Description This method is called after **`print_header`**. It emits HTML code to display the logo and begin the actual page layout.

9 Content Printing Methods

The following methods are called at various times to print pieces of HTML content. You can override them to change the appearance of the content. (In many cases, you need only change CSS files, but sometimes you need more control than is possible using CSS.)

9.1 Quick-Link Related Methods

9.1.1 begin_quick_links

Arguments None.

Return Value Ignored.

Description Called when CanIt wants to render the "Quick Links" section of the page.

9.1.2 print_quick_link

Arguments

- **`$url`**: The URL of the quick link.
- **`$text`**: The text of the quick link.
- **`$active`**: A flag set to true if the current page is **`$url`** or false otherwise.

Return Value Ignored.

Description Emits HTML code to print a "Quick Link".

9.1.3 print_quick_link_separator

Arguments None.

Return Value Ignored.

Description Called in between each pair of "Quick Links" emitted by CanIt.

9.1.4 end_quick_links

Arguments None.

Return Value Ignored.

Description Called when CanIt has finished rendering the "Quick Links" section of the page.

9.2 Menu-Related Methods

9.2.1 begin_main_menu

Arguments None.

Return Value Ignored.

Description Called when CanIt begins rendering the top-level menu.

9.2.2 print_menu_entry

Arguments

- **`$url`**: The URL of the menu item.
- **`$text`**: The text of the menu item.
- **`$active`**: A flag set to true if the current page is **`$url`** or false otherwise.
- **`$title`**: HTML text suitable for use as the "title" attribute in an **`<A>`** tag.

Return Value Ignored.

Description Called for each main menu entry. This method prints the menu entry.

9.2.3 print_menu_entry_separator

Arguments None.

Return Value Ignored.

Description Called between main menu entries. Responsible for printing a separator between the entries.

9.2.4 end_main_menu

Arguments None.

Return Value Ignored.

Description Called when CanIt has finished rendering the top-level menu.

9.2.5 begin_secondary_menu

Arguments None.

Return Value Ignored.

Description Called when CanIt begins rendering the second-level menu.

9.2.6 print_menu2_entry

Arguments

- **`$url`**: The URL of the menu item.
- **`$text`**: The text of the menu item.
- **`$active`**: A flag set to true if the current page is **`$url`** or false otherwise.
- **`$title`**: HTML text suitable for use as the "title" attribute in an **<A>** tag.

Return Value Ignored.

Description Called for each second-level menu entry. This method prints the menu entry.

9.2.7 print_menu2_entry_separator

Arguments None.

Return Value Ignored.

Description Called between second-level menu entries. Responsible for printing a separator between the entries.

9.2.8 end_secondary_menu

Arguments None.

Return Value Ignored.

Description Called when CanIt has finished rendering the second-level menu.

9.2.9 begin_menu3

Arguments None.

Return Value Ignored.

Description Called when CanIt begins rendering a third-level menu.

9.2.10 print_menu3_entry

Arguments

- **`$url`**: The URL of the menu item.
- **`$text`**: The text of the menu item.
- **`$active`**: A flag set to true if the current page is **`$url`** or false otherwise.
- **`$title`**: HTML text suitable for use as the "title" attribute in an **<A>** tag.

Return Value Ignored.

Description Called for each third-level menu entry. This method prints the menu entry.

9.2.11 print_menu3_entry_separator

Arguments None.

Return Value Ignored.

Description Called between third-level menu entries. Responsible for printing a separator between the entries.

9.2.12 end_menu3

Arguments None.

Return Value Ignored.

Description Called when CanIt has finished rendering the third-level menu.

9.3 Table Drawing Methods

Many CanIt web pages contain tabular data. CanIt invokes theme functions to draw its tables. In most cases, you can customize table appearance purely by changing the CSS. However, for additional control, you can override the table drawing methods discussed in this section.

9.3.1 begin_standard_table

Arguments

- **`$extra_attrs`**: Extra attributes to include in the **`<TABLE>`** tag.

Return Value Ignored.

Description Emits HTML code to begin drawing a table. It should emit a **`<TABLE>`** tag *and* the inital **`<TR>`** tag to start the first row.

9.3.2 begin_standard_table_and_headers

Arguments

- **`$headings`**: An array of table headings (each element is a string.)
- **`$extra_attrs`**: Extra attributes to include in the **`<TABLE>`** tag.
- **`$cell_extra_attrs`**: Extra attributes to include in each **`<TH>`** tag.

Return Value Ignored.

Description This method calls **`begin_standard_table`** and then calls **`standard_table_header_cell`** for each element of **`$headings`**.

9.3.3 standard_table_header_cell

Arguments

- **`$text`**: The text to put in the TH cell.
- **`$extra_attrs`**: Extra attributes to include in the **`<TH>`** tag.

Return Value Ignored.

Description Emits a **`<TH>`** tag, the table heading, and a **`</TH>`** tag.

9.3.4 centered_table_header_cell

Arguments

- **`$text`**: The text to put in the TH cell.
- **`$extra_attrs`**: Extra attributes to include in the **`<TH>`** tag.

Return Value Ignored.

Description The same as **`standard_table_header_cell`** except the **`<TH>`** tag has an **`align="center"`** attribute.

9.3.5 begin_another_header_row

Arguments

- **`$extra_attrs`**: Extra attributes to include in the **`<TR>`** tag.

Return Value Ignored.

Description Closes a row by emitting a **`</TR>`** tag; opens a new row by emitting a **`<TR>`** tag.

9.3.6 begin_standard_table_body

Arguments None.

Return Value Ignored.

Description Closes the header row by emitting a **`</TR>`** tag.

9.3.7 begin_standard_body_row

Arguments

- **`$extra_attrs`**: Extra attributes to add to the **`<TR>`** tag.

Return Value Ignored.

Description Begins a body row by emitting a **`<TR>`** tag. The theme also tracks whether this is an odd-numbered or even-numbered row so that rows can be colored alternately using CSS classes.

9.3.8 standard_table_body_cell

Arguments

- **`$text`**: Text to put in the table cell.
- **`$extra_attrs`**: Extra attributes to add to the **`<TD>`** tag.

Return Value Ignored.

Description Emits a table cell, which is essentially **`$text`** enclosed in **`<TD>`** and **`</TD>`** tags.

9.3.9 centered_table_body_cell

Arguments

- **`$text`**: Text to put in the table cell.
- **`$extra_attrs`**: Extra attributes to add to the **`<TD>`** tag.

Return Value Ignored.

Description The same as **`standard_table_body_cell`** except that the **`<TD>`** tag has a **`style="text-align: center"`** attribute added.

9.3.10 standard_table_body_cell_nowrap

Arguments

- **`$text`**: Text to put in the table cell.
- **`$extra_attrs`**: Extra attributes to add to the **`<TD>`** tag.

Return Value Ignored.

Description The same as **`standard_table_body_cell`** except that the **`<TD>`** tag has a **`style="whitespace: nowrap"`** attribute added.

9.3.11 end_standard_body_row

Arguments None.

Return Value Ignored.

Description Closes the body row by emitting a **`</TR>`** tag.

9.3.12 end_standard_table

Arguments None.

Return Value Ignored.

Description Closes the table by emitting a **`</TABLE>`** tag.

9.4 Miscellaneous Methods

9.4.1 menu_location

Arguments None.

Return Value Ignored.

Description Prints the location of the menus. Depending on the theme, this is likely to be "above" or "to the left".

9.4.2 guide_links

Arguments None.

Return Value Ignored.

Description Prints links to the online versions of the User's Guide and possibly the Administration Guide and Installation Guide.

10 Global Variables

10.1 $Config

The **$Config** global array contains many configuration settings. Please see the various **config*.php** files in ***web_root*** for details. User-changeable configuration items are commented in those files. You can change the configuration settings in ***web_root*/site/config.php**.

In your theme, you should normally *never* read from the **$Config** array directory. Instead, call the **conf()** function (described in Section 11.5).

10.2 $CanItPrefs

The **$CanItPrefs** global array contains user preferences; it is populated from the database whenever a page is rendered. See ***web_root*config.php** for details.

In your theme, you should normally *never* read from the **$CanItPrefs** array directory. Instead, call the **getpref()** function (described in Section 11.6).

11 Useful Global Functions

The following global functions (defined in **classes/utils.php**) may be useful:

11.1 product_name

Arguments None.

Return Value The name of the product. This is the internal product name and is always one of **CanIt-PRO** or **CanIt-Domain-PRO**.

Description Returns the product name.

11.2 display_product_name

Arguments None.

Return Value The name of the product for display purposes. This is the name that end-users see and can be overridden if you like.

Description Returns the display product name.

11.3 is_canit_pro

Arguments None.

Return Value True if the product is **`CanIt-PRO`**; false otherwise.

11.4 product_has_realms

Arguments None.

Return Value True if the product has realms (that is, if the product is **`CanIt-Domain-PRO`**); false otherwise.

11.5 conf

Arguments

- **`$item`**: The name of a configuration item

Return Value The value of the global **`$Config[$item]`** variable or the empty string if no such configuration item exists.

11.6 getpref

Arguments

- **`$name`**: The name of a preference item

Return Value The value of the global **`$CanItPrefs[$name]`** variable or the empty string if no such preference exists.

11.7 is_root

Arguments None

Return Value True if the logged-on user has "root" privileges. In CanIt-Domain-PRO, this includes root privileges in any realm.

11.8 is_super_root

Arguments None

Return Value True if the logged-on user has "root" privileges. In CanIt-Domain-PRO, this includes only root privileges in the "base" realm.

11.9 is_demo

Arguments None

Return Value True if the logged-on user is the "demo" user.

11.10 is_guest

Arguments None

Return Value True if the logged-on user has not authenticated. This can happen (for example) if you permit unauthenticated voting.

11.11 should_hide_sensitive_info

Arguments None

Return Value True if the logged-on user is either "demo" or an unauthenticated user. You can use this to test whether or not to obfuscate sensitive information.

11.12 get_uid

Arguments None

Return Value The login name of the currently logged-in user, or the empty string if no user is logged in.