Summer Cart 5.0
Design Customization Guide

Introduction

Summer Cart can easily be customized to match your desired look and feel and this guide will show you how to do this right. First, we will cover the basics and explain how layouts, boxes and skins work, and show you the options you have to customize a Summer Cart store through the Admin Panel – using the powerful Design Editor that is now integrated into the software. For the more advanced scenarios you would want to create a custom skin. We will cover everything you need to know about using and extending existing skins, as well as making your own. We also explain our code style and conventions, and advise you to follow them so your code stays in shape – and that will be important when you have to maintain and update your skins. Finally, our How Do I section provides quick answers to common questions.

Prerequisites

Before you begin, make sure you have the following prerequisites in place:

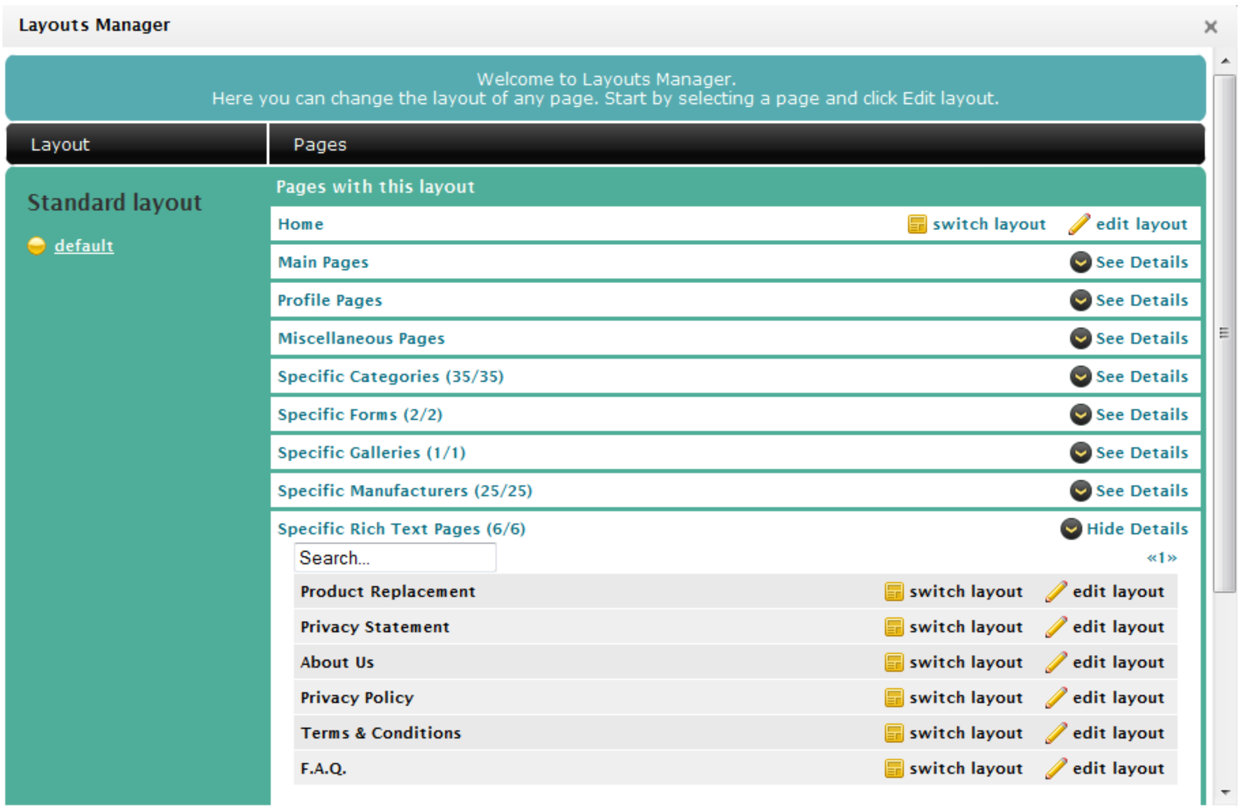
- A stand-alone Summer Cart 5.0 installation;
- Access to the store's Admin Panel;
- Access to the store files (e.g. through FTP);
- Certain knowledge in HTML and CSS.

Contents

The Basics

Layouts

Layouts define the arrangement of elements on a page. Each page in Summer Cart has an associated layout, and multiple pages can use the same layout. By default, there is a single layout called “Default layout” and all pages are set to use it. Through the Admin Panel you can create more layouts and associate pages to them. In Summer Cart 5.0 it is now possible that specific items, such as categories, manufacturers or products use their own layout.


Layouts Manager

Welcome to Layouts Manager.
Here you can change the layout of any page. Start by selecting a page and click Edit layout.

Layout	Pages
Standard layout default	Pages with this layout

Home	switch layout	edit layout
Main Pages		See Details
Profile Pages		See Details
Miscellaneous Pages		See Details
Specific Categories (35/35)		See Details
Specific Forms (2/2)		See Details
Specific Galleries (1/1)		See Details
Specific Manufacturers (25/25)		See Details
Specific Rich Text Pages (6/6)		Hide Details

Search... «1»

Product Replacement	switch layout	edit layout
Privacy Statement	switch layout	edit layout
About Us	switch layout	edit layout
Privacy Policy	switch layout	edit layout
Terms & Conditions	switch layout	edit layout
F.A.Q.	switch layout	edit layout

Fig. 1.1 Standard layout and associated pages

The structure of a page in Summer Cart is illustrated on fig. 1.2. Pages have a header defined in the skin. Furthermore some skins have a number of built-in sub headers and footers, and through the Design Editor you can select which sub header and footer should be used in a layout. Then pages have a number of horizontal and vertical columns. Horizontal columns are built into the skin. A typical example is a main menu in the top area of the site. Vertical columns can be set through the Design Editor. A layout can have up to 4 columns, with one holding the page content (“wide” column) and up to 3 columns for boxes (“thin” columns). These are called Left, Right and Extra Right columns. You have several options on the columns ordering. Figure 1.3 illustrates a layout with just 2 columns: thin left and wide middle.

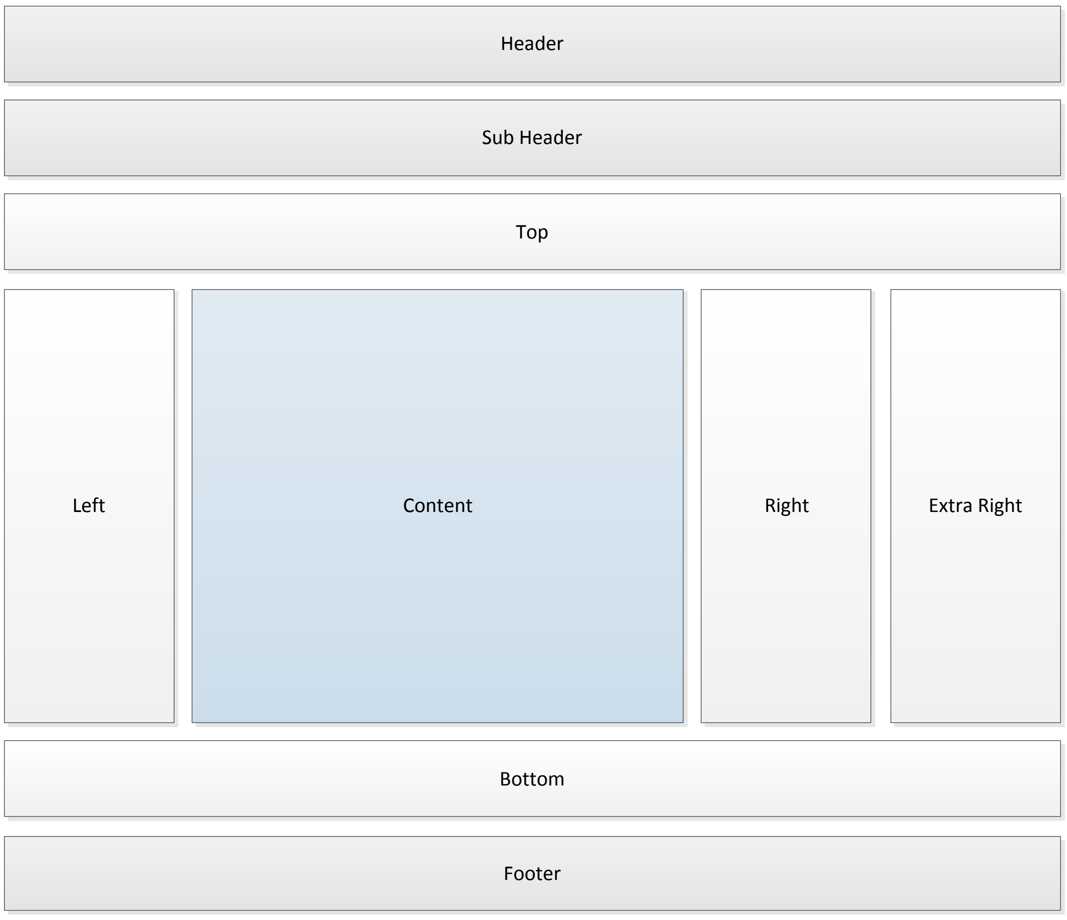
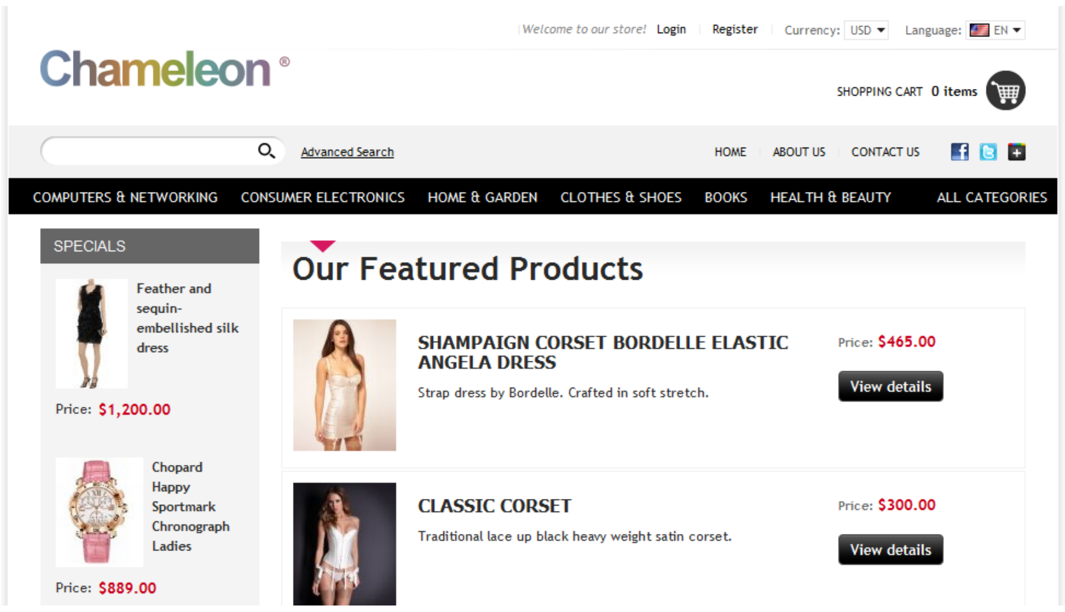


Fig. 1.2 Page structure



Feather and sequin-embellished silk dress

Price: $1,200.00

Chopard Happy Sportmark Chronograph Ladies

Price: $889.00

Our Featured Products

SHAMPAIGN CORSET BORDELLE ELASTIC ANGELA DRESS

Strap dress by Bordelle. Crafted in soft stretch.

Price: $465.00

CLASSIC CORSET

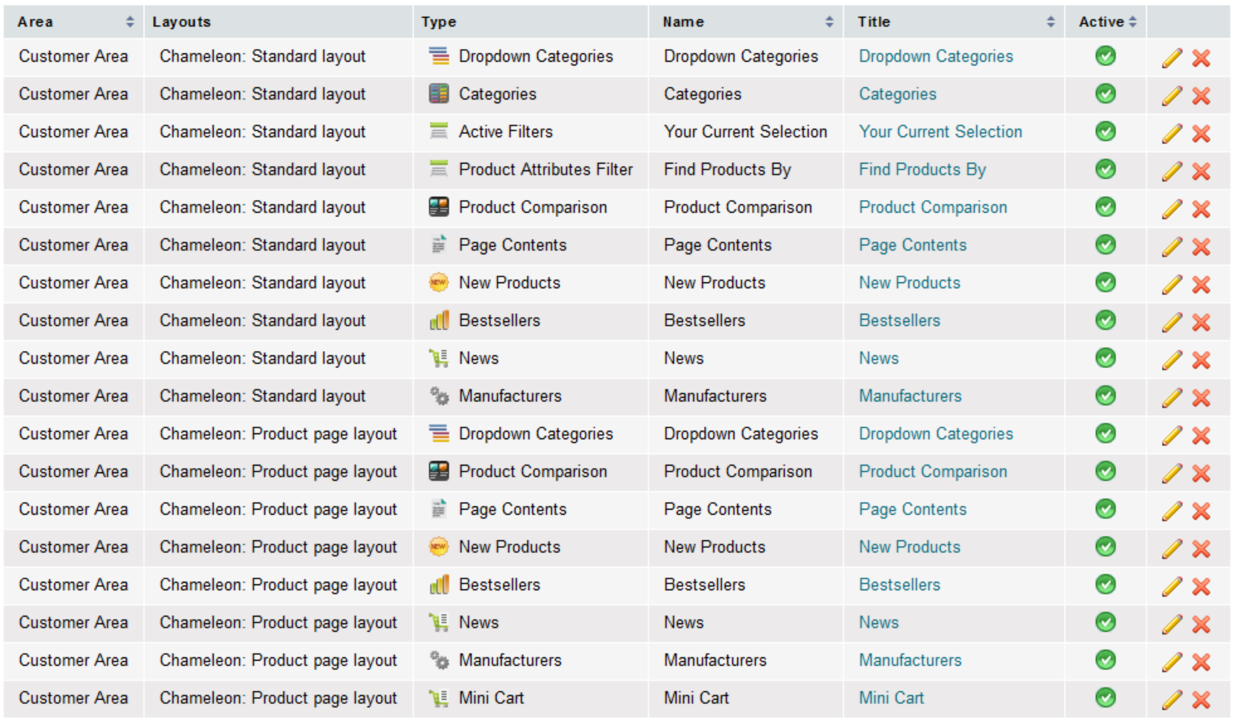
Traditional lace up black heavy weight satin corset.

Price: $300.00

Fig. 1.3 Layout with left thin column and middle wide column

Boxes

In the Boxes page in the Admin Panel you will see a list of all the boxes that are currently available in the system. Most of them will have one or more options you can customize. For example, the Bestsellers box has the options of what number of bestsellers to show, and what period of time (e.g. last 30 days) to use when calculating total sales for a product. You can also create your own custom HTML boxes and use them in layouts. Note that each box has its own ID, and if you create a custom skin you can set CSS rules for that specific box using its ID (more on that in the chapter “Skins”).



Area	Layouts	Type	Name	Title	Active	
Customer Area	Chameleon: Standard layout	Dropdown Categories	Dropdown Categories	Dropdown Categories	✓	
Customer Area	Chameleon: Standard layout	Categories	Categories	Categories	✓	
Customer Area	Chameleon: Standard layout	Active Filters	Your Current Selection	Your Current Selection	✓	
Customer Area	Chameleon: Standard layout	Product Attributes Filter	Find Products By	Find Products By	✓	
Customer Area	Chameleon: Standard layout	Product Comparison	Product Comparison	Product Comparison	✓	
Customer Area	Chameleon: Standard layout	Page Contents	Page Contents	Page Contents	✓	
Customer Area	Chameleon: Standard layout	New Products	New Products	New Products	✓	
Customer Area	Chameleon: Standard layout	Bestsellers	Bestsellers	Bestsellers	✓	
Customer Area	Chameleon: Standard layout	News	News	News	✓	
Customer Area	Chameleon: Standard layout	Manufacturers	Manufacturers	Manufacturers	✓	
Customer Area	Chameleon: Product page layout	Dropdown Categories	Dropdown Categories	Dropdown Categories	✓	
Customer Area	Chameleon: Product page layout	Product Comparison	Product Comparison	Product Comparison	✓	
Customer Area	Chameleon: Product page layout	Page Contents	Page Contents	Page Contents	✓	
Customer Area	Chameleon: Product page layout	New Products	New Products	New Products	✓	
Customer Area	Chameleon: Product page layout	Bestsellers	Bestsellers	Bestsellers	✓	
Customer Area	Chameleon: Product page layout	News	News	News	✓	
Customer Area	Chameleon: Product page layout	Manufacturers	Manufacturers	Manufacturers	✓	
Customer Area	Chameleon: Product page layout	Mini Cart	Mini Cart	Mini Cart	✓	

Fig. 1.4 List of built-in boxes

For maximum flexibility with your custom HTML boxes, you can uncheck the “Show Title” checkbox, in which case the only thing Summer Cart will add to your custom HTML is put it into a div with the ID of the box, so you can target your box in your CSS rules.

At any time you can easily remove a box from all layouts that include it by unchecking the box’s Active checkbox. This way the box will disappear from all layouts and the Customer Area, but as soon as you activate the box again it will reappear in its original place.

For a complete list of built-in boxes (with screenshots) please see the References section.

Rich Text Pages

Rich Text Pages are static HTML pages you can add through the Admin Panel. As with all pages, these can have different layouts. To apply CSS you will have to either embed the CSS to the page, or create a skin.

Dynamic Pages

Dynamic pages are ones built into the system, with their content generated dynamically. These are for example the Home page, the Products page, News page and so on. Dynamic pages use HTML templates, and PHP scripts parse these templates and generate the actual HTML code. From the Admin Panel you can add static HTML and CSS code to their tops and bottoms, and change their layout. However, to fully customize their HTML or CSS (not just add code to tops and bottoms) you will have to create a skin.

Forms

You use forms when you want to collect feedback from the customers. The most ubiquitous example is a Contact Us form. A form has a title and description, and a number of input controls such as your regular textboxes, multiline textboxes, checkboxes, radio buttons and drop-downs. You add these through the Admin Panel, without having to write any HTML code. In addition to these, you can use a number of built-in input controls with specific function, such as email and number textboxes, countries and states drop-downs, file upload control and a custom HTML control. The input from the user is sent through email to the email address set in the form settings.

Plugins

Plugins are packages of extra features that can be added to a Summer Cart store when the user installs a plugin through the Admin Panel. A plugin may contain additional templates, and overwrite existing templates. A plugin may also contain additional modules that have their own templates and CSS files.

Menus

In Summer Cart you are not restricted on the number of menus you can have on a site. Typically you would have a Main Menu displayed in the header of the site, with links to pages like Home, About Us, Contact Us, etc. Another place often used for a menu is the footer of the site with links to F.A.Q. page, Privacy Policy, Terms & Conditions, and others. In Summer Cart there are 4 possible menu locations:

Menu Location	Description
Top	Menu is displayed in the site header.
Footer	Menu is displayed in the site footer.
Fast Menu	Menu is displayed in the site header, typically on the very top of the site, separate from the Top menu.
Box	Menu is displayed in a box. When you add a new menu with Box location, a new box will be added to layout elements, one with the name of your new menu.

In a menu you can add a number of menu items. Each menu item is a link to either a Rich Text Page, Dynamic Page, URL, Form, or a Gallery. Menu items can also have icons that you upload.

Skins

Skins contain HTML templates for pages and boxes, CSS files, images, and script files. At any time you can change the look and feel of a Summer Cart store by changing its skin. Note that this will not affect layouts, which work independently from skins. For example, if you put a Search box in the Top area of a layout, even if you change the skin the Search box will remain in the Top area, although it may look completely different.

From the Admin Panel you can see a list of all the skins installed on the site. Some of the skins have settings that the administrator can customize by clicking the Settings link next to the skin's name. Skins are covered in great detail later in this guide.

Design Editor

Summer Cart 5.0 ships with a powerful and easy to use design editor that allows you to customize the look and feel of the store in minutes. This is a list of things you can do with the design editor:

- Create and modify layouts;
- Assign pages to layouts;
- Specify vertical columns of a layout and their widths;
- Change page header and footer;
- Change backgrounds of pages, headers and footers, and individual columns;
- Change appearance of buttons;
- Change appearance of text elements;
- Arrange and customize boxes.

How it works

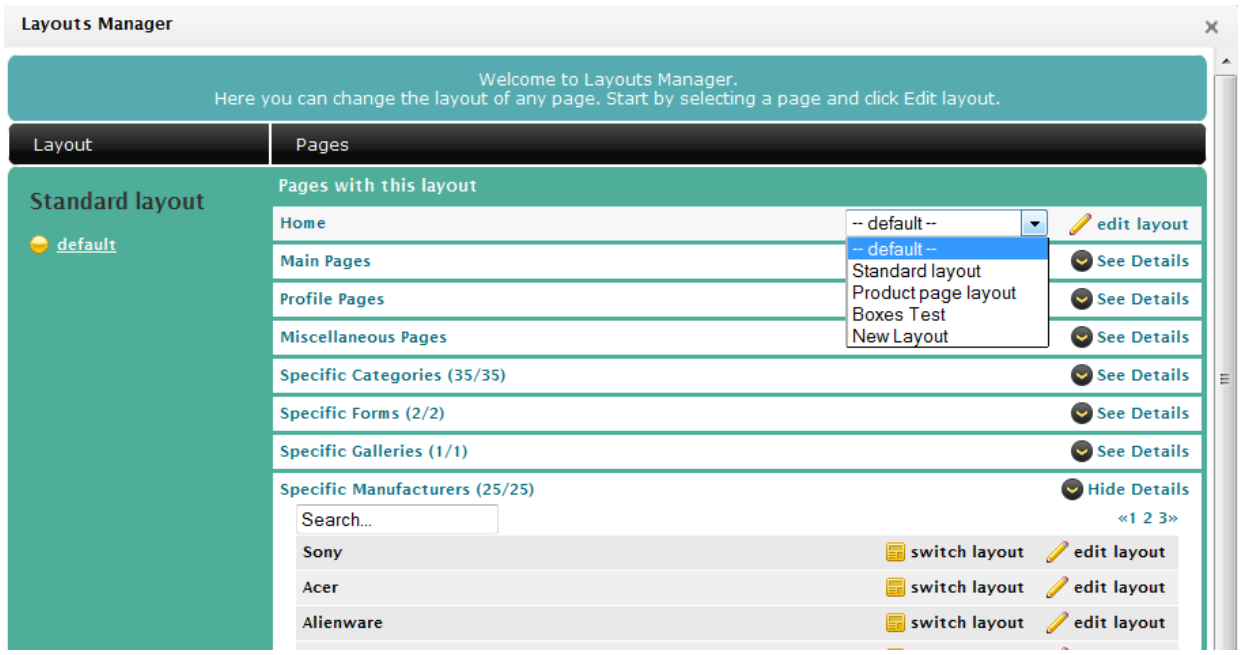
The Designer Editor uses some special tags in the skin's CSS file. Through these tags you can specify that a certain CSS line is related to a certain button or text element, or page style. Summer Cart recognizes these special tags and shows related options in the Design Editor. When you change these options in the Design Editor and save the layout, Summer Cart will generate a new CSS file and replace the original CSS line with a generated one based on your selected options. The page is then made to reference the new CSS file.

For example, the <mi:pageStyle> tag is used to specify that a certain CSS line defines the page style.

```
html, body {margin: 0; padding: 0; color: #292929; font-family: "Trebuchet
MS", Tahoma, Arial, Helvetica, sans-serif; font-size: 12px;}/* <mi:pageStyle
/> */
body { background: #fff url(../images/backgrounds/homepage.jpg) 50% 0% no-
repeat;}/* <mi:pageStyle /> */
```

Note that the <mi:pageStyle> tag appears in comments, so it will be ignored by any editor or browser. It is recognized only by Summer Cart. Also note that in this example the tag appears on two separate lines. Now if through the Design Editor you change the font size of the page, it will be saved in line 1, while if you change the background of the page, it will be saved in line 2. Most tags will appear on a single line.

Layouts Manager



Layouts Manager

Welcome to Layouts Manager.
Here you can change the layout of any page. Start by selecting a page and click Edit layout.

Layout	Pages
Standard layout	Pages with this layout
default	

Home	-- default --	edit layout
Main Pages		See Details
Profile Pages		See Details
Miscellaneous Pages		See Details
Specific Categories (35/35)		See Details
Specific Forms (2/2)		See Details
Specific Galleries (1/1)		See Details
Specific Manufacturers (25/25)		Hide Details

Sony	switch layout	edit layout
Acer	switch layout	edit layout
Alienware	switch layout	edit layout

Fig. 2.1 Layouts manager

In the Layouts Manager you associate pages with layouts. Click on “switch layout” to switch the page to a different (or new) layout. Click on “edit layout” to edit the page’s layout.

Note that not only specific pages can have different layouts, but even specific items (e.g. categories, products) can have their own layout. Summer Cart allows you to target specific items of the following types:

- Categories,
- Forms,
- Galleries,
- Manufacturers,
- News,
- Products,
- Rich Text Pages.

Columns

Layouts have horizontal and vertical columns. Horizontal columns are typically programmed in the skin (e.g. a top menu), while vertical columns are dynamic and can be specified in the Design Editor.

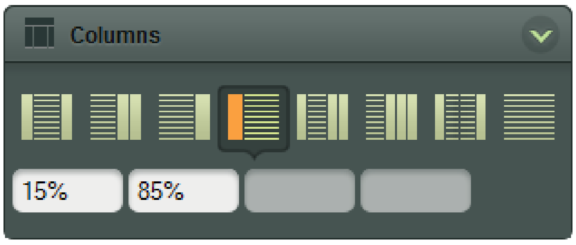


Fig 2.2 Columns

A layout may have between 2 and 4 vertical columns. We have two types of columns, called "thin" (for columns with boxes) and "wide" (for content). In the screenshot above we have a "thin-wide" layout, because it has two columns, a thin one and a wide one. For each column you can specify the column width. This arrangement will result in the following HTML code:

```
<div class="mainWrapper">
  <table id="main">
    <tbody>
      <tr>
        <td id="column-left" class="v-column thin">
        <td id="column-middle" class="v-column wide">
      </tr>
    </tbody>
  </table>
</div>
```

From left to right, the names of the columns are:

- column-left
- column-middle
- column-right
- column-extra-right

All vertical columns will have the "v-column" class, and additionally the "thin" or "wide" class depending on the type of the column.

Note that the widths you type in the editor will be directly written in the main.css file, so if you leave the width box blank, a default width will be used. You can use any values that you'd normally use in a CSS, e.g. "15%" or "120px", etc.

Page Style

Background

The Background tab allows you to change the page's background color and image, and you can use one of the many built-in seamless textures for background. For example, the settings from the screenshot above resulted in this generated CSS class:

```
body.layout-9 {
  background-color: #FFFFFF;
  background-image: url("/skins/skeleton/.../11.png");
  background-repeat: repeat;
}
```

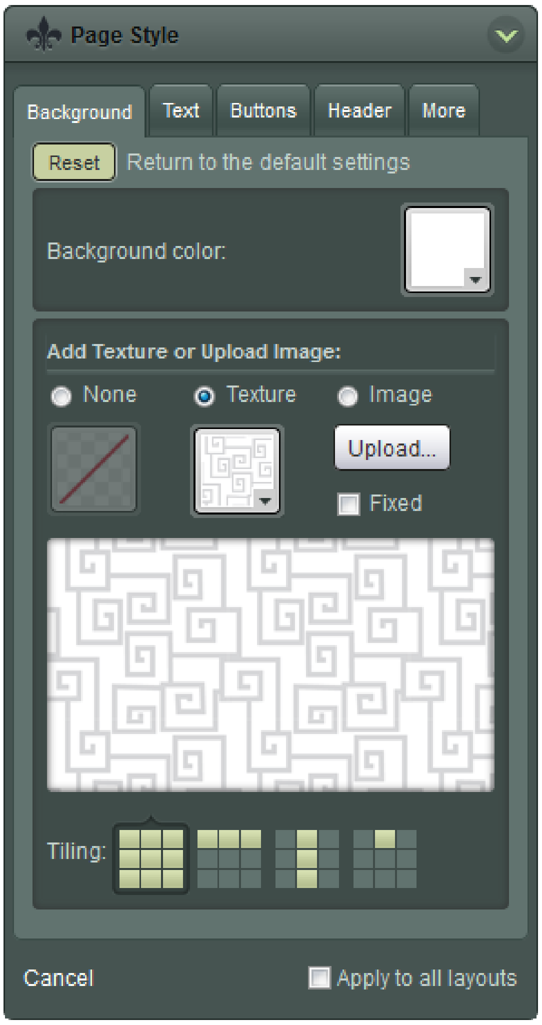


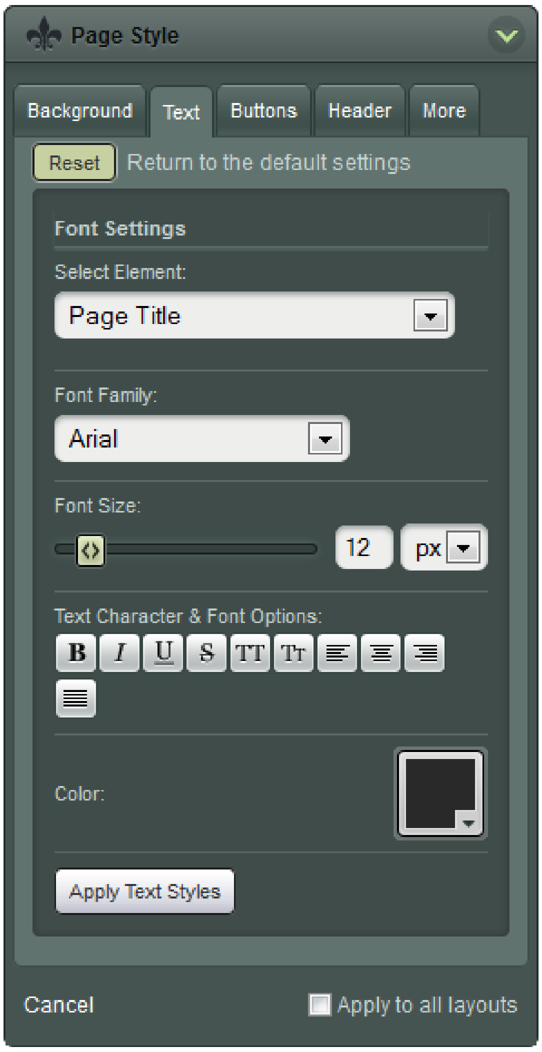
Fig. 2.3 Page Style, Background tab

Text

The Text tab allows you to change the appearance of text elements: font, font size, color and effects. This works through the <mi:textelement> tag. For example, the h1-title class has the following definition:

```
.h1-title h1 {padding:0px; float:left; display:inline-block; width:100%;
color:#292929; font-size:30px; line-height:30px; padding-bottom:43px;
margin:0px;} /* <mi:textelement name="Page Title" /> */
```

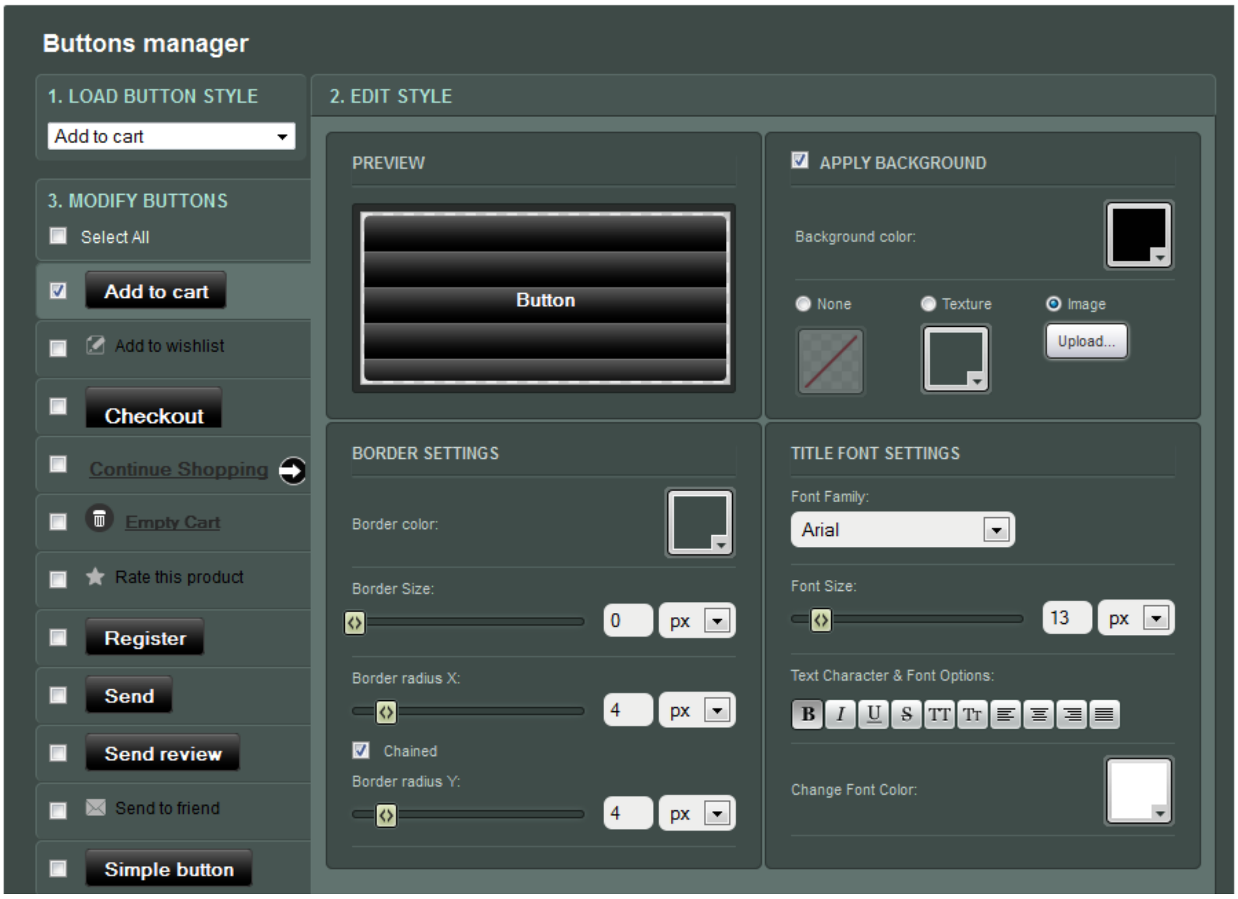
The comment at the end specifies that this line is associated with a <mi:textelement> (Text Element) with a name of "Page Title". Summer Cart will collect all these text elements from the CSS and display them in the "Select Element" drop-down, so the user can modify the values through the Design Editor. When you change these options and save the layout, Summer Cart will generate a new CSS file and replace the original CSS line with a generated one based on your selected options. The page is then made to reference the new CSS file.



Select Element: Page Title
Font Family: Arial
Font Size: 12 px
Text Character & Font Options:
Color:

Fig. 2.4 Page Style, Text tab

Buttons



Buttons manager

1. LOAD BUTTON STYLE
Add to cart

3. MODIFY BUTTONS
☐ Select All
☑ Add to cart
☐ Add to wishlist
☐ Checkout
☐ Continue Shopping
☐ Empty Cart
☐ Rate this product
☐ Register
☐ Send
☐ Send review
☐ Send to friend
☐ Simple button

2. EDIT STYLE

PREVIEW
Button

☑ APPLY BACKGROUND
Background color:
○ None ○ Texture ● Image
Upload...

BORDER SETTINGS
Border color:
Border Size: 0 px
Border radius X: 4 px
☑ Chained
Border radius Y: 4 px

TITLE FONT SETTINGS
Font Family: Arial
Font Size: 13 px
Text Character & Font Options:
Change Font Color:

Fig. 2.5 Page Style, Buttons tab, Buttons Manager

The Buttons tab allows you to change the appearance of buttons. As you can see on the screenshot above, you first have to select a button, then you edit the style of the button (border, background, font) and finally you select to which buttons your style should be applied to. By default only buttons that use the selected button style will be checked. This works through the <mi:button> tag:

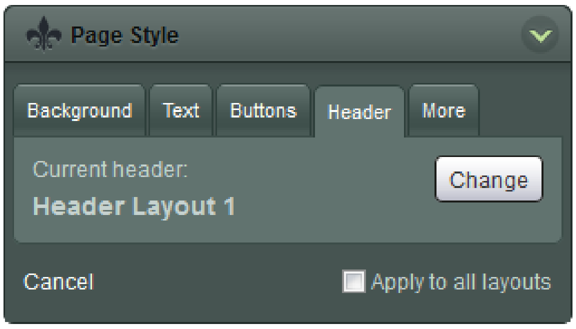
```
.button-wishlist a { color: #000; font-size: 12px; font-weight: normal;
padding: 0 0 0 22px; } /* <mi:button class="button-wishlist" /> */
```

For example, changes applied through the Design Editor to the “Add to cart” button will result in the following generated CSS code:

```
body.layout-6 .button-cart {
  background-color: #010101;
  background-image: url("/skins/chameleon/.../button.png");
  border-radius: 4px 4px 4px 4px;
}
body.layout-6 .button-cart a {
  color: #FFFFFF;
  font-size: 13px;
  font-weight: bold;
}
```

Header

The Header tab allows you to change the layout's sub header based on one of the built-in header styles. As explained in the Skins chapter, each skin may have multiple header styles, and one can be selected by the user through the Design Editor.



Current header:
Header Layout 1

Fig. 2.6 Page Style, Header tab

Click the Change button to switch to another sub header:

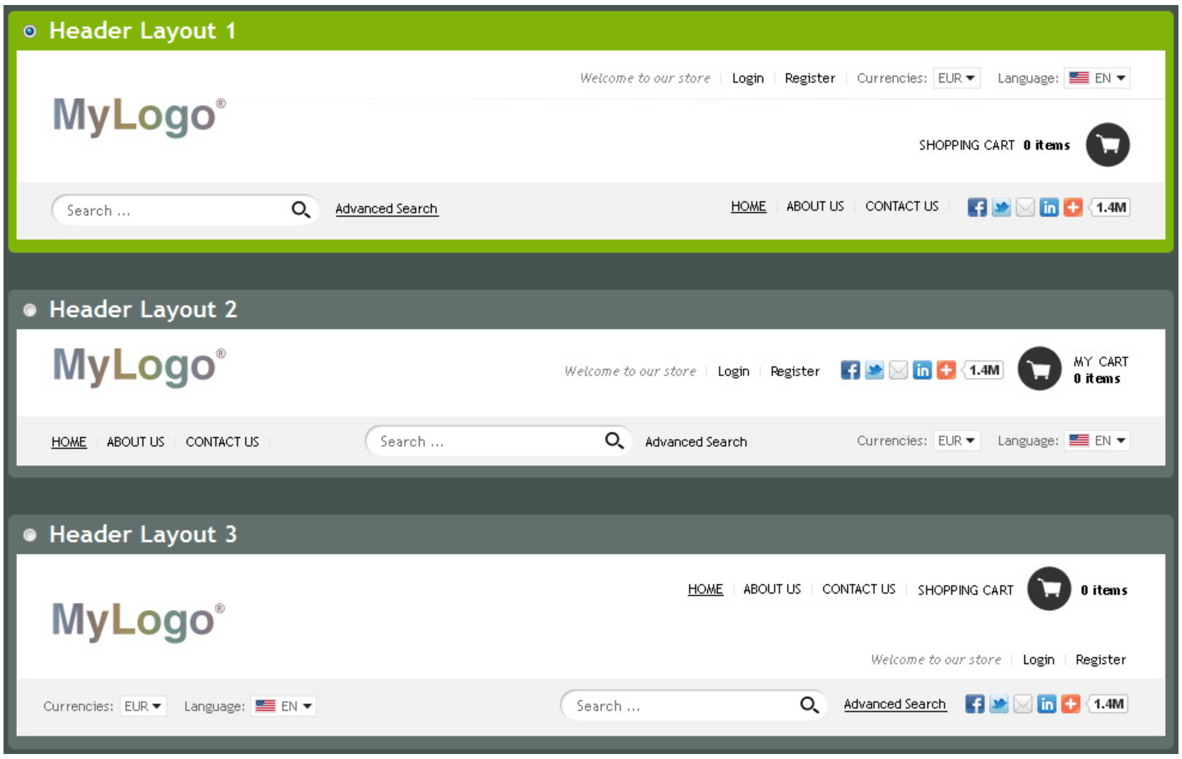


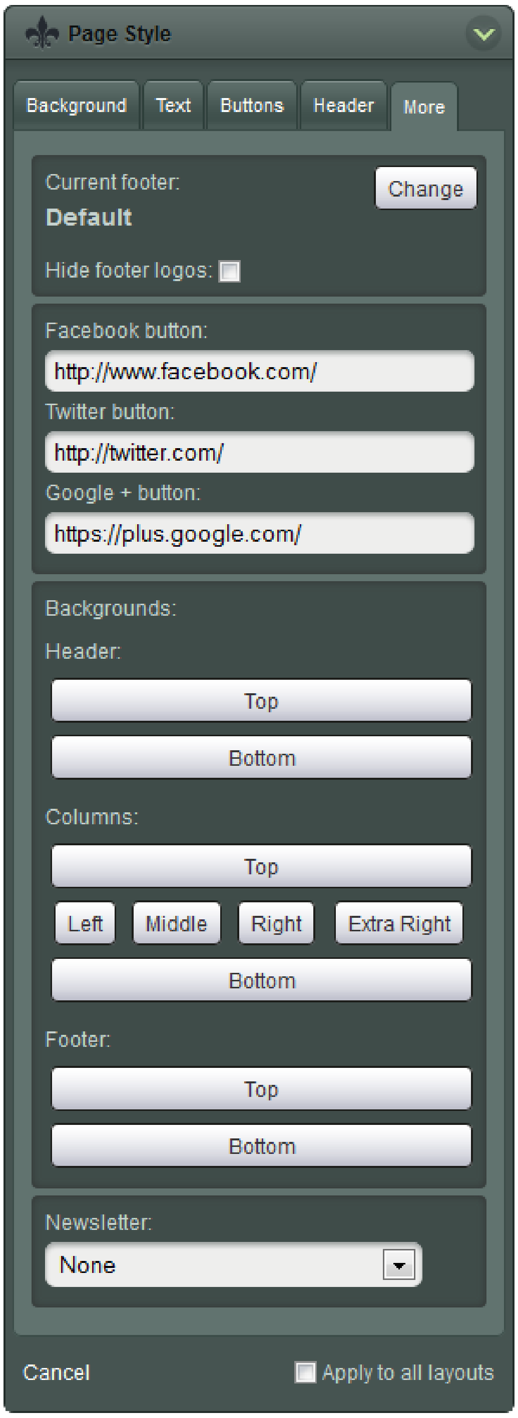
Fig. 2.7 Header options for the Chameleon skin

Note that if the skin doesn't have any sub headers, the Headers tab will not be visible.

More

The More tab provides several options:

- Change the footer of the layout;
- Change the URLs of the Facebook, Twitter and Google+ buttons;
- Change the appearance of the top and bottom header, all horizontal and vertical columns, and top and bottom footer.


Page Style

Background | Text | Buttons | Header | More

Current footer: **Default** Change

Hide footer logos: ☐

Facebook button: http://www.facebook.com/

Twitter button: http://twitter.com/

Google + button: https://plus.google.com/

Backgrounds:

Header: Top Bottom

Columns: Top Left Middle Right Extra Right Bottom

Footer: Top Bottom

Newsletter: None

Cancel ☐ Apply to all layouts

Fig. 2.8 Page Style, More tab

The Chameleon skin has the following header and footer definitions:

```
<div id="headerPart-top" class="subheader1 header">
  <div class="header-settings">...</div>
  <h2 id="logo">...</div>
  <div id="header-cart">...</div>
  <div id="headerPart-bottom" class="header-menu-search">...</div>
  <div class="clear"></div>
</div>
```

```
<div id="footer" class="clearfix">
  <div id="footerPart-top" class="footer-shop-info">...</div>
  <div id="footerPart-bottom" class="footerbg">...</div>
</div>
```

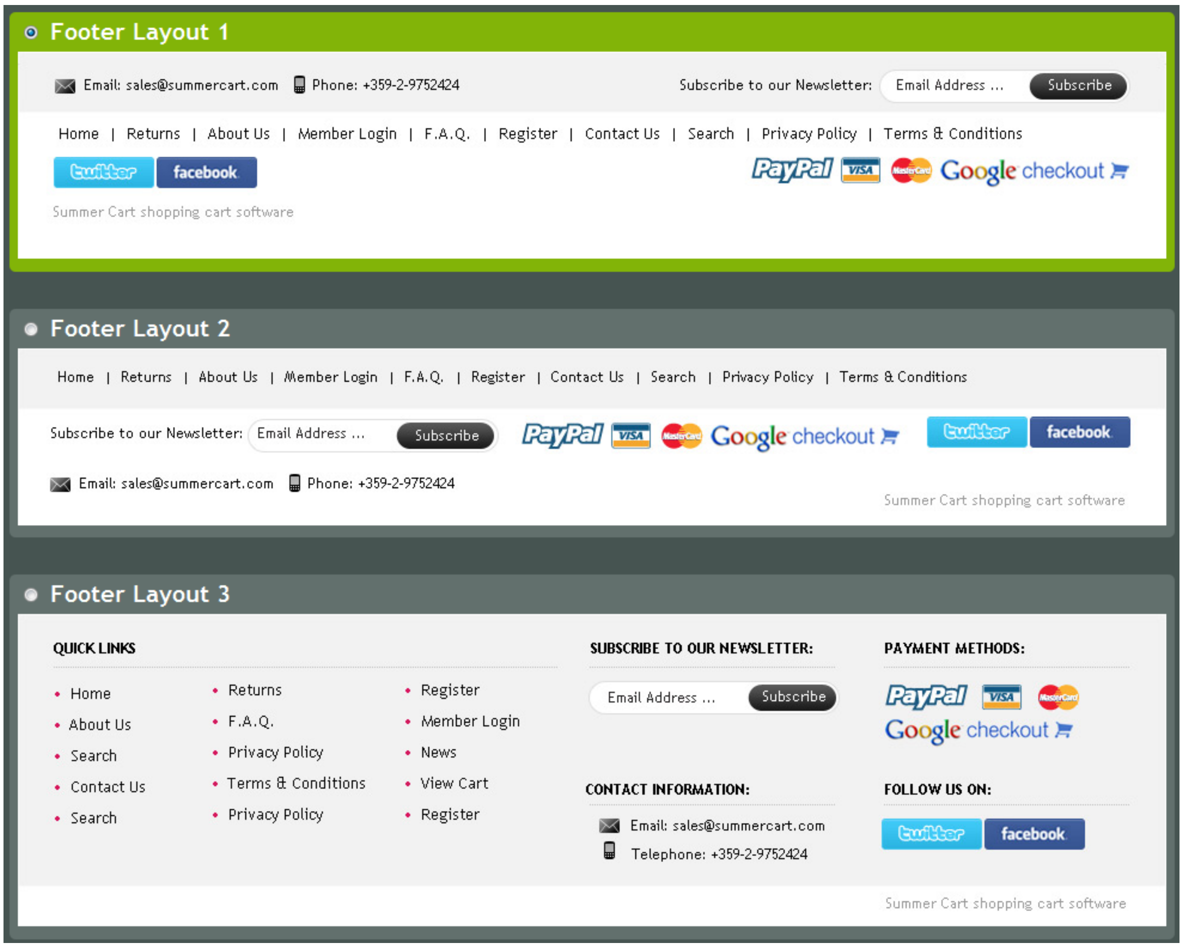
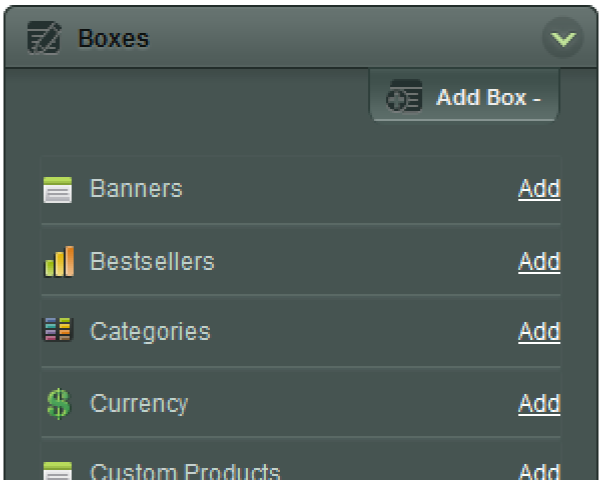
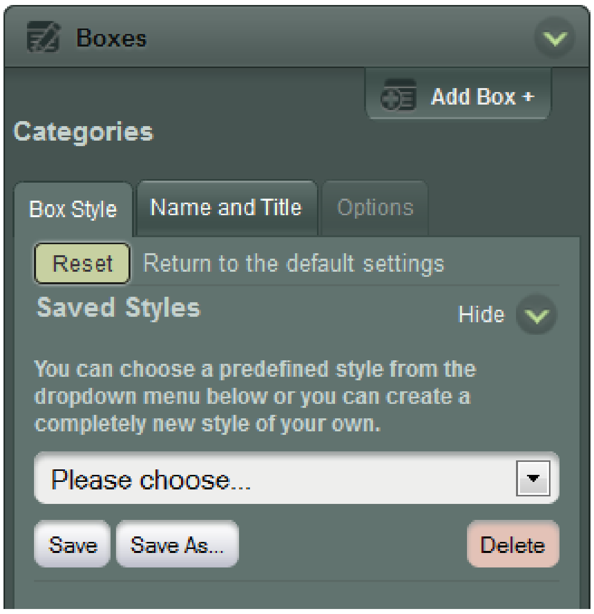

Footer Layout 1
Email: sales@summercart.com
Phone: +359-2-9752424
Subscribe to our Newsletter:
Email Address ...
Subscribe
Home | Returns | About Us | Member Login | F.A.Q. | Register | Contact Us | Search | Privacy Policy | Terms & Conditions
twitter
facebook
PayPal
VISA
Google checkout
Summer Cart shopping cart software
Footer Layout 2
Home | Returns | About Us | Member Login | F.A.Q. | Register | Contact Us | Search | Privacy Policy | Terms & Conditions
Subscribe to our Newsletter:
Email Address ...
Subscribe
PayPal
VISA
Google checkout
twitter
facebook
Email: sales@summercart.com
Phone: +359-2-9752424
Summer Cart shopping cart software
Footer Layout 3
QUICK LINKS
Home
About Us
Search
Contact Us
Search
Returns
F.A.Q.
Privacy Policy
Terms & Conditions
Privacy Policy
Register
Member Login
News
View Cart
Register
SUBSCRIBE TO OUR NEWSLETTER:
Email Address ...
Subscribe
PAYMENT METHODS:
PayPal
VISA
Google checkout
CONTACT INFORMATION:
Email: sales@summercart.com
Telephone: +359-2-9752424
FOLLOW US ON:
twitter
facebook
Summer Cart shopping cart software

Fig. 2.9 Footer options for the Chameleon skin

Boxes



You can add boxes to the layout and arrange them into columns by simple drag & drop. Once a box is positioned, click on the Edit button on top of the box to edit it. You can specify background options, fonts and font colors, box title in multiple languages, and other options that are specific to the box. (Built-in boxes will often have additional options.)



You can choose a predefined style from the dropdown menu below or you can create a completely new style of your own.

Often you will want multiple boxes to share the same options. To do this, apply the options to a single box, then save the style as illustrated on the screenshot above. To apply the style to another box, click to edit the box and select the style from the drop-down list.

Please see the References section for a complete list of built-in boxes.

Skins

Summer Cart 5.0 ships with six built-in skins for the Customer Area, and one for the Admin Area. You can change the current skin of the site from Admin Panel > Website Content > Skins.

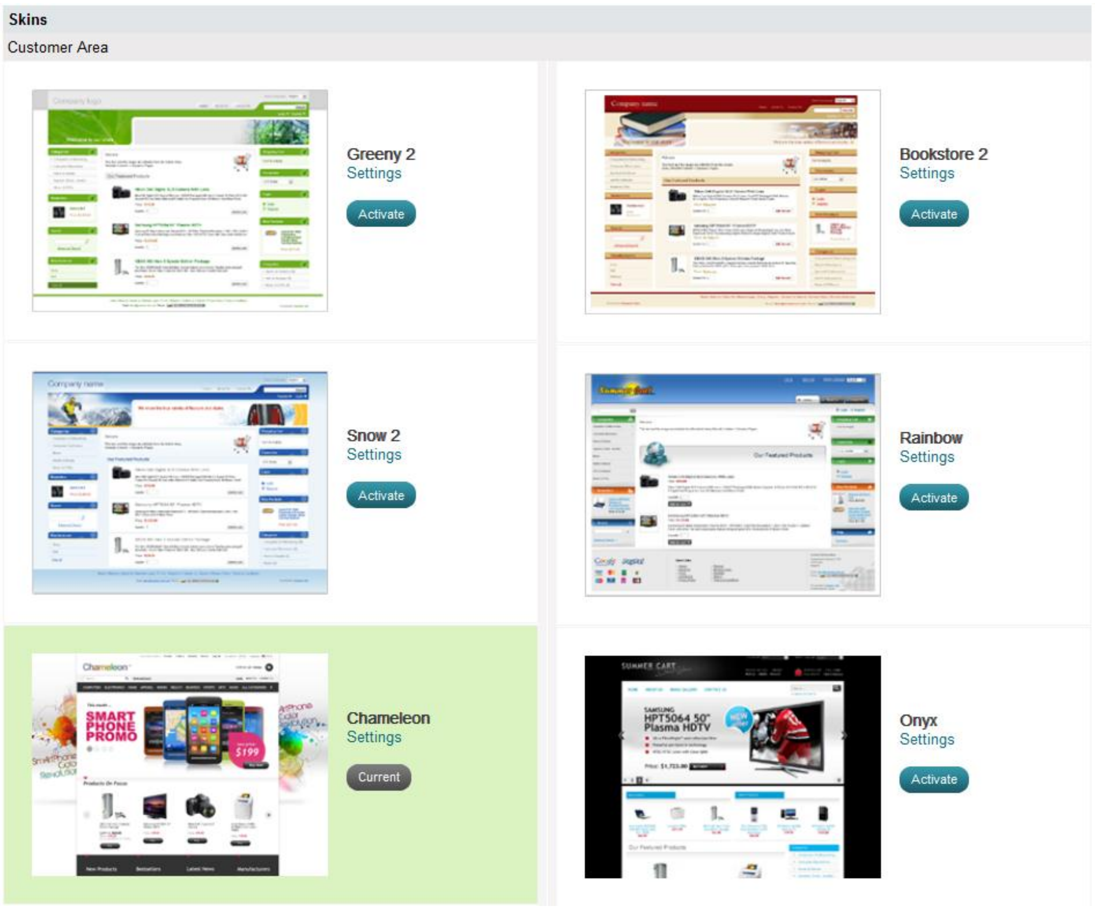


Fig. 3.1 List of installed skins

Skins are located in the /skins folder of your Summer Cart installation.

By default there are 8 subfolders:

/**bookstore2** – the Bookstore 2 skin;
/**chameleon** – the Chameleon skin;
/**default** – the default skin for the Admin Panel. Normally you shouldn't modify it;
/**greeny2** – the Greeny 2 skin;
/**onyx** – the Onyx skin;
/**rainbow** – the Raindow skin;
/**skeleton** – this is the basic skin. All Customer Area skins inherit from the skeleton skin;
/**snow2** – the Snow 2 skin.

Creating a new skin

To create a new skin, simply create a new folder in /skins, for example “myskin”. You are strongly advised to create a new skin by copying one of the existing skins. This will save you a lot of work as these skins contain much CSS and HTML that you can simply customize, and not start from scratch.

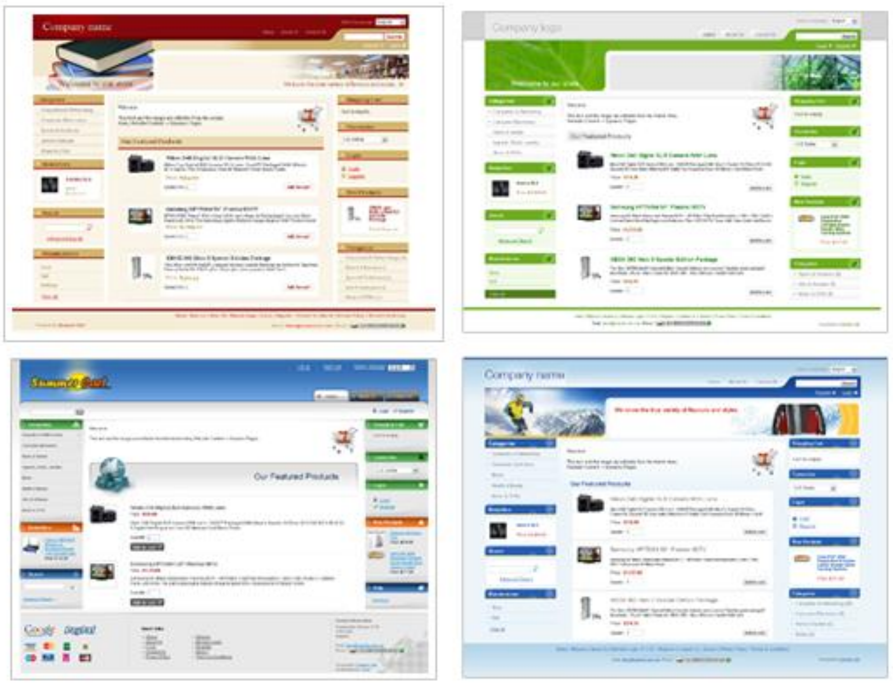


Fig. 3.2 Comparison of skins with purely CSS customizations

Skin inheritance

The best practice in skin inheritance is that all skins for the Customer Area must inherit the skeleton skin. This way a skin can only contain the files that it customizes. If a certain HTML template, CSS or JavaScript file was not changed by a certain skin, it should not contain the file, and in this case Summer Cart will use the version from the parent skin. For example, of the 6 skins that are built into Summer Cart, 3 overwrite just the CSS file (these are Bookstore, Greeny and Snow). Nevertheless, the sites look very different.

Skin definition file

All skins must have a skin definition file – skin.xml. Create it for your new skin, or customize an existing one. Let’s look at the contents of this file (I used the Chameleon skin for illustration):

```
<?xml version="1.0" encoding="UTF-8"?>
<Skin>
  <SCVersion compare="equal">4.0</SCVersion>
  <Name>Chameleon</Name>
  <SkinLocation>chameleon</SkinLocation>
  <Screenshot>skin-screenshot.png</Screenshot>
  <Areas>
    <Admin>false</Admin>
    <Customer>true</Customer>
  </Areas>
  <SkinParentLocation>skeleton</SkinParentLocation>
  <Settings>
    <Setting type="image" name="Logo" maxWidth="400" maxHeight="200"
      displayName="Logo">customer/images/logo.png</Setting>
  </Settings>
  ...
</Skin>
```

Skin definition files start with a few lines that describe the skin. These are:

Property	Description
SCVersion	The version of Summer Cart this skin is compatible with. Must match the currently installed version.
Name	Name of the skin.
SkinLocation	Location of the skin inside the /skins folder.
Screenshot	Screenshot of the skin, used on the Skins page in Admin Panel.
Areas	Whether the skin is for the Customer Area, the Admin Panel or both. Typically you would want your skin to be for the Customer Area only.
SkinParentLocation	The parent skin. This would typically be the skeleton skin.
Settings	Customizable skin settings that can be modified through the Admin Panel (see below).

Skin settings

With skin settings you can make your skins customizable through the Admin Panel. These are settings you define in your skin.xml file and use in your templates, and their values can be changed by the user in the Admin Panel. Let’s look at the settings of the Chameleon skin from the previous page for illustration:

```
<Settings>
  <Setting type="image" name="Logo" maxWidth="400" maxHeight="200"
    displayName="Logo" >customer/images/logo.png</Setting>
</Settings>
```

This defines a setting named Logo, of type image (one that the administrator can upload), and specifies that the uploaded image must have max size of 400x200 pixels and its default value would be /customer/images/logo.png. Now if we go to the Skins page in the Admin Panel and click on Chameleon's Settings link, we will see our setting:

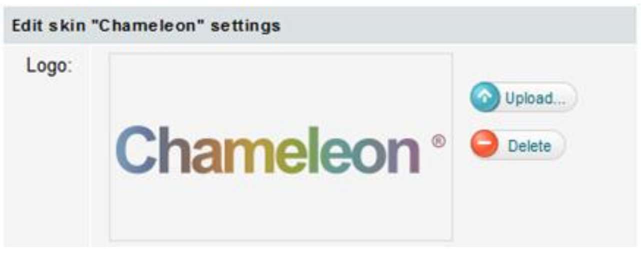


Fig. 3.3 Chameleon skin settings

Note that if the administrator deletes the value (clicks the Delete button next to the picture), the default value for the setting (in this case "/customer/images/logo.png") will be used.

There are 5 setting types supported by Summer Cart. These are:

Setting Type	Description
text	Text setting. The administrator specifies a text value that is used regardless of the language of the customer.
text_ml	Multilanguage text setting. The administrator can specify values for the setting in multiple languages.
checkbox	Boolean setting. Actual value would be either 1 (true) or 0 (false) depending whether the user checked the checkbox.
image	Image setting. Actual value would be the relative path to an image uploaded by the user.
google_map	Google Map setting. User can specify the latitude and longitude coordinates, zoom level (1-23) and map type (hybrid, roadmap, satellite or terrain). Default values for all these properties can be specified in the skin definition file.

Now let's see how a setting can actually be used in your HTML code. For example, if you have a setting named **Logo**, you can use it in your HTML by typing %%SKINSETTING**LOGO**%%, and this will be replaced with the actual value of the setting. For example, Chameleon inherits the skeleton skin (which also defines a Logo setting) and the HTML code that uses the setting looks like this:

```
<h2 id="logo">
 <a href="%%URL_INDEX%%">
  <img src="%%SKINSETTINGLOGO%%" alt="%%COMPANY_NAME%%" />
 </a>
</h2>
```

In the actual page this will be replaced with whatever the user uploaded in the Admin Panel.

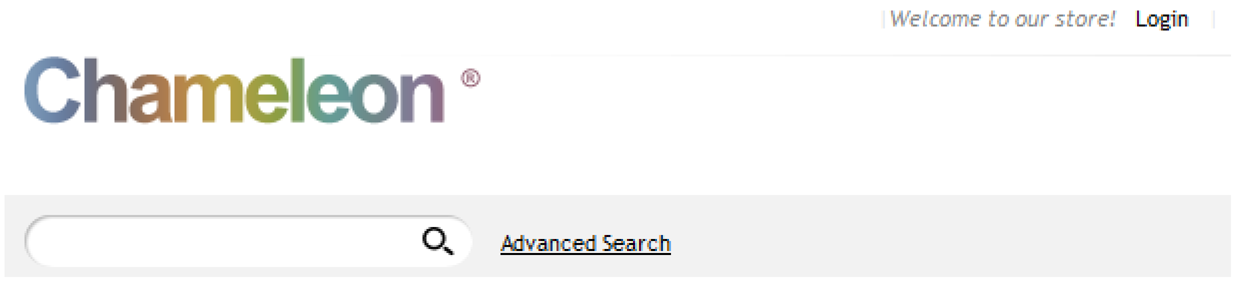


Fig. 3.4 Chameleon logo

Following is an example of a Settings section with all 3 types of settings:

```
<Settings>
  <Setting type="text" name="TextSetting"
    displayName="Text">Default Value</Setting>
  <Setting type="text_ml" name="MultilanguageTextSetting"
    displayName="Multilanguage Text"><EN>Default value</EN><BG>Стойност по
подразбиране</BG></Setting>
  <Setting type="checkbox" name="CheckboxSetting"
    displayName="Checkbox">false</Setting>
  <Setting type="image" name="Logo" maxWidth="400" maxHeight="200"
    displayName="Logo">customer/images/logo.png</Setting>
  <Setting type="google_map" name="GoogleMap" displayName="Google Map>
    <EnableGoogleMap>1</EnableGoogleMap>
    <Latitude>50.839931</Latitude>
    <Longitude>4.375988</Longitude>
    <Zoom>15</Zoom>
    <MapType>roadmap</MapType>
  </Setting>
</Settings>
```

The usage of image and text settings is straightforward. The best example for an image setting is for the site's logo image. You would use text settings when you want to allow the site administrator to change a text through the skin settings. Note that the recommended way of working with texts is that you use the <mi:text> tag, which allows the administrator to change the actual text in all languages through Admin Panel > Settings > Languages. Checkbox settings have only values of 0 (false) or 1 (true). You can use them for example in class names when the value of the checkbox determines whether some element will be visible or not. Here is an example from the Rainbow skin:

```
<Settings>
  <Setting type="checkbox" name="ShowFooterIcons"
      displayName="Show footer icons">true</Setting>
</Settings>
```

The following code illustrates how this setting is used in the skin's footer template:

```
<td id="footer_icons">
  <img src="%%SKIN_URL%%customer/images/footer_icons.png" alt=""
      class="showFooterIcons%%SKINSETTINGSHOWFOOTERICONS%%" />
</td>
```

If the setting was checked, this will result in CSS class “showFooterIcons1”, while if it was not checked the CSS class would be “showFooterIcons0”. Then the CSS file itself looks like this:

```
td#footer_icons {
  width: 372px;
  height: 184px;
}
td#footer_icons .showFooterIcons0 {
  display: none;
}
```

Finally, Google Map settings are used mainly in Contact Us pages. The Page Styles section further in this chapter provides information how you can have settings for a specific page, and the most common scenario is to have a Google Map setting specifically for the Contact Us page.

Skin settings in multiple languages

If you want your skin settings to have different display names for different languages, add a /lang folder under your skin folder (look at the built-in skins for reference). This folder can contain multiple xml files named after the language you are customizing for. In this xml file you can define the display names of your skin settings in the respective language. For example, a bg.xml file that specifies the display name of the Logo setting in Bulgarian will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<Language langCode="BG">
  <SettingDisplayName name="Logo">Лого</SettingDisplayName>
</Language>
```

Headers and footers

One of the new features in Summer Cart 5.0 is that your skin can include multiple header and footer templates, and the administrator can select the actual header and footer to be used in a specific layout on the site through the Admin Panel. Different layouts can use different headers and footers.

As headers typically have elements that are common to all header variants, we have a master header template (header.tmpl) and header variants are called “subheaders” because they are added to the master header, typically at the end. (In the Chameleon skin, the master header contains only the <head> tag, while subheaders start shortly after the <body> tag) If you want to have subheaders that are completely different to each other, you can leave the master header template empty and add all the code in the subheaders.

This is not the case with footers, where your skin can contain multiple footers and there is no master footer template.

Header and footer options are defined in the skin definition file:

```
<Subheaders>
  <Subheader>
    <Name>Header Layout 1</Name>
    <Template>customer/subheaders/subheader1.tmpl</Template>
    <Screenshot>.../header-variant-1.jpg</Screenshot>
  </Subheader>
  <Subheader>
    <Name>Header Layout 2</Name>
    <Template>customer/subheaders/subheader2.tmpl</Template>
    <Screenshot>.../header-variant-2.jpg</Screenshot>
  </Subheader>
  ...
</Subheaders>
<Footers>
  <Footer>
    <Name>Footer Layout 1</Name>
    <Template>customer/footers/footer1.tmpl</Template>
    <Screenshot>.../footer-variant-1.jpg</Screenshot>
  </Footer>
  <Footer>
    <Name>Footer Layout 2</Name>
    <Template>customer/footers/footer2.tmpl</Template>
    <Screenshot>.../footer-variant-2.jpg</Screenshot>
  </Footer>
  ...
</Footers>
```

As you can see a header or footer definition is straightforward, having “Name” (shown in the Admin Panel), “Template” (path to a *.tmpl file) and “Screenshot”.

This is how the master header template, header.tmpl, links to a subheader:

```
<!DOCTYPE .../>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  ...
</head>

<body class="lang-%%CURRENT_LANGUAGE_CODE%% %%BODY_CLASSES%%"
%%LANGUAGE_DIRECTION%%>

<div id="body-container">
<div id="container">

%%HTML_SUBHEADER%%
```

The Chameleon skin has 5 header options and 5 footer options. One can be selected for a specific layout through the design editor:

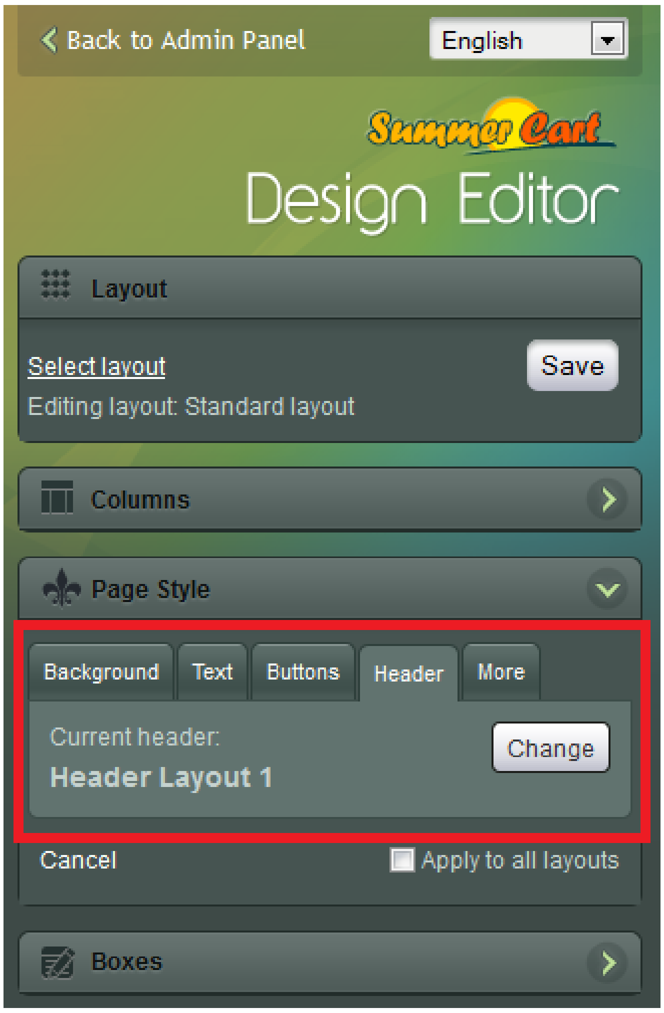


Fig. 3.5 Changing a layout's header and footer

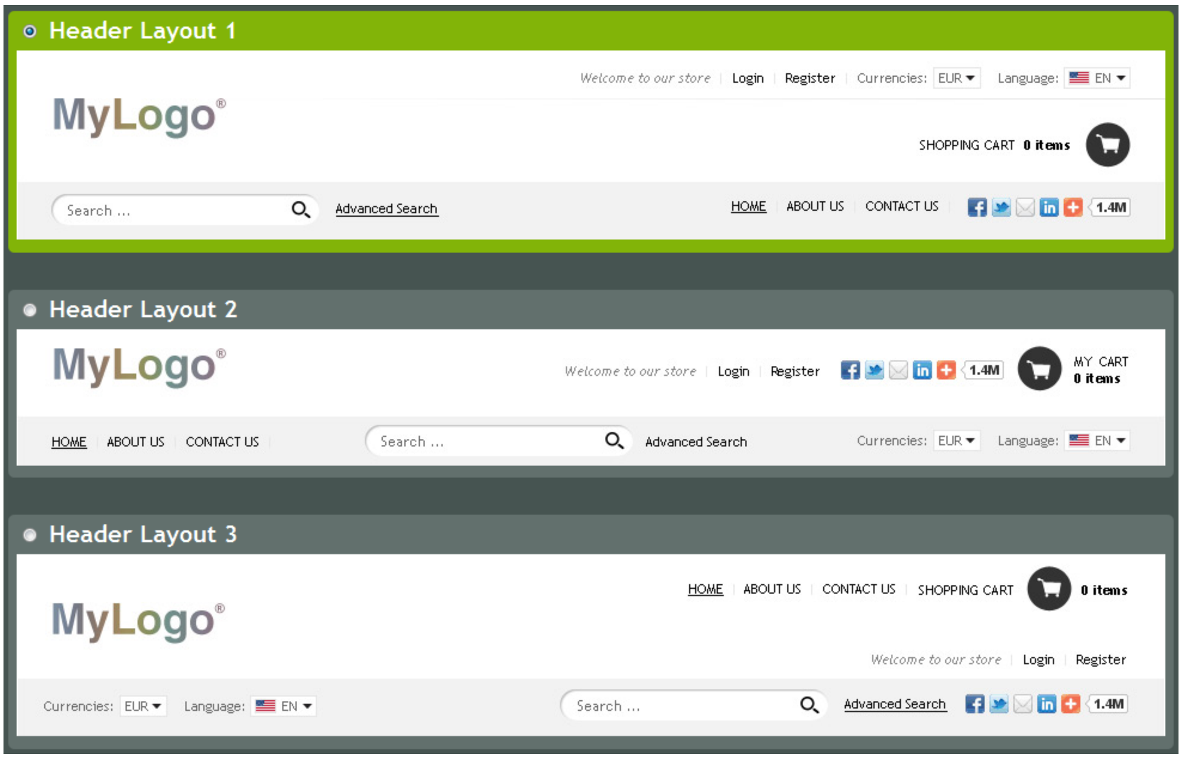

Header Layout 1
MyLogo
Welcome to our store | Login | Register | Currencies: EUR | Language: EN
SHOPPING CART 0 items
Search ...
Advanced Search
HOME | ABOUT US | CONTACT US
1.4M
Header Layout 2
MyLogo
Welcome to our store | Login | Register
1.4M
MY CART 0 items
HOME | ABOUT US | CONTACT US
Search ...
Advanced Search
Currencies: EUR | Language: EN
Header Layout 3
MyLogo
HOME | ABOUT US | CONTACT US | SHOPPING CART 0 items
Welcome to our store | Login | Register
Currencies: EUR | Language: EN
Search ...
Advanced Search
1.4M

Fig. 3.6 Standard header options shipping with the Chameleon skin

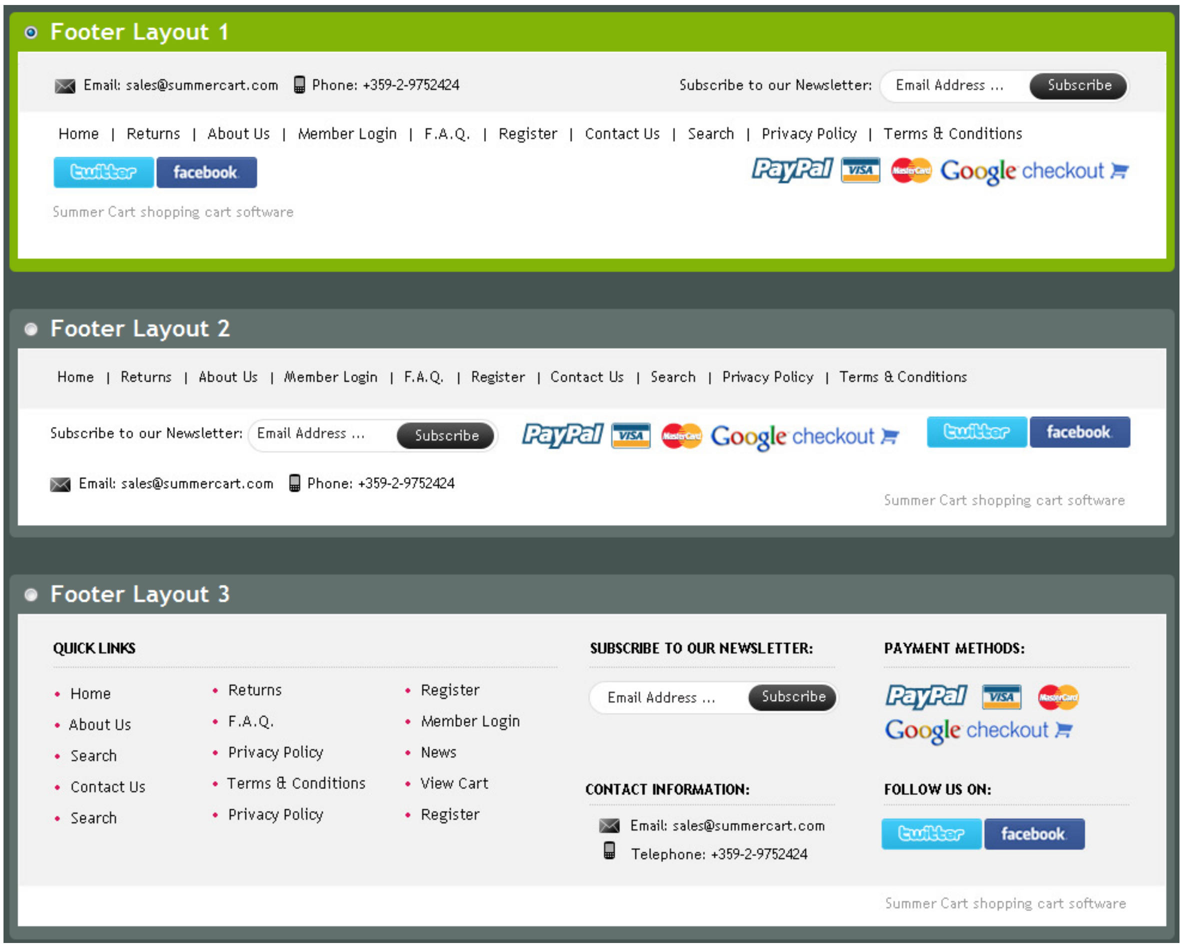

Footer Layout 1
Email: sales@summercart.com Phone: +359-2-9752424
Subscribe to our Newsletter: Email Address ... Subscribe
Home | Returns | About Us | Member Login | F.A.Q. | Register | Contact Us | Search | Privacy Policy | Terms & Conditions
twitter facebook
PayPal VISA MasterCard Google checkout
Summer Cart shopping cart software
Footer Layout 2
Home | Returns | About Us | Member Login | F.A.Q. | Register | Contact Us | Search | Privacy Policy | Terms & Conditions
Subscribe to our Newsletter: Email Address ... Subscribe
PayPal VISA MasterCard Google checkout
twitter facebook
Email: sales@summercart.com Phone: +359-2-9752424
Summer Cart shopping cart software
Footer Layout 3
QUICK LINKS
Home
About Us
Search
Contact Us
Search
Returns
F.A.Q.
Privacy Policy
Terms & Conditions
Privacy Policy
Register
Member Login
News
View Cart
Register
SUBSCRIBE TO OUR NEWSLETTER:
Email Address ... Subscribe
CONTACT INFORMATION:
Email: sales@summercart.com
Telephone: +359-2-9752424
PAYMENT METHODS:
PayPal VISA MasterCard
Google checkout
FOLLOW US ON:
twitter facebook
Summer Cart shopping cart software

Fig. 3.7 Three of the standard footer options shipping with the Chameleon skin

Skin layouts

A skin must have at least one layout defined in its definition file (skin.xml). By convention it is named “Standard layout” and marked as being the default layout, so pages use it unless specified otherwise.

```
<CMSLayouts>
 <CMSLayout>
  <CMSLayoutName>Standard layout</CMSLayoutName>
  <CMSLayoutIsDefault>1</CMSLayoutIsDefault>
  <CMSLayoutType>thin-wide</CMSLayoutType>
  <CMSLayoutStyleJson>...</CMSLayoutStyleJson>
  <CMSLayoutColumns>
    <CMSLayoutColumn>
      <Column>left</Column>
      <Width>20%</Width>
    </CMSLayoutColumn>
     <CMSLayoutColumn>
       <Column>middle</Column>
       <Width>80%</Width>
     </CMSLayoutColumn>
  </CMSLayoutColumns>
  <Subheader>Subheader Layout 1</Subheader>
  <Footer>Footer Layout 1</Footer>
  <Boxes>
      <Box>
        <CMSLayoutBoxColumn>top</CMSLayoutBoxColumn>
        <CMSBoxType>Dropdown Categories</CMSBoxType>
      </Box>

      <Box>
        <CMSLayoutBoxColumn>left</CMSLayoutBoxColumn>
        <CMSBoxType>Categories</CMSBoxType>
      </Box>
      <Box>
        <CMSLayoutBoxColumn>left</CMSLayoutBoxColumn>
        <CMSBoxType>Active Filters</CMSBoxType></Box>
      <Box>
        <CMSLayoutBoxColumn>left</CMSLayoutBoxColumn>
        <CMSBoxType>Product Attributes Filter</CMSBoxType>
      </Box>
      <Box>
        <CMSLayoutBoxColumn>left</CMSLayoutBoxColumn>
        <CMSBoxType>Product Comparison</CMSBoxType>
      </Box>
      ...
  </Boxes>
  <CMSLayoutPages>
    <CMSLayoutPage>com.summercart.dynamic_page.index</CMSLayoutPage>
  </CMSLayoutPages>
 </CMSLayout>
 ...
</CMSLayouts>
```

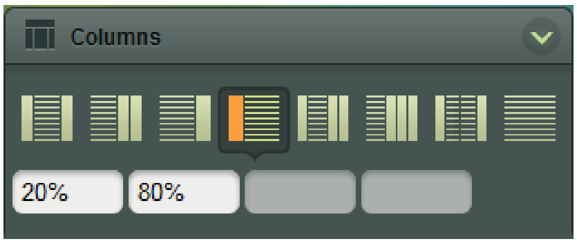


Fig. 3.8 Selecting columns and relative widths in the layout settings

Property	Description
CMSLayoutName	Name of the layout. Use "Standard layout" by convention for the default layout.
CMSLayoutIsDefault	Whether this is the default layout. Only one layout can be set as default. If a layout is not the default, you can specify which pages to use the layout through the CMSLayoutPages property (see below). The administrator can later change the layout of any page.
CMSLayoutType	Specifies the columns of a layout, and can be later modified through the Columns section in the Design Editor (see fig. 3.8). "thin" means a thin column for boxes, while "wide" means a wide column for content. The type on fig. 3.8 is therefore "thin-wide".
CMSLayoutStyleJson	Json data that can contain predefined options that are read by the PHP scripts. You would typically leave this by default.
CMSLayoutColumns	This is an optional section where you can define the initial widths of the columns in the layout. If the section is missing, default values will be used. You can see what the default values are if you open a layout in the Design Editor and change the type of the layout in the Columns section. The system will automatically assign the default widths for all columns. For example, by default the "thin-wide" type has widths of 20% and 80%, so in the code above we could have omitted this section. Column names are "left", "middle", "right" and "extraright".
Subheader	What subheader to be used by default. Optional.
Footer	What footer to be used by default. Optional.
Boxes	Defines which boxes will be there on the layout by defualt and in which sections. In the example above, we have the "Dropdown Categories" box in the top section, and "Categories", "Active Filters", "Product Attributes Filter" and "Product Comparison" in the left section. (From top to bottom they are ordered like they appear in the skin definition file.). For a complete list of box names please see the References section at the end of this document.

CMSLayoutPages	If this is not the default layout, in this section you specify which pages should use the layout by default. For a complete list of page names please see the References section at the end of this document.

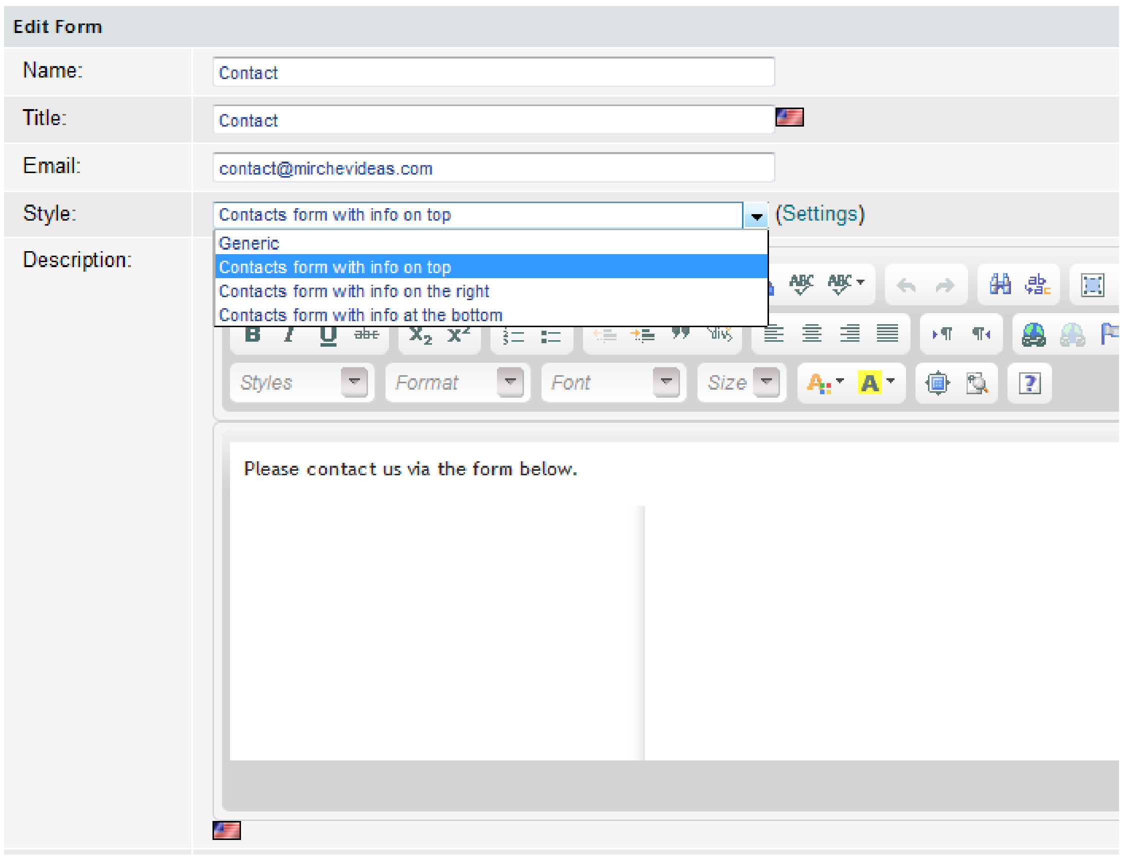
Page styles

In Summer Cart 5.0 you can define multiple styles for a single page. The administrator can then select which of the styles will be actually used for the page. Each style has its own template and may have its own set of settings. For example, the Chameleon skin defines 3 different page styles for forms. This way the administrator can select a different page style for each form in the system (e.g. the one on the Contact Us page).

```
<PageStyle pageType="com.summercart.dynamic_page.form">
  <Name>Contacts form with info on top</Name>
  <PageStyleCode>contacts-map-top</PageStyleCode>
  <Screenshot>...</Screenshot>
  <Template>customer/forms/form-info-top.tmpl</Template>
  <Settings>
    <Setting type="text_ml" name="Header1" displayName="Header 1">
      <EN>Address</EN><BG>Адрес</BG></Setting>
    <Setting type="text_ml" name="Text1" displayName="Text 1" />
    <Setting type="text_ml" name="Header2" displayName="Header 2">
      <EN>Working hours</EN><BG>Работно време</BG></Setting>
    <Setting type="text_ml" name="Text2" displayName="Text 2" />
    <Setting type="google_map" name="GoogleMap" displayName="Google Map">
      <EnableGoogleMap>1</EnableGoogleMap>
      <Latitude>50.839931</Latitude>
      <Longitude>4.375988</Longitude>
      <Zoom>15</Zoom>
      <MapType>roadmap</MapType>
    </Setting>
    <Setting type="image" name="HeadingImage" displayName="Heading image"
     maxWidth="683" maxHeight="400" />
  </Settings>
</PageStyle>
```

Property	Description
pageType	Determines for which page is the page style. For a list of pages see References > Pages later in this document.
Name	Name of the page style, displayed in the Style dropdown in the Admin Panel (see fig. 3.9).
PageStyleCode	Unique code of the page style.

Screenshot	Path to a screenshot of the page style. Chameleon and other built-in skins use screenshots of size 100x100 px.
Template	Path to the page style template file. The file will typically start with a <div> tag with id matching the code of the page style, so you can target the page style in the CSS.
Settings	List of available settings for the page style, which can be changed by the administrator. Check fig. 3.10 for the settings and fig. 3.11 for the result. Identical to skin settings, so for available options consult with the Skin Settings section earlier in this chapter.

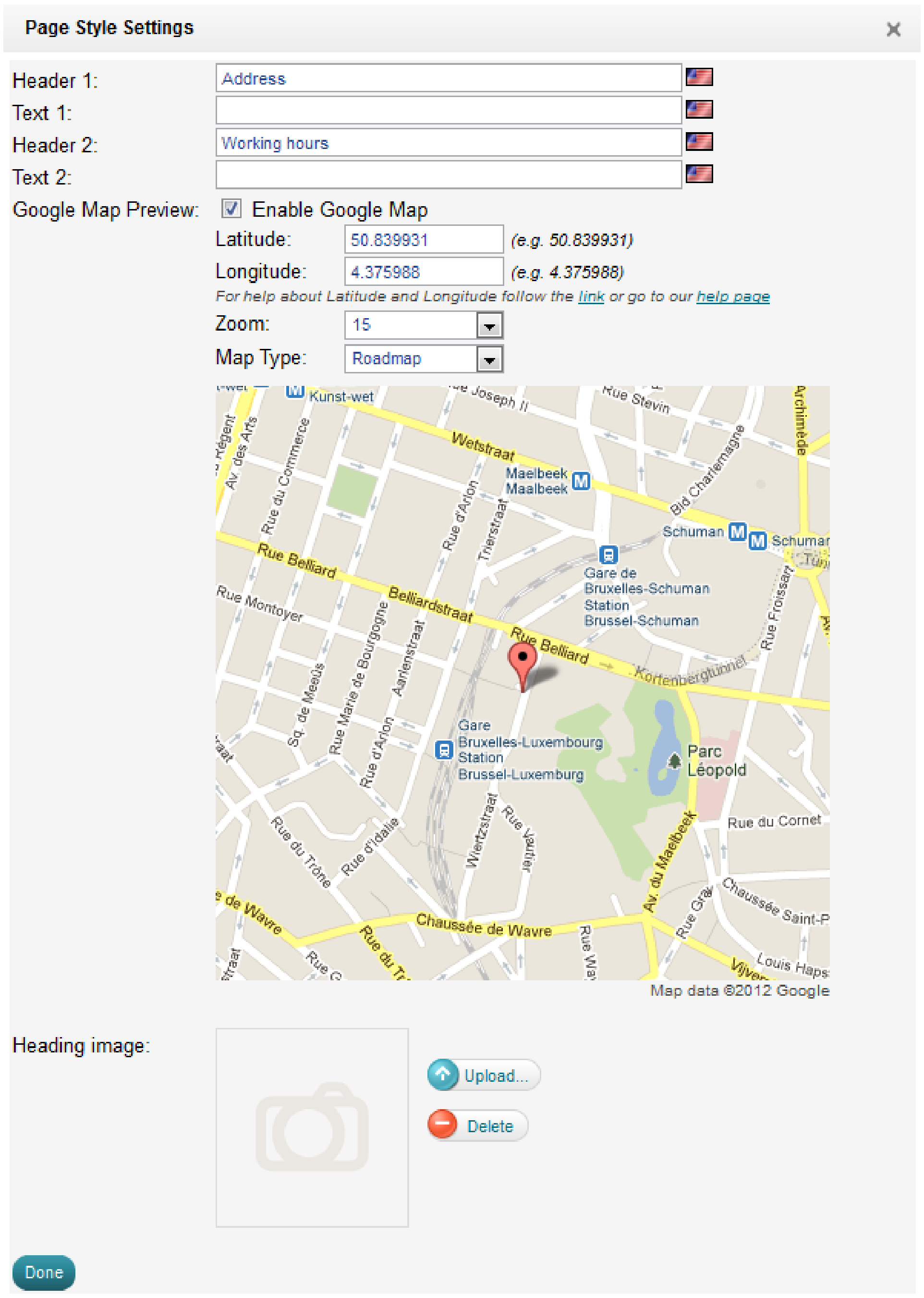


Edit Form

Name:	Contact
Title:	Contact
Email:	contact@mirchevideas.com
Style:	Contacts form with info on top (Settings)
Description:	Please contact us via the form below.

Generic
Contacts form with info on top
Contacts form with info on the right
Contacts form with info at the bottom
Styles
Format
Font
Size

Fig. 3.9 Changing the style of a form



Page Style Settings

Header 1:	Address
Text 1:	
Header 2:	Working hours
Text 2:	
Google Map Preview:	[✓] Enable Google Map
Latitude:	50.839931 (e.g. 50.839931)
Longitude:	4.375988 (e.g. 4.375988)

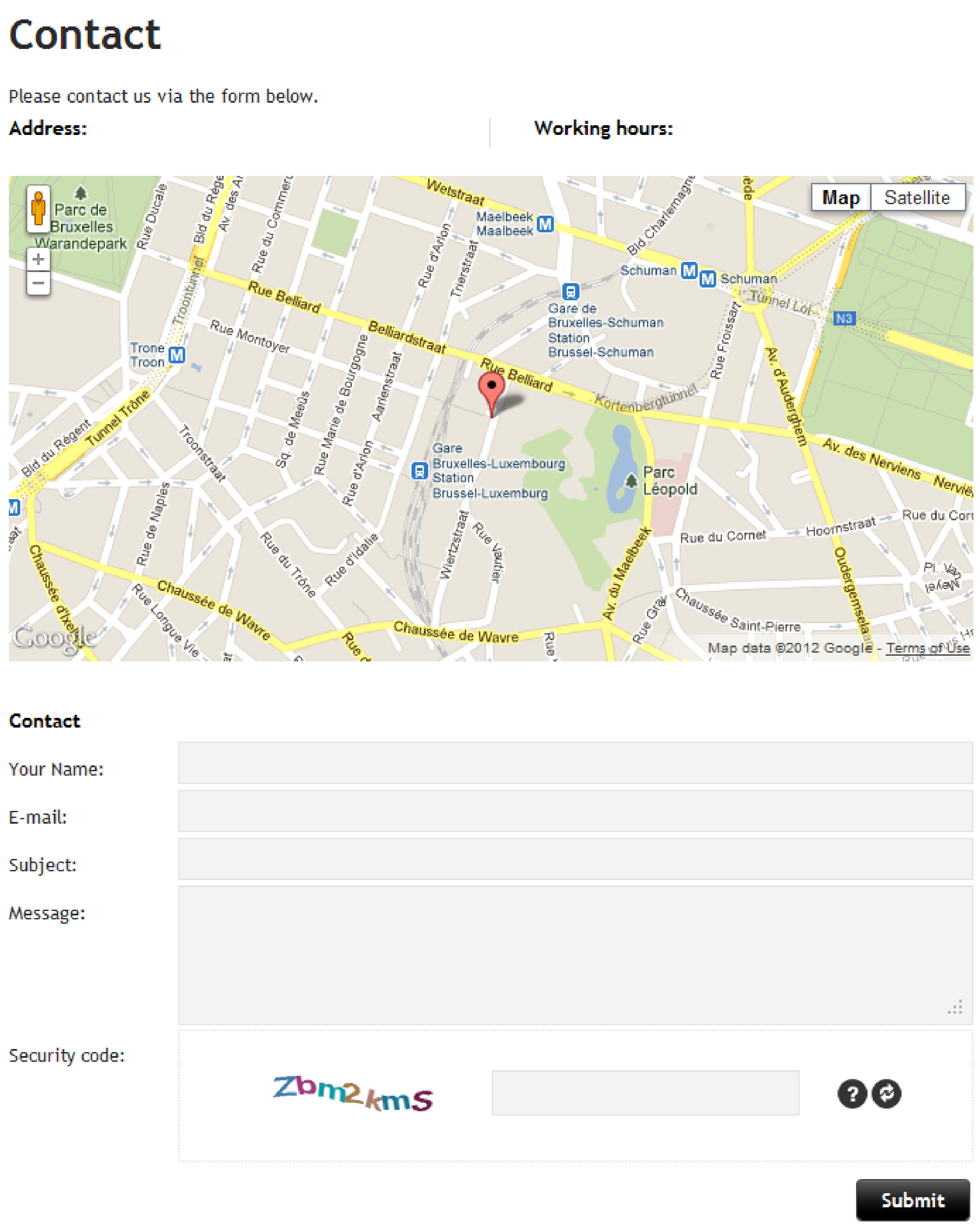
For help about Latitude and Longitude follow the link or go to our help page

Zoom:	15
Map Type:	Roadmap

Map data ©2012 Google

Heading image:

Fig. 3.10 Settings for the page style as defined in skin.xml



Contact

Please contact us via the form below.

Address: | **Working hours:**

Map
Satellite
Parc de Bruxelles Warandepark
Trone Troon
Maelbeek Maalbeek
Schuman
Gare de Bruxelles-Schuman Station Brussel-Schuman
Gare Bruxelles-Luxembourg Station Brussel-Luxemburg
Parc Léopold
Rue Belliard
Belliardstraat
Wetstraat
Av. des Nerviens
Chaussée de Wavre
Rue du Cornet
Chaussée Saint-Pierre
Map data ©2012 Google - Terms of Use

Contact

Your Name:	
E-mail:	
Subject:	
Message:	
Security code:	Zbm2kmS

Fig. 3.11 The Contact Us page with the selected page style and settings

File structure

A bare bone skin in Summer Cart needs only its own folder and a skin.xml file. These are the only required elements of a skin, although such a skin will be a complete copy of the basic skeleton skin. Note that there are several pages (typically ones listing products) where items can be displayed in either a list or a grid, and the customer can switch between the two at any time. These pages have two separate templates – one for the List view (e.g. index.tmpl) and one for the Grid view (e.g. index_grid.tmpl). Please consult with the References section for a full list of skin files and folders.

Template structure

Templates in Summer Cart are XHTML files with extension ".tmpl". Templates can contain various placeholders that are parsed at runtime by PHP scripts to produce dynamic content. You can find all the templates in the system in the skeleton skin, and if you want to customize a certain template file you must copy it over from the skeleton to your own skin in the same folder (e.g. /customer/cmsboxes). Because your skin inherits from the skeleton skin, any file that is missing from your skin's folder will be loaded from the skeleton skin.

There are two types of dynamic content placeholders in Summer Cart: the %% variables and the <mi:> tags.

The %% variables are of the form %%VARIABLE_NAME%%. These will be replaced with actual values at runtime. Most of the %% variables are specific to a page, and the PHP script that is serving the page knows what variables it supports. These you can see in the template you are customizing. There are also variables that are accessible from any page. Please consult with the References section for a full list of %% variables and their usage.

The <mi:> tags are of the form <mi:tagName>…</mi:tagName>. The main difference between the <mi:> tags and the %% variables is that the tags have an HTML value inside of them, and at runtime this value is processed to form the end result. With the %% variables the PHP script just replaces the variable with the generated dynamic content. Tags supported by Summer Cart are:

Tag	Description
<mi:skinurl>URL**</mi:skinurl>**	Converts the given URL to one that is relative to your skin folder. The main benefit over the %%SKIN_URL%% variable is that if the URL is not found inside your skin, it will be looked for in its parent skin. This way if you need to customize a single template you do not need to copy over from the parent skin all the base files that your template references.
<mi:text>text**</mi: text>**	Summer Cart is a platform that supports multiple languages. Typically, you would want any text that appears on the Customer Area to be localizable. For this to work you should use <mi:text> tags, and inside them put your original text (typically that would be the text in English). Then when you go to Admin Panel > Settings > Languages, all these tags in your templates are extracted and displayed in a list. The store owner can install a language pack or hire a translator

	to go through the list and translate the texts, and then the localized versions will be displayed to the end user based on his language preferences.
<mi:section>…</mi:section>	These are dynamic sections of HTML, much like a template within a template. Each section has a name, and this is how the PHP script that parses the template finds the section. Most sections in Summer Cart are inline, which means that their content is placed directly between the opening and the closing tags. Some sections have a src attribute, and these have their contents loaded from an external file. Basically you should remember to keep all the sections in place, because the PHP script uses them to show or hide certain HTML code, as well as to repeat certain HTML code multiple times, for example with product listings. **Note**: If you copied a template from the skeleton to your skin, and that template has sections with their content loaded from other files, you must copy these other files to your skin as well.

HTML and CSS structure

The content of each Summer Cart page is generated from a number of templates as illustrated on fig. 2.4. First, the header template is processed, then the top area, the left area, the page template (actual page content is rendered inside the page template), the right area, the bottom area and finally the footer template. All these templates are located in the /customer folder.

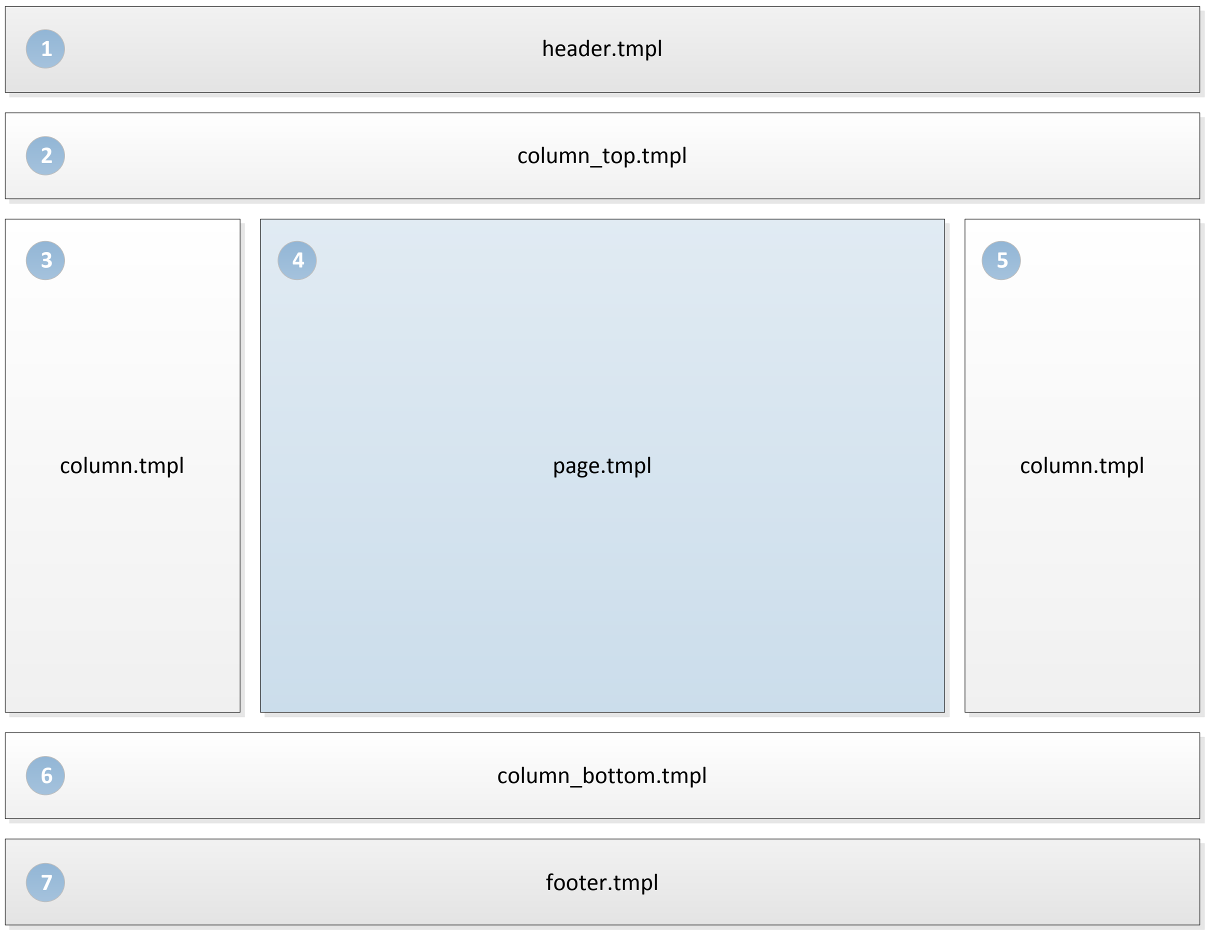


Fig. 3.12 Illustration of what templates are used to build a page

The CSS files for a skin are located in its /css folder. A typical skin would have all of its CSS rules defined in a main.css file, and the other css files in the folder would be related to 3rd party libraries (e.g. jquery.css, colorbox.css) or browser-specific customizations (e.g. ie6.css).

Now let us examine the HTML and CSS structure of a typical Summer Cart page, in this case the Home page. Note that the common elements can be referenced by ID, and some of them also have CSS classes so you can reference groups of elements by common attribute. For example, all horizontal areas (the top and bottom areas) have the “h-column” class, while the vertical ones (the left and right areas) have the “v-column” class, although all four areas can be referenced by their ID as well.

```
<html>
  <head>
    ...
  </head>
  <body class="lang-en dynamic-page dynamic-page-index">
    <div id="body-container”>
      <div id="container">
        <div id="header">...</div>
        <div id="column-top" class="h-column clearfix">...</div>
        <div class="mainWrapper">
          <table id="main">
          <tr>
            <td id="column-left" class="v-column">...</td>
            <td id="content">...</td>
            <td id="column-right" class="v-column">...</td>
          </tr>
          </table>
        </div>
        <div id="column-bottom" class="h-column clearfix">...</div>
        <div id="footer" class="clearfix">...</div>
      </div>
    </div>
  </body>
</html>
```

The Body of each page uses a number of CSS classes. What classes are used depends on whether it is a Dynamic or Rich Text page, and whether additional CSS classes are specified in the layout that the page is using. One common CSS class that is set on all pages is the language class. In the example above, it is “lang-en”. What language class will be used depends on what language the customer selected in the Customer Area. The available languages are set up in Admin Panel > Settings > Languages. Each language has a Language Code setting, and it is used in the class name (e.g. the code for English is “en”, so the language class name is “lang-en”). This gives you the option to apply CSS rules for a specific language only.

Dynamic pages’ bodies have a language class, as well as the “dynamic-page” and “dynamic-page-…” classes (“…” being the respective page name). In addition they have the CSS classes inherited from the page’s layout.

```
<body class="lang-en dynamic-page dynamic-page-index">
```

All dynamic pages have their own ID’s, e.g. “index-page”, “login-page”, etc. Some of the pages have 2 different views – List view and Grid view, and they have two separate templates. For example, the Home page has two templates: “index.tmpl” and “index_grid.tmpl”, and the customer can switch between the two at any time. The container divs of the two templates use different CSS classes.

```
<div id="index-page" class="list-page">...</div>
```

The element above is from index.tmpl, while the one below is from index_grid.tmpl.

```
<div id="index-page" class="grid-page">...</div>
```

Rich Text pages' bodies have a language class, as well as classes inherited from their layout.

```
<body class="lang-en">
```

Boxes have 2 classes: "box" and "box-#", where # is the box ID in Summer Cart. This way you can write CSS rules that target specific boxes, even custom HTML boxes created in the Admin Panel. Each of the built-in boxes also has its own HTML ID (e.g. "cart-box", "login-box", etc.). Boxes typically contain two divs, one with the box title, and one with the box content.

```
<div id="cart-box" class="box box-%%CMSBOXID%%">
  <div class="box-title-wrapper">
    <h2 class="box-title"><mi:text>Shopping Cart</mi:text></h2>
  </div>
  <div class="box-content">
    ...
  </div>
</div>
```

When you create custom HTML boxes from the Admin Panel, you specify the box title and content. In this case the "cms_html_box.tmpl" template is used from /customer/cmsboxes. These boxes have the "html-box" CSS class. From the Admin Panel you can change the visibility of the title div, and so a <mi:section> tag is used around the box title div to show or hide it.

```
<div class="box box-%%CMSBOXID%% html-box">
  <mi:section name="cmsHtmlBoxHasTitle">
    <div class="box-title-wrapper">
      <h2 class="box-title">%%CMSHTMLBOXTITLE%%</h2>
    </div>
  </mi:section>
  <div class="box-content">%%CMSHTMLBOXCONTENT%%</div>
</div>
```

There are several boxes that have similar content. For example, the Bestsellers box and the New Products box both show a list of products. These boxes have common CSS classes, so you can reference all of them with a single CSS rule.

```
<div id="bestsellers-box" class="box box-%%CMSBOXID%% products-box">
  ...
</div>
```

On numerous places in the templates, you will see 2 or 3 divs being placed in one another. This is so various desired effects, such as round corners, can be defined purely in CSS, without the need of changing the HTML code.

```
<div class="page-content">
<div class="page-content-bg">
<div class="page-content-bg-repeat">
...
</div>
</div>
</div>
```

Layouts can be targeted by setting the “CSS Class Names” field in the layout details with a list of comma-separated CSS class names. Note they will be added directly to the document body classes.

Layout areas can be targeted by their ID (e.g. “column-top”, “column-right”) as well as by their orientation (e.g. “h-column”, “v-column”).

```
<div id="column-top" class="h-column clearfix">...</div>
```

Menus are targeted based on location. The main menu can be targeted using the “main-menu” class, and the fast menu using the “secondary-menu” class. Both menus are displayed in the header of the site, and therefore their HTML template can be found in header.tmpl. The footer menu can be targeted using the “footer-menu” class, and its HTML is in the footer.tmpl file. Boxes that contain menus will have the “box-menu” class, and you can target a specific menu using the box ID (e.g. “box-23”).

Titles (headers) are typically used inside two divs, so you can easily customize all titles (using the “title-wrapper” class), titles with specific level (“h1-title”, “h2-title”, etc.), or titles with specific alignment (“title-left”, “title-right”).

```
<div class="h1-title title-wrapper">
  <div class="title-left">
    <h1 class="title-left">Contact</h1>
  </div>
</div>
```

Breadcrumbs are generated inside two divs, “breadcrumb” and “breadcrumb-bg”, and the links are put into spans. This way you can easily put icons between breadcrumb links. The active link would have the “active” class.

```
<div id="breadcrumb">
<div class="breadcrumb-bg">
  <span>
    <a href="/index.html">Home</a>
  </span>
  <span>
    <a href="/category/7/game-consoles.html">Game consoles</a>
  </span>
  <span class="active item fn">Sony Playstation 3 120GB Slim Console</span>
</div>
</div>
```

Links are best targeted as a whole, or per specific page or box using page and box-based CSS selectors.

Inputs can be targeted as a whole (e.g. input[type="…"]) or using the two common classes, “input-text” for textboxes and “input-submit” for buttons. Throughout the templates, buttons are used in their own span element with class “button”. This allows for various effects purely through the CSS, such as expandable buttons with round corners.

```
<input type="text" class="input-text" name="Email" id="Email" …/>
```

```
<div class="button-container">
  <span class="button">
    <input class="input-submit" type="submit" value="Submit">
  </span>
</div>
```

Finally, there are a number of common CSS rules that are used in many templates. These rules are listed near the end of the main.css file so they have priority over other rules. Following is a list of these common rules:

```
.fleft {float: left;}
.fright {float: right;}
.right, th.right {text-align: right;}
.left, th.left {text-align: left;}
.center {text-align: center;}
.inline {display: inline;}
.wide {width: 100%;}
.clear {clear: both;}
.clearfix {display: block;}
.hidden {display: none;}
```

Code style and conventions

Here we will discuss the code style we are following in Summer Cart and the conventions we are using. Although you are not required to follow the same style and conventions, you are strongly advised to do so, as your code will be easier to read and maintain.

For HTML, use lower-case HTML tags and double quotes.

```
<input class="input-submit" type="submit" value="Submit">
```

Make sure you read and understood the HTML Structure and CSS Structure chapters, and you follow the same structures for your templates.

For CSS, keep your CSS rules in a single file called “main.css”, and keep 3rd party CSS rules (e.g. jquery.css) and browser customizations (e.g. ie6.css) in separate CSS files. Use lower case class names separated by dashes. Keep CSS rules on single lines.

```
#main-menu {float: right; margin: 0; padding: 0; list-style: none none;}
#main-menu li {float: left; margin: 0 15px 0 0; padding: 0 0 0 0; list-style:
none none;}
```

CSS rules are organized into sections using /* Section */ comments. Sections would typically be named after the element being customized. For example, the CSS rules for the Login box will be listed after a /* Login box */ comment.

```
/* Login box */
#login-box a {float: left; width: 80px;}
#login-box a.wishlist {margin: 0;}
```

How do I...

Customize the CSS of a site through the Admin Panel?

You can't. To modify the CSS of a site you have to create a new skin (or copy an existing one). A basic skin that only customizes the CSS contains only a few files and can be created in a few minutes. Read how in the Skins section.

Set CSS rules for a specific layout?

Layouts have a property called "CSS Class Names". Whatever you type here will be the CSS class of the body of pages that use this layout.

Apply custom HTML code?

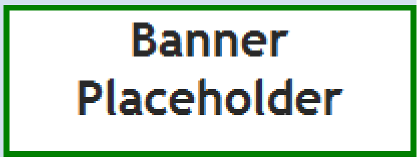
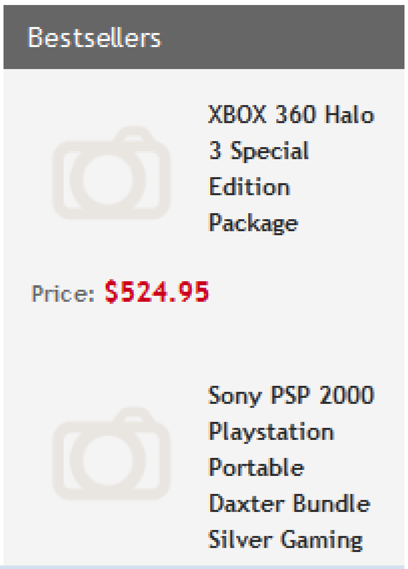
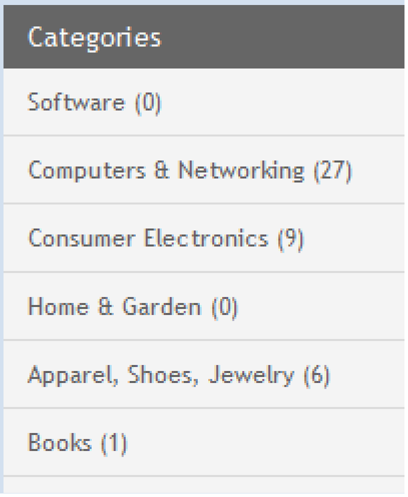
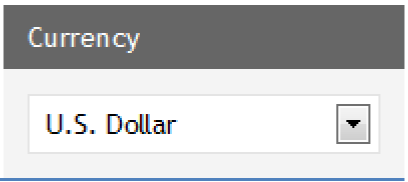
There are several places where you might want to put custom HTML. If you need the code in a specific page, you can put it in the Top or Bottom area that all pages have, including rich text pages and dynamic pages. You can also create a custom XHTML box, and type any HTML there. Then you can reuse the box across multiple pages and layouts.

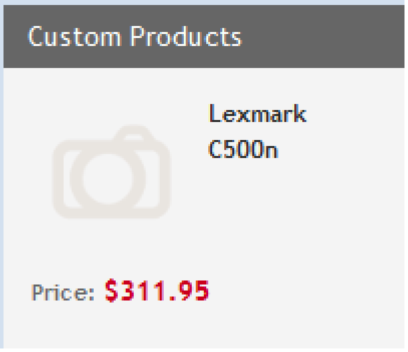
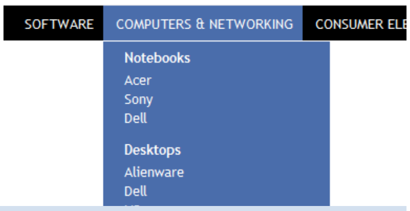
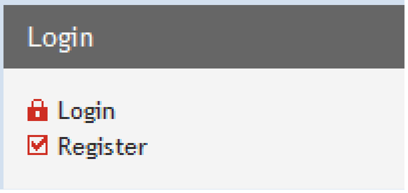
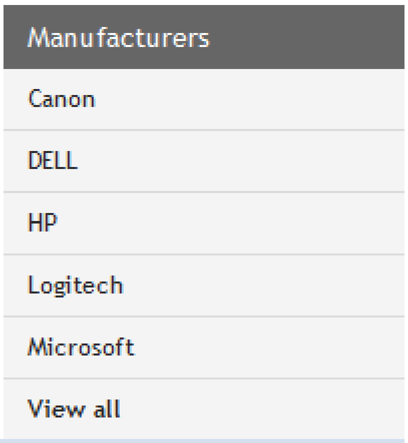
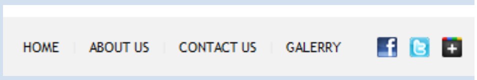
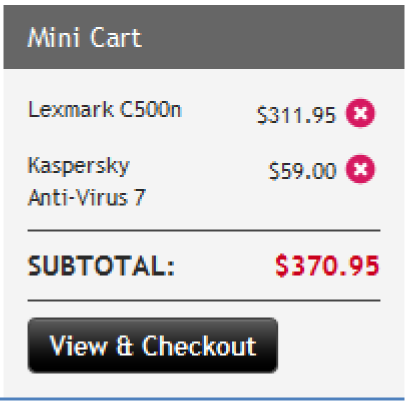
Modify the HTML or CSS code of an element?

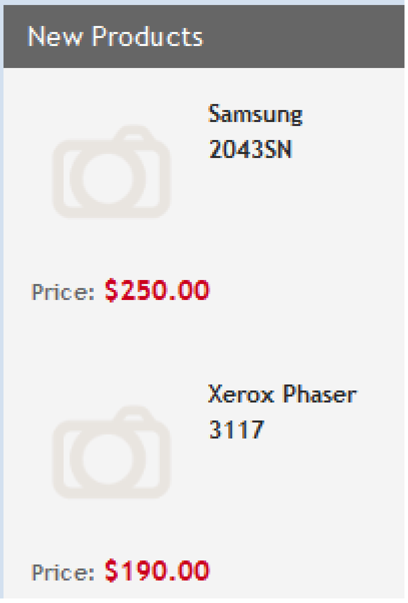
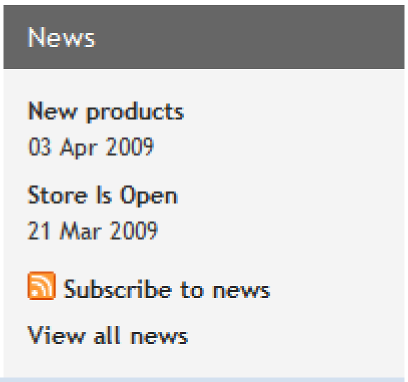
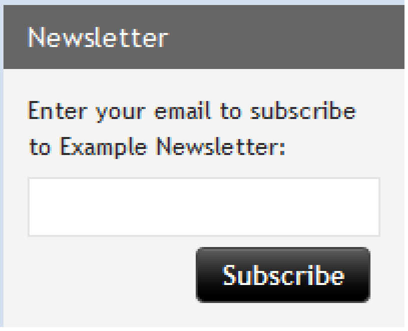
You can apply many design changes directly in the Design Editor. However, if you need to modify the HTML or CSS code of an element (e.g. a dynamic page or box), consult with the References section to find the name of the template file that corresponds to the page or box you want to modify. For design changes (no programming modifications) you are advised to create a new skin and apply your changes there. Skins can inherit ones that already exist – read how in the Skins chapter.

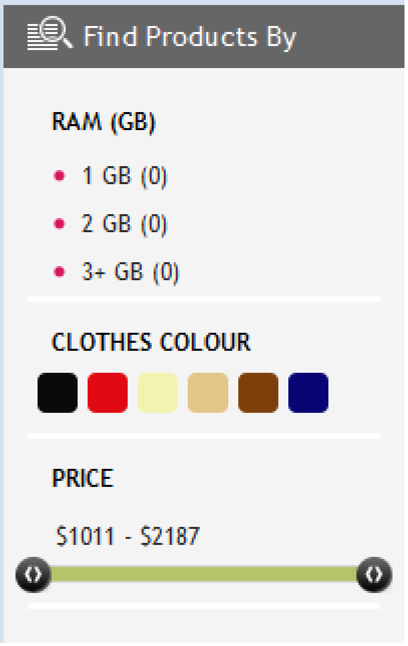
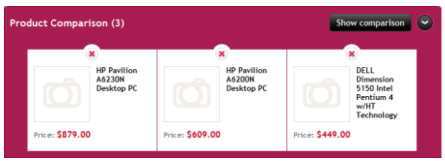
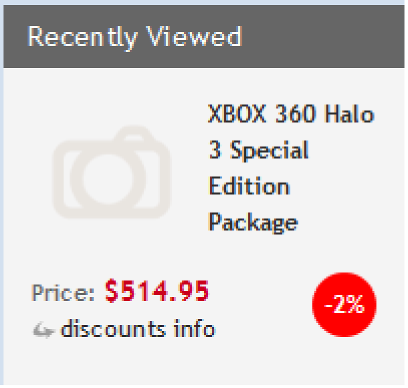
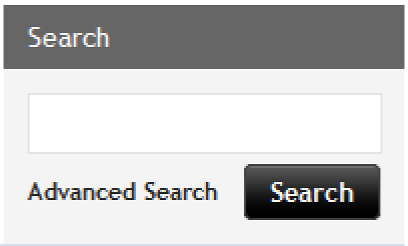
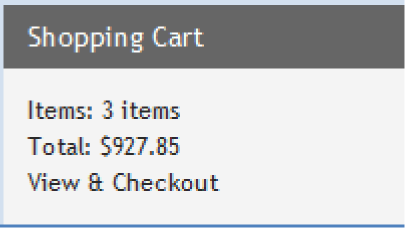
References

Boxes

Box Name	Files	Screenshot
Banners	modules\system\banners\customer\banners.php	 Banner Placeholder
Bestsellers	modules\system\blocks\customer\bestsellers_box.php <skin>\customer\cmsboxes\bestsellers.tmpl	 Bestsellers XBOX 360 Halo 3 Special Edition Package Price: $524.95 Sony PSP 2000 Playstation Portable Daxter Bundle Silver Gaming
Categories	modules\system\blocks\customer\categories_box.php <skin>\customer\cmsboxes\categories.tmpl	 Categories Software (0) Computers & Networking (27) Consumer Electronics (9) Home & Garden (0) Apparel, Shoes, Jewelry (6) Books (1)
Currency	modules\system\blocks\customer\currency_box.php <skin>\customer\cmsboxes\currencies.tmpl	 Currency U.S. Dollar

Custom Products	modules\system\blocks\customer\ custom_products_box.php <skin>\customer\cmsboxes\ custom_products.tmpl	
Dropdown Categories	modules\system\blocks\customer\ dropdown_categories_box.php <skin>\customer\cmsboxes\dropdown_categories dropdown_categories.tmpl dropdown_subcategories.tmpl	 SOFTWARE COMPUTERS & NETWORKING CONSUMER ELE Notebooks Acer Sony Dell Desktops Alienware Dell
Login	modules\system\blocks\customer\ login_box.php <skin>\customer\cmsboxes\ login.tmpl	
Manufacturers	modules\system\blocks\customer\ manufacturers_box.php <skin>\customer\cmsboxes\ manufacturers.tmpl	 Manufacturers Canon DELL HP Logitech Microsoft View all
Menu	modules\system\blocks\customer\ menu_box.php <skin>\customer\cmsboxes\ menu.tmpl	
Mini Cart	modules\system\blocks\customer\ mini_cart.php <skin>\customer\cmsboxes\mini_cart mini_cart.tmpl mini_cart_items.tmpl	 Mini Cart Lexmark C500n $311.95 Kaspersky Anti-Virus 7 $59.00 SUBTOTAL: $370.95 View & Checkout

New Products	modules\system\blocks\customer\ new_products_box.php <skin>\customer\cmsboxes\ new_products.tmpl	 New Products Samsung 2043SN Price: $250.00 Xerox Phaser 3117 Price: $190.00
News	modules\system\news\customer\ news.php <skin>\modules\news\customer news.tmpl news_box.tmpl news_view.tmpl	 News New products 03 Apr 2009 Store Is Open 21 Mar 2009 Subscribe to news View all news
Newsletter	modules\system\newsletter\ newsletter.php <skin>\modules\system\newsletter\customer newsletter_box.tmpl newsletter_subscribe_success.tmpl newsletter_unsubscribe.tmpl newsletter_unsubscribe_success.tmpl	 Newsletter Enter your email to subscribe to Example Newsletter: Subscribe
Page Contents	modules\system\blocks\customer\ controller_commands_box.php	

Product Attributes Filter	modules\system\product_attributes_filter\ product_attributes_filter.php <skin>\customer\cmsboxes\product_attributes filter.tmpl filter_attributes.tmpl	 Find Products By RAM (GB) 1 GB (0) 2 GB (0) 3+ GB (0) CLOTHES COLOUR PRICE $1011 - $2187
Product Comparison	modules\system\product_comparison\ product_comparison.php <skin>\modules\product_comparison product_comparison_box.tmpl product_comparison_details_box.tmpl	 Product Comparison (3) Show comparison HP Pavilion A6230N Desktop PC Price: $879.00 HP Pavilion A6200N Desktop PC Price: $609.00 DELL Dimension 5150 Intel Pentium 4 w/HT Technology Price: $449.00
Recently Viewed	modules\system\blocks\customer\ recently_viewed.php <skin>\customer\cmsboxes\ recently_viewed.tmpl	 Recently Viewed XBOX 360 Halo 3 Special Edition Package Price: $514.95 -2% discounts info
Search	modules\system\blocks\customer\ search_box.php <skin>\customer\cmsboxes\ search.tmpl	
Shopping Cart	modules\system\blocks\customer\ cart_box.php <skin>\customer\cmsboxes\ cart.tmpl	 Shopping Cart Items: 3 items Total: $927.85 View & Checkout

XHTML Box	modules\system\blocks\customer\ cms_html_box.php <skin>\customer\cmsboxes\ cms_html_box.tmpl

Dynamic pages

Page Code	Page Name	Path
com.summercart.dynamic_page.index	Home	index.php
com.summercart.dynamic_page.categories	Categories	categories.php
com.summercart.dynamic_page.product	Product Details	product.php
com.summercart.dynamic_page.cart	View Cart	cart.php
com.summercart.dynamic_page.manufacturers	Manufacturers	manufacturers.php
com.summercart.dynamic_page.manufacturers_list	Manufacturers List	manufacturers.php?list
com.summercart.dynamic_page.search	Search	search.php
com.summercart.dynamic_page.form	Forms	form.php
com.summercart.dynamic_page.login	Login	login.php
com.summercart.dynamic_page.forgot_pass	Forgotten Password	forgot_pass.php
com.summercart.dynamic_page.register	Registration	register.php
com.summercart.dynamic_page.profile	Customer Profile	profile.php
com.summercart.dynamic_page.orders	Orders	orders.php
com.summercart.dynamic_page.view_order	Order Details	view_order.php
com.summercart.dynamic_page.checkout_final.success	Checkout Success	checkout_final.php?success
com.summercart.dynamic_page.checkout_final.pending	Checkout Pending	checkout_final.php?pending
com.summercart.dynamic_page.checkout_final.failure	Checkout Failure	checkout_final.php?failure
com.summercart.dynamic_page.gallery	Gallery	gallery.php
com.summercart.productcomparison	Product Comparison	module.php?ModuleName=com.summercart.productcomparison
com.summercart.newsletter.subscribe	Newsletter Subscribe	module.php?action=subscribe&ModuleName=com.summercart.newsletter

com.summercart.newsletter.unsubscribe	Newsletter Unsubscribe	module.php?action=unsubscribe&ModuleName=com.summercart.newsletter
com.summercart.wishlist	Wishlist	module.php?ModuleName=com.summercart.wishlist
com.summercart.dynamic_page.points	Points	points.php
com.summercart.news	News	module.php?ModuleName=com.summercart.news
com.summercart.news.list	News List	module.php?ModuleName=com.summercart.news&list
com.summercart.dynamic_page.page_not_found	404 Page Not Found	page_not_found.php
com.summercart.dynamic_page.page	Rich Text Page	page.php
com.summercart.dynamic_page.addresses	Customer Addresses	addresses.php
com.summercart.dynamic_page.order_egoods	Order Egoods	order_egoods.php
com.summercart.dynamic_page.error	Error	error.php
com.summercart.ratingactions	Rating Actions	module.php?ModuleName=com.summercart.ratingactions

%% variables

Variable	Description
%%SKINSETTING…%%	Allows you to access your skin settings. See the Skin Settings chapter for further details.
%%SKIN_URL%%	Replaced at runtime with your skin's relative URL. Note that this variable is obsolete, and you should use <mi:skinurl> instead (see next page).
%%CURRENT_LANGUAGE_CODE%%	The code of the language selected by the user
%%HTTPS%%	The base URL of the store using https protocol if the site administrator checked the "Use SSL" option in Settings > Security in the Admin Panel. This is used for pages that have sensitive information such as checkout pages.
%%URL_LOGIN%%	URL of the Login page
%%URL_FORGOT_PASS%%	URL of the Forgot Password page
%%URL_REGISTER%%	URL of the Registration page
%%URL_LOGOUT%%	URL of the Logout page
%%URL_PROFILE%%	URL of the User Profile page
%%URL_ORDERS%%	URL of the User Orders page
%%URL_POINTS%%	URL of the User Points page
%%URL_WISHLIST%%	URL of the User Wish List page
%%URL_ADDRESSES%%	URL of the User Addresses page
%%URL_CHECKOUT%%	URL of the Checkout page
%%URL_VIEW_ORDER%%	URL of the View Order page. This requires Order ID, which is available on pages where the user can select an order, such as the Orders page. For example: %%URL_VIEW_ORDER%%?OrderID=%%ORDERID%%
%%URL_ORDER_EGOODS%%	URL of the E-Goods page. This requires Order ID, which is available on pages where the user has selected an order, such as the Order Details page. For example: %%URL_ORDER_EGOODS%%?OrderID=%%ORDERID%%
%%URL_DOWNLOAD%%	URL to download an E-Good. This requires an E-Good ID, which is available on pages where the user can select an e-good, such as the Order E-Goods page.
%%URL_MANUFACTURERS%%	URL of the Manufacturers page
%%URL_CART%%	URL of the Shopping Cart page

%%URL_SEARCH%%	URL of the Search page
%%URL_INDEX%%	URL of the Index page
%%URL_NEWS%%	URL of the News page
%%URL_PRODUCT%%	URL of the Product Details page. This requires a Product ID, which is available on pages where the user can select a product.
%%URL_SEND_TO_FRIEND%%	URL of the Send To Friend page. This requires a Product ID, which is available on pages where the user can select a product, such as the Products or Product Details page.
%%URL_MODULE%%	URL of the Modules page. The page is used to run a module that you specify by name. For example: %%URL_MODULE%%?ModuleName=com.summercart.rss &amp;UILanguage=%%CURRENT_LANGUAGE_CODE%% &amp;FeedType=News

Skin structure

Folder	Description
/**customer**	Most of the templates for the customer area are located in subfolders of the /customer folder. Itself it contains the page template, header and footer templates, layout area templates, the home page, and a few others.
/customer/**addresses**	Registered users can have multiple addresses that they can maintain through their profile. This folder contains all the templates for listing, adding and editing addresses.
/customer/**categories**	Templates for displaying products filtered by category. There are two separate list and grid templates.
/customer/**checkout**	Templates for each step of the Checkout Wizard, as well as templates for the different results of a checkout – success with processed payment (checkout_success.tmpl), success with pending payment (checkout_pending.tmpl) or declined payment (checkout_failure.tmpl)
/customer/**cmsboxes**	Templates for the built-in boxes.
/customer/**emails**	Templates for all the emails that are sent by the system, such as the ones after successful registration or new order.
/customer/**forgot_pass**	Templates for the Forgotten Password wizard that is typically available from the Login page.
/customer/**forms**	Templates for forms (forms.tmpl) and the confirmation message that is displayed after a form is successfully submitted.
/customer/**images**	Skin images.
/customer/**js**	Javascript files.
/customer/**manufacturers**	Templates for manufacturers list (all_manufacturers.tmpl), and products filtered by manufacturer. There are two separate list and grid templates.
/customer/**menus**	Templates for menus.
/customer/**myaccount**	Templates for pages that only a registered user can access, such as his profile, history of orders, and points. The only exceptions are the templates for the user's addresses, which are in the /addresses folder, and templates for user's wishlist, which are in the /modules/wishlist folder.

/customer/**popup**	Templates for popups such as the Send to Friend popup.
/customer/**product**	Templates for displaying products in a list or a grid.
/customer/**register**	Templates for the registration page (register.tmpl), as well as the page displayed after a successful registration (register_success.tmpl).
/customer/**search**	Templates for displaying search results. Products are displayed in either a list or a grid. The rest of the templates are used for Advanced Search where users can search for a product based on any property or custom attribute of a product.
/customer/**widgets**	Templates for widgets.
/**lang**	Language customizations for skin settings. See the chapter Skin Settings in Multiple Languages for further details.
/**modules**	Templates for modules.
/modules/**checkout**	Templates for the installed checkout modules.
/modules/**extra**	Templates for modules that do not fall into other categories, such as the Facebook Connect module.
/modules/**news**	Templates for the News module: list of news (news_view.tmpl), news details (news.tmpl) and news box (news_box.tmpl)
/modules/**payment**	Templates for the installed payment modules.
/modules/**product_ comparison**	Templates for the Product Comparison module.
/modules/**shipping**	Templates for shipping modules.
/modules/**system**	Templates for all the system (internal) modules, such as the modules for bundled products, newsletters, XML and RSS feeds, product ratings & reviews, related products, product prices and wishlists.