

Customizing osCommerce

A Guide to Safely and Quickly Adding Features all by Yourself

Jared Call
WebCart Consulting

**Full Version Includes
Bonus
Auto-Backup Scripts!**

Chapter 2: Getting Started

Bare essentials

To get started customizing your website, you'll need a few essentials. Here they are:

- **Web Hosting.** You need web hosting. If you don't know what you need, this can be a complicated topic. It can also be very simple. Here are my basic guidelines:
 1. Most websites will be just fine with shared hosting. Shared hosting means that your website will be running on a computer that is shared by other customers. Dedicated hosting is when you are the only website running on that computer. Unless you know for sure that you need dedicated hosting, stick with shared hosting
 2. Read recent customer opinions of hosting companies you're considering. Go to <http://www.webhostingtalk.com> or another community that reviews web hosts for real customer experiences.
 3. Don't rely on web host review sites without opportunity for feedback. Even if they're not simply there to get referral commissions, they don't have nearly the experience and feedback of a site like <http://www.webhostingtalk.com>. The most valuable information about a web host will be how their customers feel about them.
 4. Get a host that offers “dedicated SSL certificates”. Yes, you can get by for less money by using a shared SSL certificate, but it's far more professional to have someone see your site using <https://www.yourwebsite.com/checkout.php> than <https://ssl32.myhostingcompany.net/~joeuser/checkout.php>. There are other reasons for dedicated SSL, but a more professional-looking URL is as good a reason as any.
 5. If it seems like they charge too much, shop around to see what their competitors are charging.
 6. Choose a hosting company that offers a 30-day guarantee. It gives them incentive to keep you as a customer.
- **Which version of osCommerce?** Whether you installed it yourself or from your web hosting control panel, you should know which version of osCommerce you have installed. You can always get the most current version from <http://www.oscommerce.com>, but know which version you have. More details on this shortly.
- **FTP client.** Internet Explorer has one built-in. In my opinion, it stinks. Many web hosting companies offer file management utilities in their control panels. In my opinion, these are really not much better. You need an FTP client. Though there are many good ones, I recommend FileZilla, as noted in the Tools section of this guide.
- **Text editor.** You can use notepad or wordpad or some other basic text editor if you like. I like Crimson Editor (free), as noted in the Tools section. PHP Designer is also popular and free.
- **Patience.** You won't get it all right the first time. You may not even understand it the first time. Be patient, ask descriptive questions on the forums, and stick with it.

That's it. With a web hosting account, SSL, an FTP client, and a text editor, you can now edit your website. You have all the tools you need, but it might help to have an idea of how the pieces fit together. That's where the Architecture section comes in.

Architecture

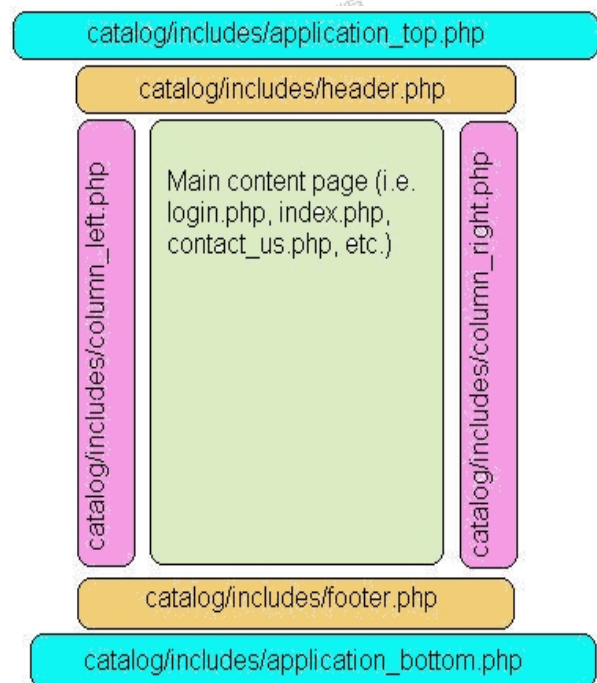
If you've already looked at some of osCommerce's PHP files or directory structure, you may have noticed that there are several pieces to each osCommerce page. Each page is composed of 7 basic structural components and some other supporting files. The structural components are:

- catalog/includes/application_top.php
- catalog/includes/header.php
- catalog/includes/column_left.php
- The main content page; i.e. index.php, contact_us.php, checkout_shipping.php, etc.
- catalog/includes/column_right.php
- catalog/includes/footer.php
- catalog/includes/application_bottom.php

They fit together something like this:

This same basic model applies to the osCommerce Admin area as well, with the exception that there is generally no column_right.php

Note that each main content page also uses catalog/stylesheets.css to establish fonts, colors, etc.



There is one piece that's not shown on the graphic. I left it out on purpose so that you'd be more likely to write it down and remember it.

The piece is catalog/stylesheets.css. The CSS Style sheet defines font sizes, font faces, backgrounds, etc, and it is referenced on every page.

There are other supporting files that you don't see very often. Here are the most important ones:

- class files: Class files, contained in catalog/includes/classes/, define “types” of things. In other words, they specify that an “order” consists of products, an order status, a purchase date, shipping address, etc.
- function files: Function files, in catalog/includes/functions/, define special functions. For example, there is one file of functions to perform database operations. Another file provides

- functions to keep track of which customers are logged in right now.
- language files: Instead of using phrases and labels in just one language, osCommerce uses special text definition markers in files like index.php or login.php. Those special text definition markers are then defined in the language files.

Here is an example of how the language files work: Open catalog/login.php. Near the top, you can see the following line:

```
require(DIR_WS_LANGUAGES . $language . '/' . FILENAME_LOGIN);
```

That line is where osCommerce loads the language definition file. We'll come back to this. For now, scroll further down the login.php file. Find this line:

```
<td class="pageHeading"><?php echo HEADING_TITLE; ?></td>
```

If you were to access this page in your browser, and to a “view source”, you'd see this:

```
<td class="pageHeading">Login</td>
```

What happened? While running login.php, osCommerce saw that it was supposed to get the language file. The language file for login.php is catalog/includes/languages/english/login.php. In that file, osCommerce read that for login.php, HEADING_TITLE really means “Login.” With other languages such as Spanish or German, osCommerce would look in catalog/includes/languages/spanish/login.php and so on.

So, with the help of a little PHP and including the appropriate language file, we went from this:

```
<td class="pageHeading"><?php echo HEADING_TITLE; ?></td>
```

to this:

```
<td class="pageHeading">Login</td>.
```

Hopefully the diagram and this brief explanation has helped clarify where each piece fits into the osCommerce whole.

**Don't miss out on all the rest of the great
info in the full eBook.**

Buy Now!