WooCommerce Cookbook

Create, design, and manage your own personalized online store with WooCommerce, the fastest growing e-commerce platform
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'WooCommerce Basics'
- A synopsis of the book’s content
- More information on WooCommerce Cookbook

About the Author

Patrick Rauland has built custom e-commerce solutions as well as high-end WooCommerce sites while working for advertising agencies. He joined the support team at WooThemes and answered thousands of support tickets. He later joined the WooThemes development team and started writing the underlying code for WooCommerce. After several months on the development team, he took on the role of product manager, and interacts with customers and the development team to help plan the future of WooCommerce.

Patrick loves going to conferences to talk about e-commerce, software development, productivity, happiness, and creating your online presence.

Patrick lives in Denver, Colorado, and loves running, yoga, and hiking.

You can reach Patrick through his programming blog at http://speakinginbytes.com.
WooCommerce Cookbook

Up until just a few years ago, building your own e-commerce site was remarkably difficult. You had to install complex software, host it yourself, and hope that, whenever you made a change, nothing broke. That's very different from where we are now.

WooCommerce was released in September 2011 and, since then, it's taken the e-commerce world by storm. As of 2015, there are nearly 400,000 sites that run WooCommerce; that is approximately 18 percent of all e-commerce sites on the Internet and bigger than every other competitor by a wide margin (source: BuiltWith.com).

Part of the success of WooCommerce is because it's built on top of the incredibly popular WordPress platform. With just a few clicks, you can install the WooCommerce plugin and turn your WordPress site into an online store. With over 300 official extensions to customize your store, you no longer need a developer. You can add thousands of free plugins and themes to customize the rest of your WordPress site and the options are nearly limitless.

There are hosting companies that only host WordPress sites, services that maintain your WordPress sites for you, and professional products that handle every aspect of your site from creating contact forms to image optimization. The list goes on. What you need to know is that WordPress is huge and still growing. As WordPress grows, so will WooCommerce.

WooCommerce isn't only built on top of WordPress. It's also built on top of the WooThemes brand. WooThemes was one of the first companies to start building commercial themes and plugins for WordPress, and over the years they've perfected their techniques and poured it all into WooCommerce. Having to swap your e-commerce software would be a nightmare. That's part of the reason that WooCommerce has taken off so fast. It's backed by one of the biggest brands in the entire WordPress space. People know that they won't disappear any time soon.

In short, you can rest easy. WooCommerce is built on top of both a growing platform and a trusted brand.

Let's get started building your e-commerce site.
What This Book Covers

Chapter 1, *WooCommerce Basics*, covers how to install the software and configure the basic settings.

Chapter 2, *Adding Products*, deals with adding simple products to the store, downloadable files to a product, images and image galleries to a product and, variable products to the store; this chapter also covers bulk-uploading products.

Chapter 3, *Changing the Product Organization*, covers configuring your store to show off your products, adding more products on the shop page, adding a search capability, adding social media, sorting products, and adding extra information to the shop page.

Chapter 4, *Running a Membership Site*, deals with creating subscription products, using members-only pricing, creating premium content, and creating a pricing table.

Chapter 5, *Setting Up Shipping Methods*, deals with configuring Free Shipping and Flat Rate Shipping, getting live shipping quotes, tracking shipments, and creating advanced tables of shipping rates.

Chapter 6, *Getting Paid*, covers how to configure payment gateways such as PayPal, Stripe, and Simplify Commerce. It also covers how to configure HTTPS and adding e-commerce tracking with Google Analytics.

Chapter 7, *Modifying the Checkout Process*, deals with configuring terms and conditions, adding customers to your newsletter, adding and removing checkout fields, adding a one-page checkout, and adding default options to the checkout.

Chapter 8, *Managing Orders and Taxes*, covers refunding orders, importing existing orders, configuring order numbers, exporting order information, and configuring taxes.

Chapter 9, *WooCommerce Theming*, deals with adding a cart to the menu, using WooCommerce hooks, overriding WooCommerce templates and WooCommerce CSS, and creating product slide shows.

Chapter 10, *Exploring More with WooCommerce*, deals with enabling reviews, creating and cloaking affiliate links, creating coupons, and sending follow-up e-mails.
Chapter 1

WooCommerce Basics

In this chapter, we will cover:

- Installing WooCommerce
- Setting your store location and currency
- Finding documentation on WooThemes.com
- Installing WooCommerce plugins
- Installing official WooThemes plugins
- Manually creating WooCommerce pages
- Creating a WooCommerce plugin
- Adding a currency to WooCommerce

Introduction

A few years ago, building an online store used to be an incredibly complex task. You had to install bulky software onto your own website and pay expensive developers a significant sum of money to customize even the simplest elements of your store. Luckily, nowadays, adding e-commerce functionality to your existing WordPress-powered website can be done by installing a single plugin. In this chapter, we'll go over the settings that you'll need to configure before launching your online store with WooCommerce. Most of the recipes in this chapter are simple to execute. We do, however, add a relatively complex recipe near the end of the chapter to show you how to create a plugin specifically for WooCommerce. If you're going to be customizing WooCommerce with code, it's definitely worth looking at that recipe to know the best way to customize WooCommerce without affecting other parts of your site.
WooCommerce Basics

The recipes in this chapter form the very basics of setting up a store, installing plugins that enhance WooCommerce, and managing those plugins. There are recipes for official WooCommerce plugins written using WooThemes as well as a recipe for unofficial plugins. Feel free to select either one. In general, the official plugins are better supported, more up to date, and have more functionality than unofficial plugins. You could always try an unofficial plugin to see whether it meets your needs, and if it doesn't, then use an official plugin that is much more likely to meet your needs.

At the end of this chapter, your store will be fully functional and ready to display products. In Chapter 2, Adding Products, we will learn how to add products to our store.

Installing WooCommerce

WooCommerce is a WordPress plugin, which means that you need to have WordPress running on your own server to add WooCommerce. The first step is to install WooCommerce. You could do this on an established website or a brand new website—it doesn't matter. Since e-commerce is more complex than your average plugin, there's more to the installation process than just installing the plugin.

Getting ready

Make sure you have the permissions necessary to install plugins on your WordPress site. The easiest way to have the correct permissions is to make sure your account on your WordPress site has the admin role.

How to do it...

There are two parts to this recipe. The first part is installing the plugin and the second step is adding the required pages to the site. Let's have a look at the following steps for further clarity:

1. Log in to your WordPress site.
2. Click on the Plugins menu.
3. Click on the Add New menu item. These steps have been demonstrated visually in the following screenshot:
4. Search for WooCommerce.

5. Click on the Install Now button, as shown in the following screenshot:
6. Once the plugin has been installed, click on the **Activate Plugin** button. You now have WooCommerce activated on your site, which means we're half way there. E-commerce platforms need to have certain pages (such as a cart page, a checkout page, an account page, and so on) to function. We need to add those to your site.

7. Click on the **Install WooCommerce Pages** button, which appears after you've activated WooCommerce. This is demonstrated in the following screenshot:

![Dashboard and Plugins page](image.png)

**How it works...**

WordPress has an infrastructure that allows any WordPress site to install a plugin hosted on WordPress.org. This is a secure process that is managed by WordPress.org.

Installing the WooCommerce pages allows all of the e-commerce functionality to run. Without installing the pages, WooCommerce won't know which page is the cart page or the checkout page. Once these pages are set up, we're ready to have a basic store up and running.

If WordPress prompts you for FTP credentials when installing the plugin, that's likely to be a permissions issue with your web host. It is a huge pain if you have to enter FTP credentials every time you want to install or update a plugin, and it's something you should take care of. You can send this link to your web host provider so they know how to change their permissions. You can refer to [http://www.chrisabernethy.com/why-wordpress-asks-connection-info/](http://www.chrisabernethy.com/why-wordpress-asks-connection-info/) for more information to resolve this WordPress issue.

**Setting your store location and currency**

WooCommerce is an e-commerce platform that can work anywhere in the world. You can enable it to use any location or currency. WooCommerce was originally created by developers in the UK, so the default settings are all UK-based. If you don't live in the UK, you'll have to change your store location and currency.
Getting ready

You'll need a WordPress site with WooCommerce installed.

How to do it...

Since you're looking for ways to customize your store functionality, you can do a lot using the WooCommerce settings pages. Before looking through code or asking for a developer's help, it's worth looking through all of the settings pages to see whether any issue you have can be fixed right there. Changing the store location and currency can both be done on the settings pages with a couple of clicks, by performing the following steps:

1. Go to WooCommerce | Settings:
2. Click on the **Base Location** setting and start typing the name of your province or country. Many countries, such as the United States, have different provinces, so it's best to see whether your province is listed in **WooCommerce**. If it is, select the province; if not, select your country. Have a look at the following screenshot:

3. Scroll down to the **Currency Options** section.

4. From the **Currency** dropdown, select your currency.

**How it works...**

WooCommerce has a huge list of countries and currencies built in. You can select the right one on the **Settings** screen and WooCommerce will save this information in the database and use it for shipping, taxes, and other purposes. You can programmatically add more countries and currencies to WooCommerce, and those updated lists will show up on the settings pages.

If your currency doesn't appear in the list, it can be manually added. You may also refer to the *Adding a currency to WooCommerce* recipe at the end of this chapter.
Finding documentation on WooThemes.com

WooCommerce has been developed by WooThemes, a WordPress theme and plugin development shop. In addition to developing the plugin, they’ve also created hundreds of pages of documentation for WooCommerce and all of the extensions.

If you want to dig into exactly what settings are available and how they interact with other settings, you should know how to find the documentation in the first place. Some people like to read through the documentation ahead of time and others like to explore the settings without reading the documentation and only refer to it if they are stuck. Either way works.

How to do it...

The documentation can be accessed directly via the WooThemes website. It can also be accessed from your WordPress site. Follow these steps to find the documentation:

1. In the WordPress admin, click on the Plugins menu to see the list of installed plugins.
2. Click on the Docs link under WooCommerce, as shown in the following screenshot, which will take you to the documentation on http://www.woothemes.com:

<table>
<thead>
<tr>
<th>Plugins</th>
<th>Add New</th>
</tr>
</thead>
<tbody>
<tr>
<td>All (1)</td>
<td>Active (1)</td>
</tr>
<tr>
<td>Bulk Actions</td>
<td>Apply</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plugin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WooCommerce</td>
<td>An e-commerce toolkit that helps you sell anything. Beautifully. Version 2.3</td>
</tr>
</tbody>
</table>

| Settings | Deactivate | Edit |

Version 2.3 | By WooThemes | View details | Docs | API Docs | Premium Support
3. Now that you're in the documentation section, navigate to the section you need. **Getting Started** is great for someone just starting their store, **Codex** is great for people who want to customize **WooCommerce** or their theme, and **Extensions** is great for any WooCommerce extensions you may have.

Installing **WooCommerce plugins**

WooCommerce is open source and free, which means that any developer can add extra functionality to WooCommerce. Some developers may add this extra functionality with a WordPress plugin and then release the plugin either for free or as a premium plugin.

There are three likely places to get a WooCommerce plugin: from WooThemes, from WordPress.org, or from a third-party website. In this recipe, we'll be installing a plugin obtained from a third-party site. Installing a WooCommerce plugin from WordPress.org is similar to installing WooCommerce. We will have a look at how to install official WooThemes plugins in the next recipe.
Getting ready

Make sure you have the necessary permissions to install plugins on your WordPress site and that you have WooCommerce installed.

How to do it...

We're going to be downloading a plugin I created as an example of how to create an extra WooCommerce settings page, by performing the following steps:

1. Go to https://gist.github.com/BFTrick/b5e3afa6f4f83ba2e54a/.
2. Click on Download Gist.
3. Make sure your plugin is in the .zip format. By looking at the downloaded file, we can see that our file ends in .tar.gz, which is not the right format:
We need to convert this to the .zip format. Here's what you need to do on a Mac operating system:

1. Double-click on the .tar.gz file, which will create a new folder.

2. Right-click on the new folder and click on Compress gistb5e3afa6f4f83ba2...9.

And this is what you need to do on a Windows operating system:

3. Windows cannot extract a .tar.gz file natively. You'll have to install a software to do so. A free tool is 7-Zip, which can be found at http://www.7-zip.org/.

4. In your WordPress admin, click on the Plugins menu and then on Add New:

5. Click on Upload Plugin:
6. On the new screen, choose the zipped file to be uploaded and click on **Install Now**.

7. Once the plugin has finished installing, click on the **Activate Plugin** link. The following screen should appear:

```
Installing Plugin from uploaded file: gistb5e3afa6f4f83ba2e54a-bb6dc1e76c9aee652bc2cd0925803b1df1cf3da9.zip

Unpacking the package...
Installing the plugin...
Plugin installed successfully.
Activate Plugin | Return to Plugins page
```

You now have a WooCommerce plugin installed on your site. If you want to see what this plugin does, you can navigate to the WooCommerce settings page and click on **Settings Demo Tab**.

**How it works...**

WordPress can import any ZIP file. As long as the plugin was created correctly, WordPress shouldn't have any problems running the plugin. If there is a problem, WordPress will let you know that the plugin has invalid code.

**Installing official WooThemes plugins**

WooThemes doesn't just create the WooCommerce plugin. They also create standalone plugins and hundreds of extensions that add extra functionality to WooCommerce. The beauty of this system is that WooCommerce is very easy to use because users only add extra complexity when they need it. If you only need simple shipping options, you don't ever have to see the complex shipping settings.

On the WooThemes website, you may browse for WooCommerce extensions, purchase them, and download and install them on your site. WooThemes has made the whole process very easy to maintain. They have built an updater similar to the one in WordPress, which, once configured, will allow a user to update a plugin with one click instead of having to through the whole plugin upload process again.
WooCommerce Basics

Getting ready

Make sure you have the necessary permissions to install plugins on your WordPress site. You also need to have a WooThemes product. There are several free WooThemes products including Pay with Amazon which you can find at http://www.woothemes.com/products/pay-with-amazon/.

How to do it...

There are two parts to this recipe. The first part is installing the plugin and the second step is adding your license for future updates. Follow these steps:

2. Click on the Downloads menu:

![Downloads menu](image)

3. Find the product you wish to download and click on the Download link for the product. You will see that you get a ZIP file.

![Available downloads](image)

4. On your WordPress site, go the Plugins menu and click on Add New.
5. Click on Upload Plugin.
6. Select the file you just downloaded and click on the Install Now button.
7. After the plugin has finished installing, click on the **Activate Plugin** link.

You now have WooCommerce as well as a WooCommerce extension activated on your site. They're both functioning and will continue to function. You will, however, want to perform a few more steps to make sure it's easy to update your extensions:

1. Once you have an extension activated on your site, you'll see a link in the WordPress admin: **Install the WooThemes Updater plugin**. Click on that link:

   ![Plugin Link](image)

   - The updater will be installed automatically.
   - Once it is installed, you need to activate the updater.
   - After activation, you'll see a new link in the WordPress admin: **activate your product licenses**. Click that link to go straight to the page where you can enter your licenses. You could also navigate to that page manually by going to Dashboard | WooThemes Helper from the menu.

   ![Plugin License](image)

2. Keep your WordPress site open in one tab and log back in to your WooThemes account in another browser tab.
6. On the WooThemes browser tab, go to **My Licenses** and you’ll see a list of your products with a license key under the heading **KEY**:

<table>
<thead>
<tr>
<th>PRODUCT</th>
<th>KEY</th>
<th>TYPE</th>
<th>EXPIRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stripe</td>
<td>WOO-a531...</td>
<td>Single site licence</td>
<td>October 20, 2015</td>
</tr>
</tbody>
</table>

You need to Download the WooThemes Helper plugin to add your license keys. Please see the documentation for more information.

7. Copy the key, go back to your WordPress site, and enter it in the **Licenses** field.

8. Click on the **Activate Products** button at the bottom of the page. The activation process can take a few seconds to complete.

9. If you've successfully put in your key, you should see a message at the top of the screen saying so.
How it works...

A plugin that's not hosted on WordPress.org can't update without someone manually reuploading it. The WooThemes updater was built to make this process easier so you can press the update button and have your website do all the heavy lifting.

Some websites sell official WooCommerce plugins without a license key. These sales aren't licensed and you won't be getting updates, bug fixes, or access to the support desk. With a regular website, it's important to stay up to date. However, with e-commerce, it's even more important since you'll be handling very sensitive payment information. That's why I wouldn't ever recommend using a plugin that can't update.

Refer to the preceding recipe, Installing WooCommerce plugins, for more details. There are several screenshots in that recipe about the general process of uploading a WordPress plugin.

Manually creating WooCommerce pages

Every e-commerce platform will need to have some way of creating extra pages for e-commerce functionality, such as a cart page, a checkout page, an account page, and so on. WooCommerce prompts to help you create these pages for you when you first install the plugin. So if you installed it correctly, you shouldn't have to do this. But if you were trying multiple e-commerce systems and for some reason deleted some pages, you may have to recreate those pages.

How to do it...

There's a very useful Tools menu in WooCommerce. It's a bit hard to find since you won't be needing it everyday, but it has some pretty useful tools if you ever need to do some troubleshooting. One of these tools is the one that allows you to recreate your WooCommerce pages. Let's have a look at how to use that tool:

1. Log in to the WordPress admin.
2. Click on **WooCommerce | System Status:**

3. Click on **Tools:**

```
System Status  Tools  Logs

Please copy and paste this information in your ticket when contacting support:

Get System Report
```

4. Click on the **Install Pages** button:
WooCommerce keeps track of which pages run e-commerce functionality. When you click on the *Install Pages* button, it checks which pages exist and if they don't exist, it will automatically create them for you. You could create them by creating new WordPress pages and then manually assigning each page with specific e-commerce functionality. You may want to do this if you already have a cart page and don't want to recreate a new cart page but just copy the content from the old page to the new page. All you want to do is tell WooCommerce which page should have the cart functionality. Let's have a look at the following manual settings:

- The *Cart, Checkout, and Terms & Conditions* page can all be set by going to *WooCommerce | Settings | Checkout*
- The *My Account* page can be set by going to *WooCommerce | Settings | Accounts*
There's more...

You can manually set some pages, such as the Cart and Checkout page, but you can't set subpages. WooCommerce uses a WordPress functionality called endpoints to create these subpages. Pages such as the Order Received page, which is displayed right after payment, can't be manually created. These endpoints are created on the fly based on the parent page. The Order Received page is part of the checkout process, so it's based on the Checkout page. Any content on the Checkout page will appear on both the Checkout page and on the Order Received page.

You can't add content to the parent page without it affecting the subpage, but you can change the subpage URLs. The checkout endpoints can be configured by going to WooCommerce | Settings | Checkout | Checkout Endpoints.

Creating a WooCommerce plugin

Unlike a lot of hosted e-commerce solutions, WooCommerce is entirely customizable. That's one of the huge advantages for anyone who builds on open source software. If you don't like it, you can change it. At some point, you'll probably want to change something that's not on a settings page, and that's when you may want to dig into the code. Even if you don't know how to code, you may want to look this over so that when you work with a developer, you would know they're doing it the right way.

Getting ready

In addition to having admin access to a WordPress site, you'll also need FTP credentials so you can upload a plugin. You'll also need a text editor. Popular code editors include Sublime Text, Coda, Dreamweaver, and Atom. I personally use Atom. You could also use Notepad on a Windows machine or Text Edit on a Mac in a pinch.

How to do it...

We're going to be creating a plugin that interacts with WooCommerce. It will take the existing WooCommerce functionality and change it. These are the WooCommerce basics. If you build a plugin like this correctly, when WooCommerce isn't active, it won't do anything at all and won't slow down your website. Let's create a plugin by performing the following steps:

1. Open your text editor and create a new file. Save the file as woocommerce-demo-plugin.php.
2. In that file, add the opening PHP tag, which looks like this: <?php.
3. On the next line, add a plugin header. This allows WordPress to recognize the file as a plugin so that it can be activated. It looks something like the following:

```php
/**
 * Plugin Name: WooCommerce Demo Plugin
 * Plugin URI: https://gist.github.com/BFTrick/3ab411e7cec43eff9769
 * Description: A WooCommerce demo plugin
 * Author: Patrick Rauland
 * Author URI: http://speakinginbytes.com/
 * Version: 1.0
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 *
 */
```

4. Now that WordPress knows that your file is a plugin, it's time to add some functionality to this. The first thing a good developer does is makes sure their plugin won't conflict with another plugin. To do that, we make sure an existing class doesn't have the same name as our class. I’ll be using the `WC_Demo_Plugin` class, but you can use any class name you want. Add the following code beneath the plugin header:

```php
if ( class_exists( 'WC_Demo_Plugin' ) ) {
    return;
}

class WC_Demo_Plugin {
}
```
5. Our class doesn't do anything yet, but at least we've written it in such a way that it won't break another plugin. There's another good practice we should add to our plugin before we add the functionality, and that's some logic to make sure another plugin won't misuse our plugin. In the vast majority of use cases, you want to make sure there can't be two instances of your code running. In computer science, this is called the **Singleton pattern**. This can be controlled by tracking the instances of the plugin with a variable. Right after the `WC_Demo_Plugin {` line, add the following:

```php
protected static $instance = null;

/**
 * Return an instance of this class.
 * @return object A single instance of this class.
 * @since 1.0
 */
public static function get_instance() {
    // If the single instance hasn't been set, set it now.
    if ( null == self::$instance ) {
        self::$instance = new self;
    }

    return self::$instance;
}
```

And get the plugin started by adding this right before the `endif;` line:

```php
add_action( 'plugins_loaded', array( 'WC_Demo_Plugin', 'get_instance' ), 0 );
```

6. At this point, we've made sure our plugin doesn't break other plugins and we've also dummy-proofed our own plugin so that we or other developers don't misuse it. Let's add just a bit more logic so that we don't run our logic unless WooCommerce is already loaded. This will make sure that we don't accidentally break something if we turn WooCommerce off temporarily. Right after the `protected static $instance = null;` line, add the following:

```php
/**
 * Initialize the plugin.
 *
 * @since 1.0
 */
private function __construct() {
```
7. And now our plugin only runs when WooCommerce is loaded. I'm guessing that at this point, you finally want it to do something, right? After we make sure WooCommerce is running, let's add some functionality. Right after the if line, add the following code so that we add an admin notice:

```php
// print an admin notice to the screen.
add_action( 'admin_notices', array( $this, 'my_admin_notice' ) );
```

This code will call a method called `my_admin_notice`, but we haven't written that yet, so it's not doing anything. Let's write that method. Have a look at the __construct method, which should now look like this:

```php
/**
 * Initialize the plugin.
 * @since 1.0
 */
private function __construct() {
    if ( class_exists( 'WooCommerce' ) ) {
        // print an admin notice to the screen.
        add_action( 'admin_notices', array( $this, 'display_admin_notice' ) );
    }
}
```

Add the following after the preceding __construct method:

```php
/**
 * Print an admin notice
 * @since 1.0
 */
public function display_admin_notice() {
    ?
    <div class="updated">
```
This will print an admin notice on every single admin page. This notice includes all the messages you typically see in the WordPress admin. You could replace this admin notice method with just about any other hook in WooCommerce to provide additional customizations in other areas of WooCommerce, whether it be for shipping, the product page, the checkout process, or any other area. This plugin is the easiest way to get started with WooCommerce customizations.

If you'd like to see the full code sample, you can see it at https://gist.github.com/BFTrick/3ab411e7cec43eff9769.

8. Now that the plugin is complete, you need to upload it to your plugins folder. You can do this via the WordPress admin or more commonly via FTP.

9. Once the plugin has been uploaded to your site, you'll need to activate the plugin just like any other WordPress plugin. The end result is a notice in the WordPress admin letting us know we did everything successfully.
Whenever possible, use object-oriented code. That means using objects (like the WC_Demo_Plugin class) to encapsulate your code. It will prevent a lot of naming conflicts down the road. If you see some procedural code online, you can usually convert it to object-oriented code pretty easily. Object-oriented programming is out of the scope of this book, but you can read more at http://codex.wordpress.org/Plugin_API.

Adding a currency to WooCommerce

WooCommerce has dozens of currencies already included in the plugin, including but not limited to US dollars, euros, British pounds, New Zealand dollars, Russian rubles, South African rand, Egyptian pounds, and the Mexican peso. If you are planning on using any of these currencies, you don't need to add your own currency and can skip this recipe. If you can't find your currency under WooCommerce | Settings | Currency, then you should follow the steps that follow to add it in.

Getting ready

We'll be writing some code to add your custom currency. You'll need a text editor to write the code and an FTP program to upload it to your site once we're done.

How to do it...

We're going to write a very small snippet and add it to WooCommerce. Once it's added, we'll select the currency from the Currency dropdown in the WooCommerce settings pages. The following are the steps that illustrate the process of creating custom currencies:

1. Open up your theme's functions.php file, located at wp-content/themes/your-theme-name/functions.php, with your text editor.
2. At the bottom of the file, we'll need to add two filters to add this currency to WooCommerce. These filters will accept an array with the existing currencies and add another option to the array. The first filter will add the currency to WooCommerce. The second will add the currency symbol. Have a look at the code for these filters:

```php
add_filter( 'woocommerce_currencies', 'add_patricks_currency' );
add_filter( 'woocommerce_currency_symbol', 'add_patricks_currency_symbol', 10, 2 );
```
3. Now let’s write the first one, which adds the currency itself. You’ll need the name of the currency, which you can append in place of the term Patrick’s Currency in the snippet that follows. You’ll also need the three-character ISO code of the currency, which can be found at [http://en.wikipedia.org/wiki/ISO_4217](http://en.wikipedia.org/wiki/ISO_4217). This can replace PC in the following code snippet:

```php
function add_patricks_currency( $currencies ) {
    $currencies['PC'] = __( 'Patrick’s Currency', 'your-theme-name' );
    return $currencies;
}
```

4. We’re halfway there. Now we need to add the currency symbol. You’ll obviously need the currency symbol, which you can use to replace the dollar sign in the snippet that follows. You’ll need to use the same currency ISO code you used in the preceding snippet. Let’s have a look at the code:

```php
function add_patricks_currency_symbol( $currency_symbol, $currency ) {
    switch( $currency ) {
        case 'PC': $currency_symbol = 'PC'; break;
    }
    return $currency_symbol;
}
```

At this point, WooCommerce should know both your currency and your currency symbol, so it’s worth uploading it and making sure we did it right. Upload the file via FTP.

Navigate to the Currency settings by going to [WooCommerce](https://example.com/woocommerce) | Settings. You should see your new currency in the Currency dropdown:

![Currency settings](image-url)
How it works...

WooCommerce is filled with hooks—places where developers can add or modify the existing code. WooCommerce was smartly built so that anyone can use these hooks to add (or remove) any number of currencies. In this case, we used a filter which is a special type of hook to change the value of something. We changed the contents in an array.

In future recipes, we'll be using actions, which allow completely new programming to fire, and not just changing the value of something.

See also

- We added this snippet to the theme's functions.php file. That works, but it's not as bulletproof as putting it in its own WooCommerce plugin. Refer to the preceding recipe, Creating a WooCommerce plugin, for more details.
Where to buy this book

You can buy WooCommerce Cookbook from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.