

The Simple E-Commerce Solution

# **User Guide**

Your guide to using the phpCart E-Commerce System.

phpCart Version: 4.10

Date: 11/1/13



# **Table of Contents**

1	INTR	ODUCTION	5
	1.1 IN	ITRODUCTION	5
	1.2 TH	HEORY OF OPERATION	<i>6</i>
	1.2.1	Cart page	<i>C</i>
	1.2.2	Checkout page	&
	1.2.3	Confirmation page	
	1.3 SY	YSTEM REQUIREMENTS	14
2	INSTA	ALLATION	15
	2.1 IN	ISTALLATION	15
	2.2 LI	ICENSE FILE	17
3	CONF	FIGURATION	19
	3.1 Lo	OG INTO THE ADMINISTRATION CONTROL PANEL	19
	3.2 CI	HANGE ADMIN PASSWORD	20
	3.3 PH	PPCART CONFIGURATION	20
	3.3.1	Company Setup	21
	3.3.2	Cart Setup	24
	3.3.3	Email Setup	29
	3.3.4	Layout Setup	
	3.3.5	, 1	
	3.3.6	Payment Setup	
	3.3.7	Shipping Setup	36
	3.3.8	Cart Tax Setup	
	3.3.9		
	3.3.10	0 Country Tax Setup	
	3.3.11	I Required Items Setup	
	3.3.12		
		OUPONS	
	3.5 O	RDERS	46
4	INITI	AL SYSTEM TEST	48
5	THE	TEMPLATE SYSTEM	49
	5.1 TH	HE STANDARD TEMPLATE	49



	5.2	CREATING YOUR OWN TEMPLATE	49
	5.3	CART TEMPLATES	51
	5.4	EMAIL TEMPLATES	51
	5.5	GATEWAY TEMPLATES	51
	5.6	THANK YOU TEMPLATES	52
	5.7	MINICART TEMPLATE	52
6	PR	REPARING YOUR STORE	53
	6.1	DEFINING PRODUCTS	53
	6.	1.1 Required Product Variables	54
	6.	1.2 Optional Product Variables	54
	6.2	Pricing	57
	6.3	OPTIONS	58
	6	3.1 Select (drop down) Box Option	59
	6	3.2 Check Box Option	60
	6	3.3 Radio Button Option	60
	6	3.4 Text Box Option	61
	6	3.5 Making an Option Mandatory	61
	6.4	Shipping	62
	6.5	TAXES	63
	6	5.1 Tax Tables	64
	6	5.2 Pre-defined Tax Amount	65
	6.6	Extra Variables	65
7	CU	JSTOMIZING PHPCART	67
	7.1	ALLOWING CUSTOMERS TO ADD MULTIPLE PRODUCTS TO CART	67
	7.2	Adding Variables to the Billing Form	69
	7.3	COUPONS	70
	7.4	Shipping	71
	7.5	TAX	72
	7.6	VALIDATION	72
	7.7	MISC.	72
	7.8	ADVANCED PHP USERS	75
	7.8	8.1 Data Arrays	
	7.8	8.2 Sessions	
	7.8	8.3 Outputting Arrays to a Database	76



8 AI	DDITIONAL CART FEATURES	77
8.1	USING SSL CERTIFICATES FOR SECURING YOUR CART	77
8.2	MAINTENANCE MODE (STORE CLOSED)	78
8.3	CREATING PRODUCT DEFINITIONS USING THE BUTTON MAKER TOOL	79
8.4	GEO IP	83
8.5	Internationalization	83
8.6	DISPLAY THE MINI-CART ON OTHER PAGES	84
8.7	2013 CANADIAN TAXES	86
9 FF	REQUENTLY ASKED QUESTIONS	88
9.1	COMMON INSTALLATION PROBLEMS	88
9.2	HOW TO UPDATE PHPCART TO THE NEWEST VERSION	88
9.3	COMMON USAGE QUESTIONS	90
9.4	HOW TO GET HELP?	91
10 T	ROUBLESHOOTING TECHNIQUES	93
10.1	USING PHPINFO() TO DETERMINE YOUR SERVER'S VERSION OF PHP AND ITS ENCODERS	93
10.2	Encoding Problems	94
10.3	LICENSING PROBLEMS	95
10.4	TEMPLATE PROBLEMS	97
11 0	ODVDICHT NOTICE	00



# 1 Introduction

### 1.1 Introduction

phpCart is the ideal shopping cart solution for merchants who wish to bring their business online with a full-featured shopping cart. phpCart is very easy to install and operate.

Furthermore, phpCart does not require a database, thus reducing the fuss and complexity normally associated with other shopping cart solutions.

phpCart is a back-end solution designed to work with any web site. You build your own product pages and just add standard html links or html forms to connect to the cart.

phpCart has a very powerful yet simple template system that gives you the flexibility to create the exact look you want for your shopping cart. The template system can quickly and easily make the shopping cart look like your web site. Setting up the main template is very easy and takes about five minutes. Once complete, phpCart takes on the look of your existing website. Alternatively, you can leave the templates as they are provided and run your e-commerce site immediately.

In order to protect the phpCart software, several of the phpCart files are encoded with either the Zend or loncube Encoder. In order to run the phpCart system, your server must have either the Zend Optimizer or the loncube modules installed. Most php web hosting servers already have these modules installed.



Only a few of the key operational files are encoded. The remaining files in the phpCart software are not encoded. This gives you the flexibility to modify key areas of phpCart to suite your needs. The gateway modules, coupon system, shipping system and tax system are not encoded. And, of course, none of the templates are encoded so you can make phpCart look just the way you want.

## 1.2 Theory of Operation

The phpCart system is very simple to use. It starts with web pages that you create to display your products and "Buy Now" buttons. The "Buy Now" buttons can be created manually using the product definition (below) or the automated 'Button Maker'. When a prospective customer visits your web site, they will see all the items that you have available for sale.

The phpCart software is first activated when your customer selects one of your products by pressing the "Buy Now" button. At that time, they are taken to the first of three cart pages.

## 1.2.1 Cart page

The first page is the 'Cart' page and shows the customer the item that they have selected.





They will see the following details about this item:

- Product ID
- Product Name
- Price
- Quantity
- Total

In addition, they will see a summary of the cart totals:

- Sub Total Price
- Shipping
- Taxes
- Total

#### The customer will then be able to:

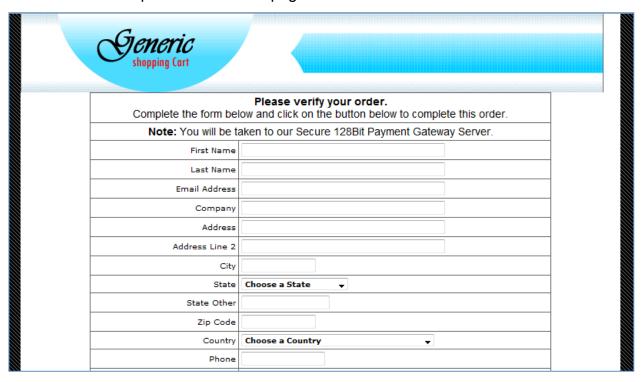


- Change the quantity of items to purchase and recalculate the price
- Click the trash can icon to remove this item
- Continue Shopping
- Clear the Cart
- Go to the Checkout page

### 1.2.2 Checkout page

When your customer presses the Checkout button, they will come to the second of three pages in the cart system.

On the Checkout page, your customer will enter their shipping and /or billing information. The top of the Checkout page will look similar to:





They will also select their payment method based on the options that you have chosen in the Admin Control Panel Configuration page. Here is an example of the payment selection area of the Checkout page:

Comments		
Payment Method	Select Payment Method ▼	
If you are paying via credit o	ard please fill out the fields below	
Card Type :	Select Credit Card Type ▼	
Credit Card Number :		
Expiration Date (mm/yyyy) :	Select Month ▼ Select Year ▼	
CVV Code :		
View Cart	Verify Your Order	
	Sub Total	\$14,99
	Discount	\$0,00
	Тах	\$0,00
	Shipping	\$12,95
	Total	\$26,00

If you have enabled the **Comments** function in the Admin Control Panel, a comments field will be included and your customer will be able to enter text into it.

The **Select Payment Method** drop down box will display all the Gateways that you have enabled in the Admin Control Panel.

A set of Credit Cart input fields will appear if you're using a local credit cart gateway provider or if you're using the manual credit card gateway. The **Select Credit Cart Type** will display all the types of credit cards that you are willing to accept.

When the customer presses the "Verify Your Order" button, the information submitted will be tested against your "Validation" parameters. If the validation check fails, the



customer will be notified that certain fields need their information provided or changed. And if it passes, they will be sent to the last page in the cart, the Confirmation page.

### 1.2.3 Shipping Selecton page

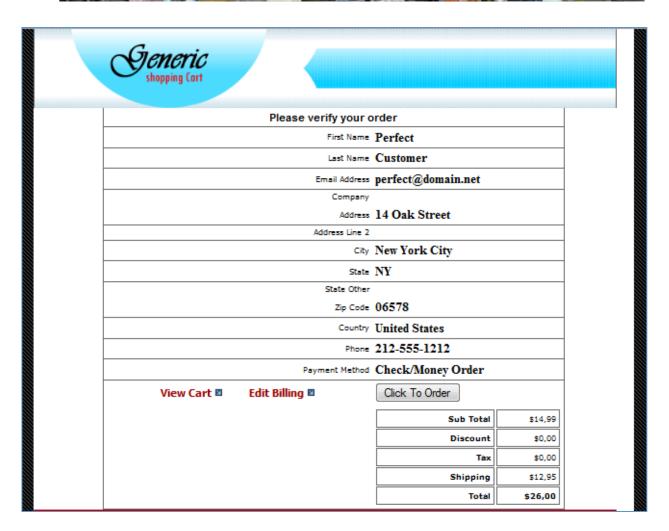
(This section does not apply to Legacy Shipping)

If you're using the Zone Shipping system and the customer's address information enables more than 1 shipping choice, the Shipping page will appear. This page show each shipping option available to the customer and the cost for that shipping.

### 1.2.4 Confirmation page

The Confirmation page presents your customer with a summary of their shipping and/or billing information and a pricing summary of their purchase. They will also see the payment method that they have selected.

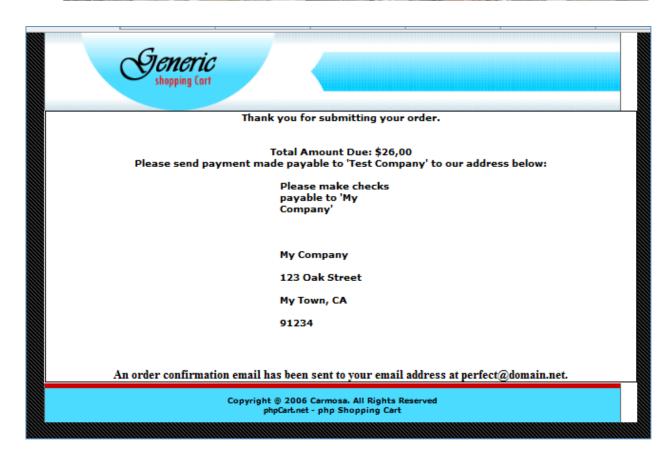




When the customer is prepared to make the purchase, they select the "Click to Order" button. This will activate the payment gateway that they have selected.

If they have chosen a "Local Gateway", their order will be processed and they will see a "Thank You" page similar to this one:





If they have chosen a "Remote Gateway", they will be sent to the gateway's web page to process their payment. They will be presented with a "Thank you" message upon their return from the Remote Gateway (if the Remote Gateway supports the success return functionality).

When the customer presses the "Click to Order" button, the phpCart software sends two emails. These are order confirmation notices to tell you that an order has been place. This order confirmation does not signify that the customer has made a payment for the order.

The first email will go to the customer and confirms their order. The email will have your Admin Control Panel company information on it and will contain:



- Customer address
- A list of the cart items including quantity and price
- Their comments
- Order pricing summary

The second email will be sent to the Admin email address. This email only tells you that the customer has placed an order. This does not mean that the customer has paid for the order. This order confirmation will provide you with:

- Customer address
- A list of the cart items including quantity and price
- Their comments
- Order pricing summary
- The Payment Method chosen by your customer

Depending on the Gateway chosen by the customer, you may need to oversee the payment side of the transaction. Most remote gateways will send you and the customer a payment confirmation email upon a successful payment.

Completing the three cart pages and making the payment concludes a successful transaction for you and your customer.



# 1.3 System Requirements

In order to run the phpCart software, you must be able to download the application from the phpCart web site. You will also need to FTP the files from your local machine to the server in BINARY mode.

The server must be running PHP V5 and must support either the Zend Optimizer or the IonCube modules.

The cart uses both cookies and sessions to manage customer purchase and contact information.

No database is required with this system.



# 2 Installation

### 2.1 Installation

Download the latest version of phpCart from our web site. Choose either Zend or lonCube encoding based on which encoder is available on your server. See Section 10.1 for instructions on how to use phpinfo() to check the encoding options available on your server.

The phpCart download is a zip file and must be unpacked to be usable. You can unpack the distribution either on your local machine or on the server.

**Step 1:** Create a directory on your website where you wish to keep phpCart (typically phpcart).

**Step 2:** Upload all of the phpCart files or the unzipped distribution package in BINARY mode.

NOTE: If you do not upload the distribution in BINARY mode all you'll see when you navigate to the Admin Control Panel is a pile of random characters on the screen.

If you are unpacking it on the server, you can do it with the following commands (via telnet):

*gunzip phpCartx.x.zip* (where x.x is the version number)

**Step 3:** If you are using phpCart on Linux, you must set the permissions on some key directories and files.



Set the read and write access to 777 on the sessions directory and the orders directory.

You can set the permissions to the *sessions* and *orders* directions by:

- 1. Use Telnet by typing *chmod 777 sessions* and *chmod 777 orders*.
- Use your FTP program to change the permissions. In most FTP programs (including Dreamweaver), you can change permissions by connecting to the Remote server and then right clicking on the directory to select *Set Permissions*.
- 3. Use your cPanel file manager to change permissions.

You must also set the file permissions to "write access" for the following files in the *admin* folder:

```
admin_1.php

configuration_1.php

coupon_1.php

payment_1.php

processor_1.php

regions_1.php

countries_1.php

currency_1.php
```

You must also set the file permissions to "write access" for the following file in the *modules* folder:

order\_number.php



### 2.2 License File

The phpCart product uses a License file to ensure that only purchased and licensed copies of phpCart are running. The License file named "license.php" must be located on your server for the phpCart product to run.

You can download a demo License that will work for 30 days. And if necessary, you can renew that demo License once for a total of 60 days. If you need more time to evaluate the phpCart for a purchase decision, please contact us.

Once you make a purchase, you will be sent an email giving you access to the License Manager web page where you can create a permanent License for your web site. This License will remain in effect for the lifetime of your domain name.

The phpCart License file can support SSL Certificates used to create a secure link between your customer's browser and your server. It will also use the https URL and display the padlock symbol. You can learn more about using an SSL Certificate with the phpCart License file below.



Here is an example of a License file:

```
<?
 2
          $License = array
 3
               "product"
                                   => "PHPCart Single License",
 4
              "domain" => "webrigger.com",

"version" => "4.60",

"expires" => "Never",

"supportexpires" => "2008-04-05 00:00:00",
 5
 6
 7
 8
9
               "items"
                                     => "0",
                                 => "0",
=> "0",
               "accounts"
10
11
               "license"
                                     => "e45c-7d8f-1ec2-29b0"
12
          );
    ?>
13
```

Note: It is important that there are no extra spaces or lines before or after the license file.



# 3 Configuration

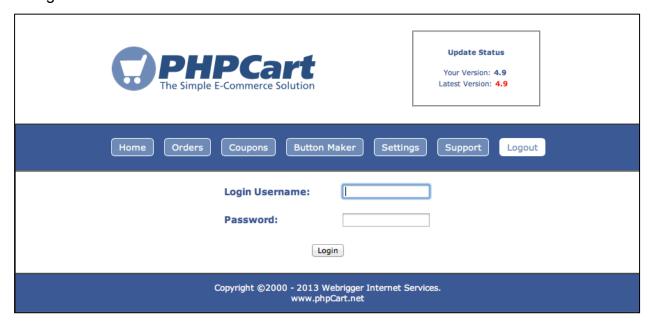
Once you upload phpCart to your server and complete the installation, you can start to configure the phpCart software.

# 3.1 Log into the Administration Control Panel

The very first thing to do after the installation is to use your browser to go to the Admin Control Panel. The location of the Admin Control Panel is at:

http://www.your domain.com/your cart directory/admin/

Going to this address will show the Admin Control Panel.



You can log in using these credentials:

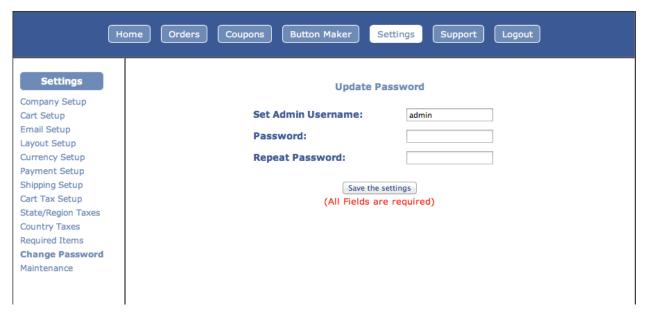
Login Username: admin Password: password



# 3.2 Change Admin Password

You should immediately change your password to protect your store!

Click the Settings tab, select **Change Password** from the left menu bar. You will then see this screen:



You can now change the Admin Username and Password.

Change the Username and/or Password to something that is easy for you to remember but will be difficult for anyone else to guess.

# 3.3 phpCart Configuration



Click the Settings tab to see all the items that you can configure. This screen will allow you to set the core values of phpCart. They include values associated with:

- Company Setup
- Cart Setup
- Email Setup
- Layout Setup
- Currency Setup
- Shipping Setup
- Cart Tax Setup
- State/Region Taxes
- Country Taxes
- · Required Items

We will now review each of these areas and how to set up their configuration.

### 3.3.1 Company Setup

The Company Setup parameters allow you to identify your company information. They include:

- Company Name
- Company Address
- Company Web Site
- Company Phone
- Company Email
- Company Admin Email
- Additional Order Email addresses
- Home Page URL
- Time Zone Offset
- Maintenance Mode (Store Closed)



The Company Name, Address, Web Site, Phone and Email are values used when communicating with your customers via email.

The **Company Admin Email** is the email address used to send purchase notifications to your (or your designee) via email. You can send additional copies of these purchase notifications by adding email addresses to the **Additional Order Email** field. To add more than one email address, just separate them with a comma and a space.

The **Homepage URL** is your store's main product page. It must be proceeded by a "/".

The **TimeZone Offset** value allows you to change the times used in phpCart. The offset is from GMT.

**Maintenance Mode** allows you to "close" your store to outsiders. While the store is closed, anyone logged into the Admin Control Panel will be able to see and use the store. This functionality is useful when you are making modifications to your store and don't want customers to see or use the store.

When your store is "closed", your visitors see a new "store closed" page, which is a combination of a new template (template/misc/store\_closed.php) and a new entry in the Language section of the Admin Control Panel. You can modify both the template and the custom language entry to suit your needs.

You can also add the following php code to the top of your product definition pages if you want them to also check if the store is open and to display the store closed message if it is closed. If your product definition pages have a .html or .htm suffix in their name, you will need to update the .htaccess file to enable execution of php on those page.

<?php



```
if (session_id() == ") {
            session_start();
}
include_once 'includes/functions.inc.php';
CheckStore();
?>
```

The following image shows you the Company Setup Screen.



|  | Company Configuration  |
|--|--|
| Your Sto   | re Time:: Feb 27th, 2013 7:18:40 am                          |
| Company Name:  | My Company's Name  |
| Company Address:   | 123 Oak Street<br>Suite 101A<br>Box 444<br>My Town, CA 91234 |
| Company Website:   | www.mycompany.com  |
| Company Phone:   | 123-456-7890   |
| Company Email:<br>(Displayed on client<br>emails, etc)   | sales@yourdomain.com   |
| Company Admin Email:<br>(Email to receive receipts,<br>orders, etc)  | sales@yourdomain.com   |
| Additional Order Emails:<br>(Orders will be emailed to<br>these also.<br>Separate with a comma<br>and space) |  |
| Homepage URL:  | index.html   |
| Time Zone Offset:  | Default (GMT -5 hours)   \$                                  |
| Maintenance Mode:<br>(Store Closed)  | No \$  |

# 3.3.2 Cart Setup

The Cart Setup screen allows you to fine tune how the cart works including selecting the way that Order ID's are created.



**Web Path to Cart Directory:** Insert the web site address including last "/" for your web site.

**File Path to Cart Directory:** Insert the path location of your web site on the server. We have provided our guess to this value in the "Should be" line.

**Domains that can access your Cart**: Insert the domains that can access your cart. Only these domains will be allowed to enter products into your cart and to conduct transactions. This is a security feature.

**One Page Checkout:** By setting the one page checkout setting, a customer's order will be accepted after they properly complete the Checkout page and press the submit button. This will allow them to skip the Confirmation page.

The one page checkout setting will not be suitable for everyone. The best people to use this are those with no shipping or tax fees.

#### **Upside:**

- faster customer checkout experience

#### Downside:

- Customer does not confirm their shipping/billing info
- Customer does not confirm CC info (for manual CC orders)
- Customer does not see shipping calculation (if you add a shipping cost)
- Customer does not see tax calculation (if you add taxes)

#### Use separate billing and shipping fields in the checkout and confirmation pages:

This setting determines which templates are used for the checkout and confirmation page.

When the setting is set to **No**, these pages are used:



templates/generic/billing.php and templates/generic/confirm.php

When the setting is set to **Yes**, these pages are used: templates/generic/billing-shipping.php and templates/generic/confirm-billing-shipping.php

**Create a file in the "orders" directory:** Select Yes if you would like to retain a copy of the order on the server and for it to be accessible in the Admin Control Panel by selecting View Orders (see below for more details).

**Order ID Format:** You can now create your own Order ID format. There are three possible components:the prefix, body and suffix.

PREFIX: You can choose to add a prefix to your Order ID. If you decide to use a prefix, it can be either the date in YYYYMMDD format or a custom string that you enter.

BODY: You can choose to use a body for Order ID. If you decide to use a body, it can be a date in YYYYMMDD format, a random number or a sequential number. You can also choose the quantity of digits that you would like the random or sequential number to display between 4 and 8 digits.

SUFFIX: You can choose to add a suffix to your Order ID. If you decide to use a suffix, it can be either the date in YYYYMMDD format or a custom string that you enter.

**Allow Duplicates in Cart:** Select Yes if you plan to create products that have different options, as each of those products will be considered a unique item. Select No, if you want customers to increase their purchase quantity in the cart.



**Allow Fractions in the Quantity Field:** Select Yes if you want to use fractions with your products.

**Set Maximum Quantity of Items in Cart:** If you would like to limit the quantity of items that can be inserted into your cart, set this value to "Yes".

**Maximum Quantity of Items:** Enter the number which is the maximum quantity of items that you will allow entered into your cart.

Combine same Product ID numbers for quantity discount: If you have a single product but have different options for it and would like to allow customers to add up items to qualify for a discount, select 'Yes". To make this work, just use the same Product ID number for each product definition.

**Send Admin Notification Email upon Cancelled Payment:** If you would like to be notified upon a customers return from a payment gateway when they have cancelled the payment, select "Yes". This is only effective if the gateway has a cancel option.

**Send Admin Notification Email upon Declined Payment:** If you would like to be notified upon a customers return from a payment gateway when they have been declined for the payment, select "Yes". This is only effective if the gateway has a declined option.

**Enable Payment Notification (IPN):** This is a legacy setting and is only used for people using the Download Manager Product (which is no longer available).

**Require Hyperlink Keys:** Select Yes if you would like to use this security feature where a hash key is created for each product and must be submitted with the product or option hyperlink (or HTML form code) for it to show up in the cart.



The following image shows you the Cart Configuration Screen.

| Cart Configuration  |   |
|---|---|
| Web Path to Cart Directory* (http://subdomain.domain.suffix/phpcart/)             | http://www.yourdomain.com.com/phpcartV4.10/   |
| File Path to Cart Directory* (Should be: /home/phpcart/public_html/demo/phpcart/) | /home/phpcart/public_html/demo/phpcart/   |
| Domains that can access your Cart:<br>(yourdomain.com www.yourdomain.com)         |   |
| One Page Checkout:  | No ‡  |
| Use separate billing and shipping fields in the checkout and confirmation pages:  | Yes ‡   |
| Create a file in the "orders" directory:  | Yes ‡   |
| Order ID Format:  | Prefix: None Date (YYYYMMDD) Custom  Body: None Date (YYYYMMDD) Random Sequential Number 6 (next order) |
|   | Display Digits:  4 5 6 7 8  Suffix: None  Date (YYYYMMDD)  Custom                                       |
| Allow Duplicate Items In Cart:<br>(Needed if your products have options)          | Yes ‡   |
| Allow Fractions in Quantity Field:  | No ‡  |
| Set Maximum Quantity of Items in Cart: Maximum Quantity of Items:                 | No ‡  |
| Combine same Product ID numbers for quantity discount:                            | Yes ‡   |
| Send Admin Notification Email upon Cancelled Payment:                             | Yes ‡   |
| Send Admin Notification Email upon Declined Payment:                              | Yes ‡   |
| Enable Payment Notification: (IPN) (Only used for Download Manager)               | No ‡  |
| Require Hyperlink Keys:   | No ‡  |
| Save the settings   |   |
| * Mandatory Fields  |   |

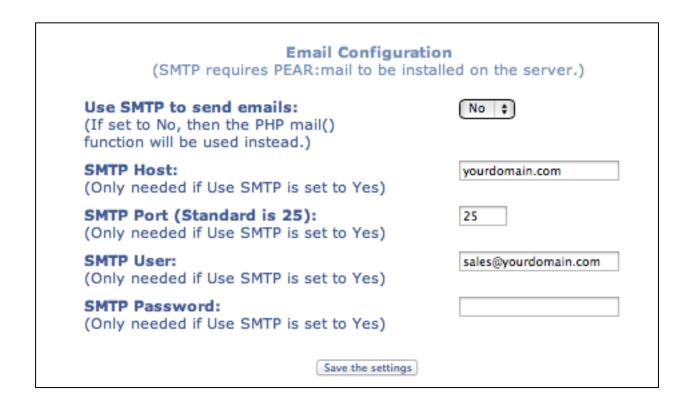


### 3.3.3 Email Setup

The Email Configuration screen allows you to choose how your system sends email to you and your customers.

You can choose between the standard php mail() functionality and SMTP functionality.

Note: You must have PEAR:mail installed on your server to be able to use the SMTP email functionality.



### 3.3.4 Layout Setup

The Layout Configuration sets many of the values used in displaying your cart and in many ways acts like a CSS file.



| Template:                                       | generic 🛊           |
|---|---------------------|
| Cart Description:<br>(Sent to payment gateways) | Shopping Cart Order |
| Header Text Color:                              | #ffffff             |
| Header Background Color:                        | #6878a7             |
| Item Text Color:                                | #000000             |
| Item Background Color:<br>(odd rows)            | #dddddd             |
| Item Background Color:<br>(even rows)           | #bbbbbb             |
| Font:<br>(preferred order, comma-separated)     | verdana             |
| Font size:                                      |                     |
| Language:                                       | english   ‡         |
| Date Format:<br>(ex: M jS, Y g:i:s a)           | M jS, Y g:i:s a     |
| Number of Decimal Places:                       | 2                   |
| Decimal Character:                              |                     |
| Thousands Separator:                            | ,                   |
| Display Comments field on<br>checkout:          | Yes 🛊               |
| Activate Coupon System:                         | Yes 💠               |
| Hide discount on subtotal if \$0:               | Yes ‡               |

**Template:** This field allows you to choose a directory to use for your standard template. The templates are located in the /templates folder. If you add a new folder in this directory, it will show up in this drop down box.



**Cart Description:** This label is used when sending orders to remote gateways. It allows you to set a value that is custom to your store.

**Header Text Color:** Choose the hex value for the text color that you would like displayed in the Header. Must start with the # symbol.

**Header Background Color:** Choose the hex value for the background color that you would like displayed in the Header. Must start with the # symbol.

**Item Text Color:** Choose the hex value for the text color that you would like displayed in the Items Table. Must start with the # symbol.

**Item Background Color (odd rows):** Choose the hex value for the text color that you would like displayed in all odd rows of the Items Table. Having alternating colors on even and odd rows is helpful for tables with lots of items. Must start with the # symbol.

**Item Background Color (even rows):** Choose the hex value for the text color that you would like displayed in all even rows of the Items Table. Having alternating colors on even and odd rows is helpful for tables with lots of items. Must start with the # symbol.

**Font (preferred order, comma separated):** Insert the font names that you would like to use in your cart. You can have more than one so that if the primary one is missing from your customer's browser, it will default to the next available font. Just make sure to separate them with a comma and a space.

**Font size:** Insert a number between one and five to choose the size of font to display. The larger number is the larger size.

**Language:** The language drop down box allows you to select from the four available language options that are currently available in phpCart. They are: English, French,



Italian and Spanish. If you would like to use a different language, it's very easy to do, just read the Internationalization instructions below.

**Date Format:** You can display dates in your cart (and in your emails) using this Date Format. Refer to the php date() command for all the applicable abbreviations.

**Number of Decimal Places:** Choose the quantity of decimal places you want after the decimal symbol.

**Decimal Symbol:** Choose the decimal symbol that you want displayed in your cart (and your emails).

**Thousands Separator:** Choose the symbol that you would like to use to separate the thousands value in the cart (and in your emails).

**Display Comments Field on Checkout:** Select Yes if you would like a comments box to be displayed in your cart. The comments entered by your customers will be included in the Admin email sent to you.

**Activate Coupon System:** Select Yes if you would like to have the coupon input box should up in your cart. You can create coupons to provide unique discounts for your customers.

**Hide Discount if Subtotal is \$0:** Select Yes if you would like the coupon discount to be hidden if the subtotal of an order is \$0.

## 3.3.5 Currency Setup

The Currency Setup screen shows you all the currencies that are available to be used in your shopping cart.



There must be one currency setup to use the phpCart software. To use multiple currencies, add the currency code to your "Buy Now" buttons.

To add additional currencies, you can use the Add Currency link in the left menu bar. To edit existing currencies, click the "pencil icon" which will open the Edit Currency screen.

| rrency   |                  |                  | Manage              | e Currency          |                    |                |        |
|----------|------------------|------------------|---------------------|---------------------|--------------------|----------------|--------|
| Currency | Currency<br>Name | Currency<br>Code | Currency<br>Display | Primary<br>Currency | Conversion<br>Rate | PayPal<br>Code | Action |
| ttings   | US Dollar        | usd              | US \$               | Yes                 |                    | USD            | Ø X    |
| Setup    | Canadian Dollar  | cand             | CAN \$              |                     | 1.03               | CAD            | 0 ×    |
|          |                  |                  |                     |                     |                    |                |        |
|          |                  |                  |                     |                     |                    |                |        |
|          |                  |                  |                     |                     |                    |                |        |
| tup      |                  |                  |                     |                     |                    |                |        |

#### 3.3.5.1 Add Currency

**Currency Name:** Enter the currency name that you would like listed.

Currency Code: Enter the currency code that you will use in your "Buy Now" buttons.

Currency Display: Enter the currency symbol that you would like displayed in your shopping cart pages and on your order notification emails.

**Primary Currency:** Click this box if you would like this currency to be your Primary currency. The primary currency is used when no currency is entered in the "Buy Now" button.

**Conversion Rate:** If you would like phpCart to covert your currency, enter the amount here.



**PayPal Code:** If you are using PayPal as one of your processors, select the correct name of your currency. The list includes only the currencies that PayPal will accept.

| Currency Name: *  |  |  |  |
|---|--|--|--|
| Currency Code: * (used in product definition)                   |  |  |  |
| Currency Display: *   |  |  |  |
| Primary Currency:   |  |  |  |
| Conversion Rate:  |  |  |  |
| PayPal Code:  | Select PayPal Code 💠   |  |  |
| (Items  | with * are required)  Note:  |  |  |
| prices for each currency in you when you provide a single price | version rate blank if you provide separate ur product defintion. Use the conversion rate the and a currency code in your product conversion automatically for you. |  |  |

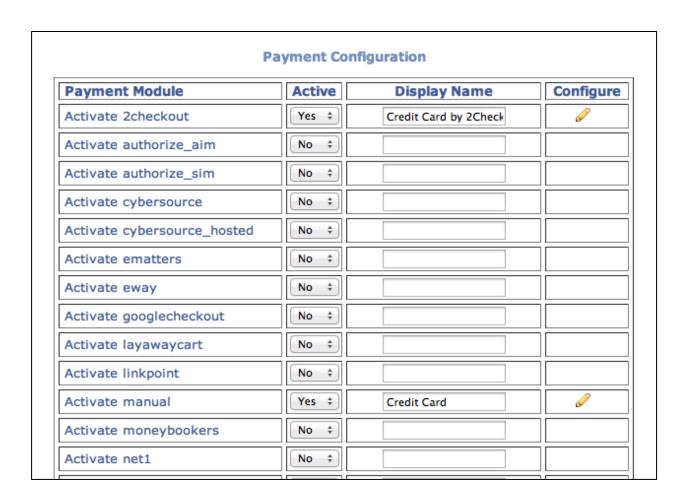
# 3.3.6 Payment Setup

phpCart supports multiple payment gateways simultaneously allowing you to let your customer choose their payment method. You can activate as many gateways as you would like to use.

There are two basic types of payment gateways: Local gateways and Remote gateways.



- Local payment gateways take all of the cart information and the payment information inputted by your customer in your cart and process it in a batch mode.
- Remote payment gateways take the cart information but send the client to their own website where the customer inputs their payment information for processing their payment.



To activate a gateway, select Yes in the Active column. Then input the name that you would like displayed in the cart's payment selection drop down box in the Display Name column.



Once a gateway is activated you must click on the Configure link next to the gateway name and set the unique configuration options for that gateway. Once that is complete, your gateway is ready for use.

Some gateways can be configured with certain return URLs depending on whether or not the payment was completed. If you wish to have the client return to the cart, you can use the following URLs:

- Payment Successful: http://www.yourdomain.com/phpCart/phpCart.php?action=complete
- Client Canceled the Order at the remote Gateway:
   http://www.yourdomain.com/phpCart/phpCart.php?action=canceled
- Payment Declined:
   http://www.yourdomain.com/phpCart/phpCart.php?action=declined

Remember, not all payment gateways support these return URLs.

## 3.3.7 Shipping Setup

phpCart has many options for calculating shipping. phpCart can apply both product specific shipping prices or cart-wide shipping prices.

Product specific shipping prices are configured when you create a product definition. See the product definition section for a full description on how to set product specific shipping prices.

The Admin Control Panel's Shipping configuration controls the cart-wide shipping prices.

The cart-wide shipping prices are broken down into three groups:



- By product
- By weight
- By order

| Shipping Con  | figuration   |
|---|--|
| Shipping/Handling Per Order:<br>(Added to the entire order)   |  |
| Level for Free Shipping:<br>(If total amount is greater or equal to this<br>amount then shipping is free)   |  |
| Shipping/Handling Per Item:<br>(Added for each product)   |  |
| Shipping By Percent of Total:<br>(Calculate a percentage of total as the<br>shipping charge)  |  |
| Shipping By Weight: (Enter a list of weights and charges separated by a colon. Separate groups with a semicolon. The list should look like this: weight:charge; weight:charge Start with 0 as the first weight) | (Example: 0:2.50;5:5.00;10:8.75<br>The above will charge 2.50 if the<br>weight is over 0, 5.00 if the weight is<br>over 5 and 8.75 if the weight is over<br>10.) |
| Minimum Shipping Charge:<br>(Mimimum amount to charge for shipping.<br>If calculating shipping does not reach the<br>minimum, then the minimum is used)   |  |
| Maximum Shipping Charge: (Maximum amount to charge for shipping. If calculating shipping exceeds the maximum amount, then the maximum is used)  |  |
| Save the se   | ttings   |

### **Cart Wide Shipping Options:**

You can assign shipping fees based on the order. There are several different ways that you can set the order fee:

• You can assign a shipping fee for each product in the cart.



- You can assign a shipping fee for the entire order.
- You can set a free shipping level where the customer receives free shipping if they hit a certain purchase amount
- You can also set a shipping fee as a percentage of the total order.
- Finally, you can set minimum and/or maximum shipping fees that will override all other shipping calculations.

### 3.3.8 Cart Tax Setup

The Tax Configuration section allows you to set the cart-wide tax behavior of your store.

**Tax All Items** – Set this value to "YES" if you would like to calculate taxes on your products. You will need to set up State/Region and/or Country taxes for this to take effect (see State/Region and Country tax Settings below).

**Tax Shipping** – Set this value to "YES" if you would like to have taxes applied to your shipping cost.

**Default Tax Rate 1** – Enter a temporary tax rate, which is used to estimate taxes until the customer enters their address information into the Billing section of the cart.

**Default Tax Rate 1** – Enter a temporary tax rate, which is used to estimate taxes until the customer enters their address information into the Billing section of the cart.



| Тах   | Configuration    |
|---|------------------|
| Tax All Items: (If you want to charge tax for all taxable customers and all products)                         | Yes ‡            |
| Tax Shipping: (If selected, only customers that are taxable will be taxed for shipping at the taxrate1 rate.) | Yes ‡            |
| Default Tax Rate 1:<br>(Prior to any state/region or country<br>being entered)                                |                  |
| Default Tax Rate 2:<br>(Prior to any state/region or country<br>being entered)                                |                  |
| S   | ave the settings |

### 3.3.9 State/Region Tax Setup

Use the State/Regions Tax screen to assign taxes to any state or region. Input the tax values that you would like to use in Taxrate 1 and Taxrate 2 before pressing the Save button at the bottom of the page.

These tax rates will become effective if you have activated Taxrate 1, Taxrate 2 or Taxrate 3 in your "Buy Now" button. If you use Taxrate 3, the cart will provide values for Tax1, Tax2 and will add the values from Taxrate 1 and Taxrate 2 together to create the Total Tax.

This screen can also be used to add or delete states and regions.



|    | States/Regions Tax Configuration |                              |           |           |        |  |  |
|----|----------------------------------|------------------------------|-----------|-----------|--------|--|--|
| ID | State/Region Name                | State/Region<br>Abbreviation | Taxrate 1 | Taxrate 2 | Delete |  |  |
| 1  | Alberta                          | AB                           |           | 5         | ×      |  |  |
| 2  | British Columbia                 | ВС                           | 7         | 5         | ×      |  |  |
| 3  | Alabama                          | AL                           |           |           | ×      |  |  |
| 4  | Alaska                           | AK                           |           |           | ×      |  |  |
| 5  | Arizona                          | AZ                           |           |           | ×      |  |  |
| 6  | Arkansas                         | AR                           |           |           | ×      |  |  |
| 7  | California                       | CA                           |           |           | ×      |  |  |
| 8  | Colorado                         | СО                           |           |           | ×      |  |  |
| 9  | Connecticut                      | СТ                           |           |           | ×      |  |  |
| 10 | Delaware                         | DE                           |           |           | ×      |  |  |
| 11 | District of Columbia             | DC                           |           |           | ×      |  |  |
| 12 | Florida                          | FL                           |           |           | ×      |  |  |

# 3.3.10 Country Tax Setup

Use the Country Tax screen to assign taxes to any country. Input the tax values that you would like to use in Taxrate 1 and Taxrate 2 before pressing the Save button at the bottom of the page.

These tax rates will become effective if you have activated either Taxrate 1, Taxrate 2 or Taxrate 3 in your product definition. If you use Taxrate 3, the cart will provide values for Tax1, Tax2 and will add the values from Taxrate 1 and Taxrate 2 together to create the Total Tax.

This screen can also be used to add or delete countries.



|    | Country Tax Configuration |                          |                          |                 |           |           |        |
|----|---------------------------|--------------------------|--------------------------|-----------------|-----------|-----------|--------|
| ID | Country Name              | 2 Letter<br>Abbreviation | 3 Letter<br>Abbreviation | 3 Digit<br>Code | Taxrate 1 | Taxrate 2 | Delete |
| 1  | Afghanistan               | AF                       | AFG                      | ddd             | 5         | 6         | ×      |
| 2  | Albania                   | AL                       | ALB                      | 004             |           |           | ×      |
| 3  | Algeria                   | DZ                       | DZA                      | 008             |           |           | ×      |
| 4  | American Samoa            | AS                       | ASM                      | 012             |           |           | ×      |
| 5  | Andorra                   | AD                       | AND                      | 016             |           |           | ×      |
| 6  | Angola                    | AO                       | AGO                      | 020             |           |           | ×      |
| 7  | Anguilla                  | Al                       | AIA                      | 024             |           |           | ×      |
| 8  | Antartica                 | AQ                       | ATA                      | 660             |           |           | ×      |
| 9  | Antigua and Barbuda       | AG                       | ATG                      | 010             |           |           | ×      |
| 10 | Argentina                 | AR                       | ARG                      | 028             |           |           | ×      |
| 11 | Armenia                   | AM                       | ARM                      | 032             |           |           | ×      |
| 12 | Aruba                     | AW                       | ABW                      | 051             |           |           | ×      |

# 3.3.11 Required Items Setup

The Required Items Setup screen serves two purposes:

- 1) Putting data into this field makes the field a "**Required**" field for your customer. If they do not input a value in a required field, they will see an error message telling them to fill in the identified fields.
- 2) The text that you enter into these fields is displayed in your cart as part of the error message when a customer on the Billing page leaves this field blank.

If you do not want a field to be required, then just leave it blank.

Example: If you do not want to make the phone billing field required then delete all the text from the Text for Phone field.



| Required Ite  | ems Configuration |  |  |  |  |
|---|-------------------|--|--|--|--|
| Leave blank if input field is not required. Otherwise enter the words that you want to precede the "Required" error message in the checkout form. |                   |  |  |  |  |
| Text for First Name: First Name   |                   |  |  |  |  |
| Text for Last Name:   | Last Name         |  |  |  |  |
| Text for Email Address:   | Email             |  |  |  |  |
| Text for Address:   | Address           |  |  |  |  |
| Text for City:  | City              |  |  |  |  |
| Text for State:   | State             |  |  |  |  |
| Text for Zip Code:  | Zip               |  |  |  |  |
| Text for Country:   | Country           |  |  |  |  |
| Text for Phone:   | Phone             |  |  |  |  |
| Save the settings   |                   |  |  |  |  |

#### 3.3.12 Maintenance

Clear Customer Carts: Each time a person puts a product into your shopping cart, phpCart creates a file in your Sessions directory to store their cart information. This function allows you to clean out some of those files, which are unnecessary once your customer has placed their order. It is highly recommended to do this at least once a month.

**Clear Order History:** Each time a customer places an order, an file is created in the Orders directory which holds the customer's order information. These files also create the Show Orders list. At some point, this list of orders and the subsequent quantity of order files will get to be very large. This function allows you to prune this list to a manageable size.



| Be careful as these ar | Routine Maintenance re permanent changes and cannot be reversed                                     |
|------------------------|---|
| Clear Customer Carts   | <ul><li>Older than 1 Month</li><li>Older than 1 Week</li><li>Older than 1 Day</li><li>All</li></ul> |
|                        | Clear Customer Carts  |
| Clear Order History    | <ul><li>Older than 1 Year</li><li>Older than 1 Month</li><li>All</li></ul>                          |
|                        | Clear Order History   |

# 3.4 Coupons

phpCart has an extensive coupon system designed to allow you to create custom discount coupons for your customers. Coupons can be used to entice new customers or to reward existing customers.



Select the Coupons Tab at the top of the Admin Control Panel to use Coupons.

| Show All Coupons   |   |    |    |           |        |    |    |     |
|--|---|----|----|-----------|--------|----|----|-----|
| Coupon Code Type Amount Minimum Qty Expire Expire Day Acti |   |    |    |           | Action |    |    |     |
| 10%tuesday   | % | 10 | 25 | unlimited | 2014   | 01 | 01 | Ø X |

To create a coupon, click the Add New Coupon link in the left menu bar. To edit an existing coupon, click the pencil icon in the Action box.

#### 3.4.1.1 Add New Coupon

**Coupon Code:** Enter the name of the coupon which you will present to customers and they will use in the cart to activate the coupon discount.

**Type:** You can create 4 different types of coupons:

- Discount by \$
- Discount by % (Percent)
- Discount Shipping
- Free Shipping

**Amount:** Enter the amount of the discount (without a currency symbol). Leave blank for free shipping.

**Quantity of Coupons:** Enter the quantity of times that this coupon can be used. Leave blank for unlimited uses.



**Minimum Order:** Enter the minimum order subtotal amount that must be reached before this coupon will go into effect.

**Expire Year:** Select the year that this coupon will expire.

**Expire Month:** Select the month that this coupon will expire.

**Expire Day:** Select the day that this coupon will expire.

| Add New Cou  | ıpon        |
|--|-------------|
| Coupon Code: *   |             |
| Type: *  | \$ Discount |
| Amount: * (Amount of discount. Leave blank for free shipping.) |             |
| Quantity of Coupons Available:<br>(leave blank for unlimited)  |             |
| Minimum Order: *   |             |
| Expire Year:   | 2013 ‡      |
| Expire Month:  | 1 \$        |
| Expire Day:  | 1 ‡         |
| (Items with * are  |             |



### 3.5 Orders

**Show All Orders** 

| Date                          | Name           | Company Name       | Order ID             | Action |
|-------------------------------|----------------|--------------------|----------------------|--------|
| Jan 18th, 2013<br>7:02:27 am  | Frank Smith    | Test Company       | 20130118-0030-<br>BB | ✓ ×    |
| Jan 18th, 2013<br>6:03:10 am  | Sally Smith    | My Private Company | 20130118-0029-<br>BB | ✓ ×    |
| Jan 18th, 2013<br>5:59:19 am  | Marvin Sals    | Sals Fine Tools    | 20130118-0028-<br>BB | ✓ ×    |
| Jan 17th, 2013<br>12:47:57 pm | Chuck Rubinski |                    | 20130117-0027-<br>BB | ✓ ×    |
| Jan 17th, 2013<br>6:27:57 pm  | Rachel Jones   |                    | 20130117-0026-<br>BB | ✓ X    |
| Jan 16th, 2013<br>7:56:15 am  | Dawn Lombardi  | Lombardi Sales     | 20130116-0025-<br>BB | ✓ ×    |
| Jan 16th, 2013<br>7:48:38 am  | Mario Jules    | Jules Inc.         | 20130116-0024-<br>BB | ✓ ×    |
| Jan 15th, 2013<br>10:32:25 am | Michelle Lins  |                    | 20130115-0000-<br>BB | ✓ X    |

You can see a list of all of your orders on the Orders tab. Each order will show the Date, Name of the purchaser, Company Name (if was filled out), and the Order ID.

Selecting the checkmark icon will open a new window, which will show you the entire order form, which is the same document that is received as the Admin Order Confirmation Email. There will also be a link on that page for a Printable Invoice.



Selecting the "X" icon will delete the Order from the list.



# 4 Initial System Test

Once you have completed the Installation and the Configuration, we highly recommend that you run a quick test on the cart to make sure that the installation and configuration were successful.

View the included prototype product page in your browser by going to: www.yourdomain.com/your\_path\_to\_the\_cart/

- 1) This will display the index.php page which is a "prototype product page".
- 2) Select any product and see that the cart page is displayed.
- 3) Press the 'Checkout' button. You should now see the Checkout page.
- 4) Enter some dummy information into the address fields but an accurate email address where you can receive emails. Select a payment method. Use the "Check/Money Order' option (if you have it set). It is a local gateway that doesn't require using the Credit Card info and is good for this basic test.
- 5) Press the 'Verify Your Order' button. You should now see the Confirmation page.
- 6) Press the "Click to Order" button. You should now see the Thank You page.

If your test follows this set of instructions, you have now completed a successful test on your shopping cart.



# 5 The Template System

# 5.1 The Standard Template

The standard phpCart release has a prototype template file to enable you to test the functionality of the software. It is located in the templates directory in the classic folder. The filename is template.php

The template.php file also uses the CSS file named, stylesheet.css.

We recommend that you make duplicates of these files before starting to create your own template files.

## 5.2 Creating Your Own Template

Setting up your template is extremely easy. You only need to create one template file to make phpCart look like the rest of your website.

- 1) First, you will need an html file of your template. Just take one of your web pages and make a copy of it and call it template.php.
- 2) Next, clean out all of the data in the template so you just have your header and footer.
- 3) Now, just place the Template System Tag, %%DATA%%, right where you want the shopping cart to be displayed.



4) Finally, upload your template.php file to the phpCart/templates/classic/ directory (after making a duplicate of this original file since this operation will overwrite the original template file).

If you would prefer to use the "generic" shopping cart then copy your template.php file into the phpCart/templates/generic/ directory and then log into the control panel and change your configuration option to the generic template (the default is the classic template).

If your template and cart don't line up correctly, you may have removed too much html code or not enough html code. You should insert your %%DATA%% tag right in between the table cell tags (%%DATA%%) that previously held the main content of the page.

Your cart will now match the look of your site by using your own web page layout as the shopping cart template. You can modify any of the existing templates that layout the rest of the shopping cart by modifying the files in the phpCart/templates/classic directory.

You can see the full documentation on how to modify a template by viewing the <a href="https://phpcart.com/phpcart

Remember, you can start out small by just making a few adjustments to the templates until you get a better understanding of how it works. Remember to make a backup copy before you start modifying the files.

Here are the remaining templates that are used in the phpCart system.



# 5.3 Cart Templates

There are three files which define the cart templates. They are:

- cart.php (the cart page)
- **billing.php** (the checkout page)
- confirm.php (the confirmation page)

You can change these files to modify the look and feel of these pages. You can also add or remove fields.

## 5.4 Email Templates

There are two files which define the email templates. They are:

- email\_customer.php (confirmation email sent to customer)
- email\_admin.php (confirmation email sent to admin)

You can change these files to modify the look and feel of the system emails. You can also add or remove fields.

# 5.5 Gateway Templates

There are two files which define the cart templates. They are:

- · canceled.php
- declined.php



You can change these files to modify the look and feel of these pages. You can also add or remove fields.

# 5.6 Thank You Templates

There are three files which define the thank you templates. They are:

- offline.php (used for successful order with an offline payment)
- thankyou.php (used for successful transaction and returned from a remote gateways)
- manual\_thankyou.php (used for an order where you will manually process the credit card transaction)

You can change these files to modify the look and feel of these pages. You can also add or remove fields.

## 5.7 MiniCart Template

The file **minicart.php** defines the display for the mini-cart. You can change this file to modify the look and feel of the mini-cart.

You can learn how to add a mini-cart onto any page by reading the "Display the Mini-Cart on Other Pages" in section 8 below.



# 6 Preparing your Store

Since phpCart does not use a database of any kind to store information about the products in your shop, all information about each product (description, price etc.) is defined by a hyperlink or a HTML form which is linked to the phpCart software.

# 6.1 Defining Products

Create a web page to use to display your products. For each product on your web page, create a "Buy Now" button. Use one of these three product definition methods:

- 1) Hyperlink link only
- 2) Hyperlink Buy Not button created with image
- 3) HTML forms with either HTML form buttons or image buttons

Any of these alternatives will work as long as you include the web path to phpcart.php in your Hyperlink or in the HTML form's action field.

You can also use the Button Maker in the Admin Control Panel to generate all three of these types of product links automatically.

You can also create your "Buy Now" definitions by using the output of a database, or by defining them using a scripting language.



### **6.1.1 Required Product Variables**

- action this value is always 'add'
- id the product ID
- descr the product description (short sentence, url encoded; see below)
- price the product price (must be numerical).

### **6.1.2 Optional Product Variables**

- · quantity the quantity of product to put in basket
- curr (currency) enter the currency code from the Currency Setup screen.
- weight weight of the product (must be numerical). Used if shipping charges are calculated by weight
- shipping Amount to charge per item for shipping/handling/postage
- shipping1 Amount to charge for the first item for shipping/handling/postage.
   See the Shipping section below for more information
- shipping2 Amount to charge for each additional item for shipping/handling/postage. See the Shipping section for more information
- tax Amount to charge for each item if the customer is taxable. See the Taxes section below for more information
- taxrateid Tax group of this item. See the Taxes section for more information.
- Options using one of the following techniques:
  - option[] This allows you to include an option such as product size or color plus an extra charge to add to the price of each item. You can have unlimited options by simply adding more option[] fields. See the Options section
  - option1 This is another way to include options. Use this form when including text options. Increment the number (option3, option4, etc.) to add more options.



- extra1 This is an extra field. See the Extra Variables section for more information
- extra2 This is an extra field. See the Extra Variables section for more information
- extra3 This is an extra field. See the Extra Variables section for more information

#### As an example, consider the following link:

```
<a href="/phpcart/phpcart.php?action=add&id=1001&descr=Nike+T-
Shirt&price=19.95">Add To Cart</a>
```

This is the minimum amount of information required to add a product to phpCart. Here is another example of a link:

```
<a href="/phpcart/phpcart.php?action=add&id=1001&descr=Nike+T-Shirt
&option[]=Green:2.95&option[]=Small&price=19&quantity=1&shipping=1.59&taxrate
id=1">Add to Cart</a>
```

In this case, the variables above have been assigned the following values:

- action = add
- id = 1001
- descr = Nike T-Shirt
- option[] = Green:2.95 (this will charge \$2.95 extra for each item)
- option[] = Small (no extra charge)
- price = 19
- quantity = 1
- **shipping** = 1.59
- taxrateid = 1

Notice how each variable in the example is separated by the ampersand (&). When your customers click on the link, the product will be added to their shopping cart automatically by phpCart.



If you have characters other than letters and numbers (special characters) in the description you will need to encode them. You can use the php function *urlencode(\$descr)* to replace these characters properly. This will replace spaces with a plus (+) and replace ampersands (&) with the html special character equivalent ( %26) which are the most common problems when including product info.

Always test your links on different machines with different browsers to make sure they all work correctly.

Here is an example of a product definition using an html form:

This example uses a drop-down for each of the options and a drop-down for the quantity.

```
<form action="/phpcart/phpcart.php" method="post" name="anything">
<input type="hidden" name="action" value="add">
<input type="hidden" name="id" value="1001">
<input type="hidden" name="descr" value="Nike+T-Shirt">
<input type="hidden" name="price" value="19">
<input type="hidden" name="quantity" value="1">
<input type="hidden" name="shipping" value="1.59">
<input type="hidden" name="taxrateid" value="1">
<select name="quantity>
<option value="1::">1 </option>
<option value="2::">2</option>
<option value="3::">3</option>
</select>
<select name="option[]">
<option value="Green:2.95:">Green - $2.95
<option value="Yellow:1.95:">Yellow - $1.95
```

<option value="Blue">Blue</option>



```
</select>
```

<select name="option[]">

<option value="Small">Small</option>

<option value="Medium:3:">Medium - \$3.00

<option value="Large:5.95:">Large - \$5.95

</select>

<input type="submit" value="Add to Cart">

</form>

In the example above we use a drop-down selection box to allow the customer to choose the quantity. You could also use a text box to allow the customer to input the quantity to order. If the quantity field is left blank then phpCart will automatically assign a quantity of one to the product.

# 6.2 Pricing

The most basic price option is just to insert "price=19" where the price value is just an integer. It can be written with or without the decimal point and pennies.

Example: (All these work)

- 19
- 19.50
- 19.5

If you wish to have different pricing for different quantity levels then enter the pricing information as follows:

price, minimum: price, minimum



Example: if you wanted to charge \$10.95 each for any quantity from 1 to 5 but \$9.95 each when the quantity was 6 or more then you would enter the price as follows:

- 10.95,1:9.95,6

If you set the quantity of the first price to a number above 1 then that becomes the minimum quantity that can be ordered and if the product is added to the cart with a quantity lower than the minimum, the quantity will be increased to meet the minimum.

# 6.3 Options

phpCart supports an unlimited number of options for each product. In addition, each option can have a price (adder) associated with it. You must use the html form method for creating product definitions to use product options.

For example, you can have an option that determines the size of the item. The choices for that option might be small, medium or large. In addition, you can assign a different price for each choice such as \$0.00, if the customer chooses the small choice, \$1.95 if the customer chooses the medium choice or \$2.95 if the customer chooses the large choice.

There are three different ways to send options to the cart:

- use the format "option[]". Using an array format is the quickest and easiest to set up options.
- use the format "optionX" (where X is a unique number for each option). This is required if you want to use radio buttons as your option.
- use the format "optionX[]". This is required when the option is a text field (see below).



When setting up options you will need to use an html form to submit all the cart data (see the section above labeled: Defining Products). Inside the form you will need to create the choices for the customer to select. You can present these choices to the user in several ways:

- Select (drop down) Box
- Check Box
- Radio Button
- Text Box

#### 6.3.1 Select (drop down) Box Option

You will use the "select" field to create each product option and the "option" field for each choice that you want to make available to your customer. The "value" parameter of the "option" field needs to list the text of the option and can also include an additional charge to be applied for that option. The charge must be separated from the text by a colon. Also, if the text of the option contains spaces or special characters, you must url encode it (use the urlencode() function in php) or manually replace any spaces with the %20 string.

#### Example:



### 6.3.2 Check Box Option

Using a check box option is very simple. Each check box represents a different option. The format of a check box option is: Name:Price:HashKey.

#### Example:

<input type="checkbox" name="option[]" value="Gift Wrap:5.95:"> Gift Wrap
<input type="checkbox" name="option[]" value="Express Delivery:8.95:">
Express Delivery

Or

<input type="checkbox" name="option1" value="Gift Wrap:5.95:"> Gift Wrap
<input type="checkbox" name="option2" value="Express Delivery:8.95:">
Express Delivery

# **6.3.3 Radio Button Option**

Radio buttons typically come in groups so you can only use the "option1" format when creating your radio button field. The format of a radio box option is:

Name:Price:HashKey.

#### Example:

Color:

<input type="radio" name="option1" value=" Blue::"> Blue <input type="radio" name="option1" value=" Red::"> Red

<input type="radio" name="option1" value="Yellow::"> Yellow

If you have a second set of options you would add them as follows:



#### Finish:

<input type="radio" name="option2" value="Smooth Finish::"> Smooth Finish
<input type="radio" name="option2" value="Rough Finish::"> Rough Finish

### **6.3.4 Text Box Option**

This text box option is slightly more difficult because you cannot control what information will be passed to the cart. You can use a combination of both naming formats and must include a second hidden field when you set up a text box option. The format of a text box option is: Name:Price:HashKey.

#### Example:

```
<input type="text" name="option1[]">
<input type="hidden" name="option1[]" value="Engraving:5">
```

# 6.3.5 Making an Option Mandatory

Options can now be made "mandatory" by inserting a extra hidden form field into your product definition that uses an "!" (Exclamation point), before the word, "option" in the html name code.

Example for Drop Down Boxes:

- <select name="!option[]" size="1">
- <option value="">Select Your Color</option>

Example for Check Boxes or Radio Boxes:

- <input type="hidden" name="!option1" value="">
- <input type="hidden" name="!option[]" value="">



If a customer leaves a mandatory option blank, then the error message, MissingOption, will be displayed. This message is located in the Localization directory and you can change it to your desired message.

## 6.4 Shipping

#### By Product:

It is easy to add variable shipping/handling/postage fees by product when adding an item to your shopping basket. You can add a fee to each item by including the shipping variable. This amount will be added for each quantity. If the customer orders five of an item then this amount will be added five times.

Use the shipping1 variable to set the shipping fee for the first item when the quantity field is set to higher than one. An example would be setting shipping1 = 2.95 for product A. If the customer orders (on the same line) a quantity of three then the shipping for all three would still be \$2.95. This is different from the "shipping" variable which would charge \$8.85 for shipping three items.

Use the shipping2 variable to set the shipping fee for each additional item after the first item. An example would be setting shipping1 = 2.95 and shipping2 = 1.95. If the customer orders (on the same line) a quantity of three then the shipping for all three items would be \$6.85 (2.95 + 1.95 + 1.95).

#### By Weight:

You can assign a shipping fee based on the weight of your products. You must do two things to make this work. First, you must assign a weight variable to the product.

Second, you must set up the Shipping By Weight table in the configuration screen. To assign the weight variable just add weight = 5.25 to your product link. When you set up



the Shipping By Weight table you must enter a list of weights and charges using the following format:

weight:fee;weight:fee;weight:fee

Enter a weight:fee pair (separated with a colon) and then add more weight:fee pairs (separated with a semi-colon). Make sure the first weight:fee has a weight of zero if you want a charge to be set for orders between 0 and the first weight.

Below is an example:

0:1.95;10:5.95:25:10.95:50:15.95

In the above example the following shipping charges for the order would be:

- If the total weight of all items was greater than 0 but less than or equal to 10 then charge \$1.95
- If the total weight of all items was greater than 10 but less than or equal to 25 then charge \$5.95
- If the total weight of all items was greater than 25 but less than or equal to 50 then charge \$10.95
- If the total weight of all items was greater than 50 then charge \$15.95

Note that with the above scenario if the total weight value of all items in the shopping cart = 0 or there is no weight value then the amount charged for shipping will be zero unless another shipping rule adds a shipping charge.

### 6.5 Taxes

Once you have completed the Tax Configuration in the Admin Control Panel and decide that you want product specific taxes, all you need to do is to tell the cart that you want to charge taxes on this product.



There are two ways to add product specific taxes to your cart.

- 1) Using the Tax Tables
- 2) Adding a defined tax percentage to each product

#### 6.5.1 Tax Tables

If you're using link-based product definitions, you would add one of these lines of code into your product definition:

&taxrateid=1

&taxrateid=2

&taxrateid=3

If you choose taxrate 1 or 2, the tax will be calculated using the tax percentage from that column. If you choose taxrate 3, you can get the tax values for tax1, tax2 and total tax which is calculated by adding tax 1 and tax 2 together.

If you are using form-based product definitions, you would add one of these lines of code inside the <form> definition:

```
<input type="hidden" name="taxrateid" value="1">
<input type="hidden" name="taxrateid" value="2">
<input type="hidden" name="taxrateid" value="3">
```

If you choose taxrate 1 or 2, the tax will be calculated using the tax percentage from that column. If you choose taxrate 3, you can get the tax values for tax1, tax2 and total tax which is calculated by adding tax 1 and tax 2 together.

You must use the proper template tags if you want more than just the total tax to be displayed In the cart, billing, and confirmation pages and the admin and customer order notification emails.



#### 6.5.2 Pre-defined Tax Amount

You can also add a pre-defined tax amount to each product as it is entered into the shopping cart. This tax amount would be in addition to the tax rate tables if you've selected them.

If you're using link-based product definitions, you would add one of these lines of code into your product definition:

&tax=5

The number in the above line will be become a tax percentage (.05 or 5%).

If you are using form-based product definitions, you would add one of these lines of code inside the <form> definition:

<input type="hidden" name="taxrate" value="5">

The value in the above line will be become a tax percentage (.05 or 5%).

### 6.6 Extra Variables

You can pass up to three extra variables to the shopping cart when you add a product.

These variables can be used for anything and are fully supported by the template system. The extra variables are:

- extra1
- extra2
- extra3



When you modify a template to display one of the three extra variables you can wrap a TS Comment around the TS Tag. If the variable is empty, the comment and tag will be removed. You can find out more about how to implement the extra variables by referring to the Template System documentation.

The default templates already use the "extra1" variable as a way to pass in a return URL to the product detail page. This allows the customer to click on a link in the shopping cart to return back to the product detail page.

If you do not include any information in the "extra1" variable then the link will not be displayed. If you want to use the "extra1" variable for something else, you can simply modify the cart.php file and remove the existing html code and add your own.



# 7 Customizing phpCart

In addition to being able to modify the templates for use with phpCart, there are also several files that are useful for changing the operation of phpCart. These files are stored in the Modules directory.

## 7.1 Allowing Customers to add Multiple Products to Cart

Typically each time a customer adds a product to the shopping cart, they are taken to the shopping cart page. They press the "Continue Shipping" link to return to your product definition pages to select additional products.

However, you can choose to allow them to select multiple products at one time before going to the shopping cart page. To use this method, you will need to use a slightly different product definition format and will need your form's action definition to send the customer to a file named phpcart-m.php. This file will be found in the standard distribution and will need to exist on your server to use it.

The result is that each product definition will have a checkbox (or a radio box) and your customers will select products by clicking on the checkbox. You will also need to locate a new "add to cart" button on your page, which the customer will use when they have finished making their product selections.

To set up your cart to use this method, follow these instructions:



- 1) Ensure that the phpcart-m.php file is on your server and located in the same directory as phpcart.php
- 2) Create your form action to post to this file: <form method="post" action="phpcart-m.php">
- 3) You must create your form fields as checkboxes or radio buttons.
- 4) You must create your fields using the following syntax <input type="checkbox" name="item\_xxx" value="id|description|price|quantity|shipping1|shipping2|weight|tax|taxid|extra1| extra2|extra3">
- 5) If you want to allow customers to override the quantity then create a matching form field. The quantity textbox must have the following naming structure:

  <input type="textbox" name="qty\_xxx" value="1">

NOTE: The "xxx" must match the same "xxx" used in the item xxx form field.

#### **EXAMPLE**:

<input type="checkbox" name="item\_item1" value="PRG-101|Particle
Beam|149.95|1|0">
<input type="textbox" name="qty\_item1" value="1">

<input type="checkbox" name="item\_item2" value="RS-200|Rocket Ship|44.95|1|4.95"> <input type="textbox" name="qty\_item2" value="1">

6) To use options, name your option fields: opt xxx[]

#### Example:



- <select name="opt\_item1[]">
- <option value="red">Red</option>
- <option value="blue">Blue</option>
- <option value="green">Green</option>
- </select>
- <select name="opt\_item1[]">
- <option value="small">Small/option>
- <option value="medium">Medium</option>
- <option value="large">Large</option>
- </select>
- 7) You must set a return page by setting a hidden form field called "url". If this is not set then you will be sent to the phpcart shopping cart to view the contents.

#### **EXAMPLE**:

<input type="hidden" name="url" value="/catalog/products.php">

NOTE: This process currently allows for duplicate entries to be entered into the shopping cart even if "No Duplicates" is set in the control panel.

### 7.2 Adding Variables to the Billing Form

Any form variable you add to the billing template (billing.php) will be available on all other templates (except the cart.php template).

You can easily have clients submit additional information on the billing screen and then add that information to the admin's email template.



An example might be to add a drop-down select box listing where the customer found your site (such as search engine, print ad, mailer, etc).

If the name of the field was "referredby" then you could add this field to the admin email with the TS Tag %%REFERREDBY%%.

# 7.3 Coupons

The coupons.inc.php file allows you access to all the coupon functions. You can modify this file to customize the operation of Coupons in your cart.

Coupons are held in an array in the file: admin/coupon 1.php

The array consists of 7 elements:

- 1) COUPONCODE Definiton: Text code to identify coupon
- 2) COUPONTYPE Types of coupons:
  - Free Shipping = free shipping
  - Shipping = discount shipping
  - % = percent discount
  - \$ = fixed price discount

3) COUPONAMOUNT Definition: Maximum amount of discount

4) COUPONMINIMUM Definition: Minimum order that must be reached for coupon

to be activated



5) COUPONEXPIRES Definiton: The day in which a coupon expires. Shown in

format: YYYYMMDD

6) COUPONQUANTITY Definition: This field is used as quantity of coupons available for use. If quantity is blank, it is unlimited. If it is zero, then no coupons are available for use. If quantity > 1, then coupon can be used.

7) COUPONEXTRA Extra data field for custom use

The last element in the coupon array is not used today and is left available for customizing.

# 7.4 Shipping

The shipping inc.php file is the file accessed by the cart to determine the shipping price for your cart. You can modify this file for custom shipping options.

In addition, a couple of other shipping files for you to use as models in creating your own shipping solutions. They are:

- shippingbyweight.inc.php This file has some excellent code if you want to change your shipping options to ship by state or country or to change the shipping by weight options.
- shippingbyzone.inc.php This file has some excellent code if you want to create a rather more complex shipping solution. It uses multi-dimensional arrays to get a shipping solution using states, regions or countries to create zones each with their own shipping price based on the cart total price.



### 7.5 Tax

The tax.inc.php file allows you to access all the tax functions. You can modify this file to customize the way that Taxes are calculated in your cart.

### 7.6 Validation

The validation.inc.php file contains all the checks made if you have enabled Validation checking in the Admin Control Panel. You can modify this code to change its behavior.

This file also contains the function that checks if an email address is a valid email address.

### 7.7 Misc.

The misc.inc.php file contains a wide range of useful functions that interact with the key phpCart operations.

The **CreateOrderFile** function is the function that stores a copy of each order into your file system so that they can be accessed by the View Order function within the Admin Control Panel.

The **CreateOrderID** function assigns a random number to each order as its invoice number. You can customize this function to provide invoice numbers that have custom prefixes, suffixes or even sequential numbering schemes.



The **GetOptionText** function unserializes all the options that have been selected by your customer and presents them with a hyphen (-) between them. You can customize how you want your options to appear in your cart.

The **Evaluate** function uses the php eval function to evalute a string of text as php.

The **Totals** function is called each time that the cart shows the totals of all the items selected by your customer. This function is useful if you want to include a price adder to the total. One example would be an extra shipping and handling charge.

The **OrderComplete** function is called just before the thank-you page is displayed. If you are using a Remote Gateway, this function is called when the customer is returned to your store with a successful transaction. Here is the proper URL:

www.yourdomain.com/phpcart.php?action=complete

The **OrderPending** function is called by a remote gateway if an order has been held as pending on the Gateway's web site and they offer a pending URL option. The customer will be returned to your store with an "order pending" transaction. Here is the proper URL:

www.yourdomain.com/phpcart.php?action=pending

The **OrderCanceled** function is called by a remote gateway if an order has been canceled on their web site and they offer a cancel URL option. The customer will be returned to your store with an "order canceled" transaction. Here is the proper URL: www.yourdomain.com/phpcart.php?action=canceled

The **OrderDeclined** function is called by a remote gateway if an order has been declined on their web site and they offer a declined URL option. The customer will be returned to your store with an "order declined" transaction. Here is the proper URL:



www.yourdomain.com/phpcart.php?action=declined

The **SendEmail** function is used to determine which mode of email you have selected in the Admin Control Panel. The options are: php mail or SMTP mail.

The **SendPHPMail** function creates the format of the php mail to be sent by the system. This function allows you to change any of the headers for your own custom solution.

The **SendSMTPMail** function creates the format of the SMTP mail to be sent by the system. This function allows you to change any of the headers for your own custom solution.



### 7.8 Advanced PHP Users

Advanced PHP users may want to extend phpCart in ways to suit their special circumstances. Some techniques may be:

- Use a mySQL database for listing products and creating product pages
- Use a mySQL database for storing records about customers and their orders.
- Use a mySQL database to manage inventory.

If you are not an Advanced PHP user but want to discuss options like these, please contact us at sales@phpcart.net

## 7.8.1 Data Arrays

The phpCart system uses three key data arrays to hold transaction and customer information. These data arrays can be used by Advanced PHP users to create custom solutions for their customers. They are:

- config (lists the settings from the Admin Control Panel)
- billing (lists the customer's shipping/billing info)
- totals (list the contents of the cart and the prices)

You can see if these arrays are available in your script and their contents by using php code like this:

```
echo '';
print_r ($config);
echo '<br>';
echo '';
print_r ($billing);
echo '<br>';
```



```
echo '';
print_r ($totals);
echo '<br>';
```

#### 7.8.2 Sessions

The phpCart system uses php Sessions to store transaction data when a customer is moving from the cart to the checkout screen to the confirmation screen while using your cart.

The data arrays above are put into the Sessions directory so that you can access them in any of these three cart screens.

## 7.8.3 Outputting Arrays to a Database

In phpCart version 4.8 and above, we have added a new function to the modules/misc.inc.php file so that you can gain access to the output of a purchase that has been confirmed by the customer. In that file, you will find this new function:

function UpdateDB(\$billing, \$data, \$product data)

- billing (lists the customer's shipping/billing info)
- data (lists the cart totals information: subtotal, shipping, tax)
- product data (list the contents of the cart and the prices)

You can add code into this function which will allow you to manipulate this purchase information including storing it into a database.



## 8 Additional Cart Features

## 8.1 Using SSL Certificates for securing your cart

A SSL Certificate is a small data file that digitally binds a cryptographic key to an organization's details. When properly installed on your web server, it displays the padlock symbol in your customer's browser. The URL displayed in their browser address bar will start with https and this creates a secure connection from your web server to the customer's browser.

phpCart is designed to work with your installed SSL Certificate whether that is your own personal SSL Certificate or a shared SSL Certificate.

To use an SSL Certificate with phpCart, you will need to use the "altdomain" location in the phpCart License file.

Here is an example of a phpCart License File with an SSL Certificate identified:

```
2
        $License = array
 3
            "product"
"domain"
                        => "PHPCart Single License",
4
5
                             => "webrigger.com",
            "altdomain"
6
                             => "secure.webrigger.com",
            "version"
7
                              => "4.60",
            "expires"
                              => "Never",
8
            "supportexpires" => "2008-04-05 00:00:00",
9
10
            "items"
                              => "0",
            "accounts"
                              => "0",
11
12
            "license"
                              => "e45c-7d8f-1ec2-29b0"
13
        );
    ?>
14
```



## 8.2 Maintenance Mode (Store Closed)

phpCart includes a maintenance mode for your store where you can "close" the store to work on it or just make it unavailable to the general public. Anyone with the exception of logged in administrators that are visiting the site will see a "Store Closed" message.

To put your store into maintenance mode, go to the Admin Control Panel, select, Cart Configuration from the drop down box and then set the "Maintenance Mode" setting to "Yes".

People trying to use your store will be presented the "Store Closed" template. The template is displayed within the main template page and you can change its contents. It is located at: template/generic/store\_closed.php.

You can also add the following php code to the top of your product definition pages if you want them to also check if the store is open and to display the store closed message if it is closed. If your product definition pages have a .html or .htm suffix in their name, you will need to update the .htaccess file to enable execution of php on those page.



## 8.3 Creating Product Definitions using the Button Maker Tool

You can create your store's product definitions manually as described in sections 6.1 and 6.2 or you can use our automated product definition creation tool, Button Maker.

The Button Maker allows you to fill out online forms and you will automatically be presented with hyperlink and/or HTML form code that you can use on your product pages. If you are using HTML forms, you can also create product options with drop down boxes, check boxes and radio boxes.

The Button Maker will also allow you to create custom hyperlink verification tags for your products and options if you are manually creating them.

Once you create the "Buy Now" buttons, just copy the html code and paste it into your product page. It's as simple as that!

With the Product Creator tool you can create:

Hyperlink buttons using a text link (no options)



	Нуре	erlink Button
Path to phpCart:		/phpcart/phpcart.php
Product ID:		0001
Product Name:		Blue T-Shirt
Price:		0.00
Link Text: (e.g. Add to Cart)		Add to Cart!
	Make Button	Reset
Currency: (Without selection, cart will use currency)	primary	Select Currency \$
Weight:		
Shipping: (each item)		
Shipping1: (first item)		
Shipping2: (each additional item)		
Tax Rate ID:		None 💠
	Make Button	Reset



2. Hyperlink buttons with a "Buy Now" image (no options)

	Hyperlink Button with Image
Path to phpCart:	/phpcart/phpcart.php
Path to image:	/images/addtocart.gif
Product ID:	0001
Product Name:	Blue T-Shirt
Price:	0.00
	Make Button Reset
Currency: (Without selection, cart will use p currency)	primary Select Currency \$
Weight:	
Shipping: (each item)	
Shipping1: (first item)	
Shipping2: (each additional item)	
Tax Rate ID:	None 💠
	Make Button Reset



3. HTML form buttons (the most popular way)

HTML Fo	orm Button
Path to phpCart:	/phpcart/phpcart.php
Product ID:	0001
Product Name:	Blue T-Shirt
Price: (e.g. Single price or range (x.00,1:y.00,5))	0.00
Button Text: (e.g. Add to Cart)	Add to Cart
Quantity:	Selectable   Not selectable   ●
Quantity Text: (e.g. Quantity)	Quantity
Make Button	Reset
The below variables	are all optional.
Currency:	
Currency: (Without selection, cart will use primary currenc	
Currency: (Without selection, cart will use primary currency Weight: Shipping:	
Currency: (Without selection, cart will use primary currence) Weight: Shipping: (each item) Shipping1:	
Currency: (Without selection, cart will use primary currence Weight: Shipping: (each item) Shipping1: (first item) Shipping2: (each additional item)	

4. Product and Option Hyperlink Verification code for your manually created products

Hyperlink Verification Key		
Enter your values here:	Make Key Reset	



## 8.4 Geo IP

phpCart includes Geo IP functionality. The Geo IP function uses the customer's IP address to identify their country, city, etc. using the free GEO IP Database from Maxmind.

The files and "flat IP database" are located in the "geoip" directory located inside the modules directory. You can update the free database at any time by going to the Maxmind web site and downloading the latest free database.

You can also purchase GEO IP databases to gain access much more information.

The GEO IP information can be used or displayed in the cart's Billing page, the Confirmation page and during the order submission, which includes both the admin and customer emails by using these new template tags:

- a) GEO\_COUNTRY
- b) GEO\_CITY
- c) GEO LAT
- d) GEO LONG

If you used the standard phpCart template tag format, %%TEMPLATE\_TAG%%, on any of the above pages, you will see the customer geo information when that page is used.

### 8.5 Internationalization

The text in phpCart is in English by default. In addition, we now support French, Italian and Spanish with thanks to our members. There are just a few phrases used throughout



phpCart that cannot easily be put in the template system so they are put in a language file that is easy to edit.

You can translate the included language file to your own language if there is no translation available at this time. To do so, make a backup copy the file "english.php" in the /localization directory.

Now, just edit the file and replace every string within the double-quotes to the equivalents in your language. Save the file in the /localization directory.

To activate your language, just make sure you have selected the English language in the configuration screen of the control panel.

## 8.6 Display the Mini-cart on other Pages

You can easily display the cart total and the number of items in the cart on other web pages by using a Mini-cart. You can control the layout of this information by editing the minicart.php template file (located in the phpcart/templates/\*\*\*yourtemplate\*\*\* template directory).

Below are the steps required to add the cart total to any web page:

- 1) The page must have a ".php" extension or the php code will not work.
- 2) The page must call the php function session\_start() as the first line of code. Example:
  - <? session\_start(); ?>
- 3) Edit the template file *minicart.php* to match your site layout



- 4) Include the phpcart-display.php file exactly in the code where you want the cart total to appear. For example:
  - <? include("phpcart/phpcart-display.php"); ?>

The following variables are available to be displayed on the mini-cart page:

- %%PRODUCTCOUNT%% The number of products in the cart (multiple quantities of the same item do not increase the product count)
- %%ITEMCOUNT%% The total number of items in the cart (multiple quantities of the same item increase the item count)
- %%RAWSUBTOTAL%% The unformatted subtotal of the cart (this does not include any shipping or tax but does include options)
- %%RAWTAX%% The unformatted tax calculated for the cart (this may change once the shopper selects a taxable region during checkout)
- %%RAWSHIPPING%% The unformatted shipping calculated for the cart (this
  may change once the shopper selects a shipping region during checkout)
- %%RAWDISCOUNT%% The unformatted discount if a coupon has been submitted
- %%RAWTOTAL%% The unformatted total of the cart
- %%SUBTOTAL%% The currency-formatted subtotal of the cart
- %%TOTALTAX%% The currency-formatted tax calculated for the cart
- %%TOTALSHIPPING%% The currency-formatted shipping calculated for the cart
- %%DISCOUNT%% The currency-formatted discount calculated for the cart
- %%GRANDTOTAL%% The currency-formatted total of the cart



## 8.7 2013 Canadian Taxes

The new 2013 Canadian Tax functionality has been to the cart. It now supports the ability to display multiple rate calculations on the cart, billing, confirmation pages and order notification emails.. This feature was added to support the BC, Canada requirement to display PST (provincial tax), GST (government tax) and HST (harmonized tax).

This new feature is support by:

- a) Allowing taxrateid to be set to 3. If '3' is selected, then the tax script adds taxrate1 and taxrate2 together to create the total tax.
- b) To display the tax1, tax2, and total tax values, the software was updated to use new Template Tags. These Template Tags are used to display all the separate tax values.
  - c) New template tags were created to enable this feature to work:
    - <!-- BEGIN TAXTITLE1 --> %%TAXTITLE1%%
    - <!-- END TAXTITLE1 -->
    - <!-- BEGIN TAXTITLE2 -->
      %%TAXTITLE2%%

<!-- END TAXTITLE2 -->

These template tags are used to display or not display the separate tax titles.

<!-- BEGIN SEPARATETAX1 --> %%SEPARATETAX1%%



<!-- END SEPARATETAX1 -->

<!-- BEGIN SEPARATETAX2 --> %%SEPARATETAX2%%

<!-- END SEPARATETAX2 -->

These template tags are used to display or not display the separate tax values.

d) The tax calculation script is located at: modules/tax.inc.php and is not encoded to allow you to change the functioning of the program to meet your specific requirements. As it's delivered it will meet the needs of most U.S. and Canadian companies.

Since this functionality is rather generic, it should be easy to adapt it to fit other countries and/or tax requirements.



## 9 Frequently Asked Questions

#### 9.1 Common Installation Problems

"I get a bunch of garbage on my screen after installation"

Make sure you upload your files via FTP in binary mode and that your web host has the Zend Optimizer or loncube module installed. This is not the same as just having the Zend Engine installed.

"Why do I get the error message 'Permission denied""?

You probably forgot to set write permission on the sessions directory. See the installation section above for more information.

## 9.2 How to update phpCart to the newest version

Here are some instructions on how to best update your system.

- 1) Make a full copy of your existing phpcart directory. This will be your backup copy. When you make the backup, name it something easy to remember for the future.
- 2) Download the new phpCart version for the type of encoder on your host (zend or ioncube). If you're not sure, you can use phpinfo(), which is explained in this User Guide.
- 3) To update do not upload all the files to the server. Only upload in BINARY MODE these items:



- admin directory
- doc directory
- images directory
- · includes directory
- localization directory
- modules directory
- processor directory
- · phpcart.php file
- phpcart-m.php file
- phpcart-c.php file
- phpcart-display.php file
- 4) The items that **DO NOT UPLOAD** are:
  - orders directory (holds all your old orders)
  - sessions directory (holds all the active sessions of your cart visitors)
  - templates directory (holds all your customized look and feel)
  - license.php (holds your current license)
  - index.php (your home page)
- 5) The last step to get everything running again is to configure your system. The configuration files are all located in the admin directory. They are:
  - a) admin\_1.php
  - b) configuration\_1.php
  - c) countries\_1.php
  - d) coupon\_1.php
  - e) payment\_1.php
  - f) processor 1.php
  - g) regions\_1.php



- And yes, you're seeing correctly, they are the only files in the admin directory that have a underscore 1 (\_1) suffix on their file names.
- There are two ways to update the configuration files.
  - a) Log into the Admin Control Panel and reset everything to look like your old system. But before you can do this, you must set the permissions on the above configuration files. The permissions must be set to 777 on these files.
  - b) If you're comfortable moving files around, you could take copies from the backup admin directory and replace these new files. You will need to make sure that the permissions for these files remain as 777.
- 6) And lastly if you use option 1 above, you'll need to reset the admin credentials including your password. If you moved the old admin\_1.php file to the new admin directory, your old control panel password will now work.

## 9.3 Common Usage Questions

- "How can I force the cart content to expire if the customer leaves my web site?"
  phpCart uses a php session variable to track customers. The cart contents will expire when the session variable expires or when the visitor closes their browser.
- "How can the customer delete all contents in his/her cart?"
   Add the following link into your web page:



<A HREF="phpcart.php?action=clear">Clear Shopping Cart</A>

"How can the customer view the contents in his/her cart?"

Add the following link in into your web page: <A HREF="phpcart.php?action=view">View Shopping Cart</A>

## 9.4 How to get help?

The best way to get help with getting your cart to run is to go the phpCart forum which can be found on our web site.

The first thing to do is to use the search function to see if others have had a similar experience as yours which may give you the answer to your question.

If you are unable to find an answer to your question, then submit a new forum thread with your question, providing the most detail that you can. The more detail that you provide, the easier it is for someone to respond to your question.

In addition to existing customers, the forum is monitored by a phpCart customer service professional. Between this person and other helpful phpCart customers, we will make sure to provide you with support to answer your questions.

If you have a question that is not technical in nature or is a bit out of the ordinary, then you should send a detailed email to this email address: sales@phpcart.net





## 10 Troubleshooting Techniques

# 10.1 Using phpinfo() to determine your server's version of PHP and its encoders

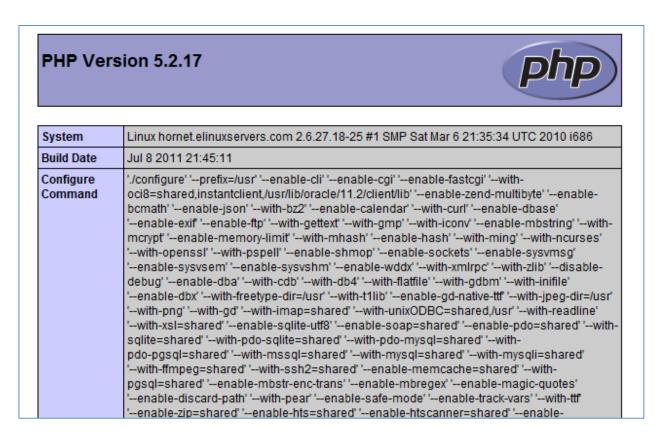
If you are uncertain which version of php that your server is running or which encoders are available on your server, you can run the php function called phpinfo().

To do this, create a new file on your server and insert this code into it. Upload the file to your server with FTP and then access the file with your browser.

```
<?php
phpinfo();
?>
```

Your browser will then display the standard php info records for your server's configuration. It will look something like:





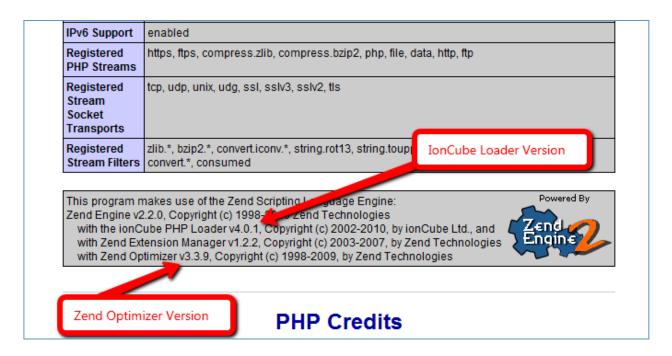
You can see at the top of this image the version of PHP that your server is running.

## 10.2 Encoding Problems

If you FTP a copy of phpCart to your server but see a bunch of special characters that look like random garbage then there is an encoder problem. You must download the proper encoded version of phpCart.

You can check if either Zend Optimizer or loncube module is available on your server by using the phpinfo (above). Look for the Zend Scripting Language Engine block. It will look like this:





In this example both Zend and loncube are available and their version numbers are pointed out.

## 10.3 Licensing Problems

Each copy of phpCart must be licensed in order to operate. The license file must be installed on your server. It is best located in the same directory as the phpcart.php file. But it can also be located in a directory that is one or two levels higher in the directory tree.

The license file is protected by a key which is calculated when the license file is created. You cannot make any changes in the license file or it will not work. In order to change any values in the license, you must create a new license.



A series of tests are run on the license to ensure that the license file is real. They include:

- Has the license file been located?
- Is the license key valid?
- Are the domain or the altdomain values in the license the same as defined by the php command: \$\_SERVER["SERVER\_NAME"]?
- Is the version valid?

The value of the domain field should be your domain without any subdirectory including the common www subdomain prefix.

The value of the altdomain field should be your SSL domain (if you are using one) and should include any subdomain that you are using for your SSL access.

If are having problems, the first test is to see if phpCart can find a copy of your license by using this command:

www.yourdomain.com/phpcart.php?action=license

Then you can see if the domain and altdomain values are correct. To do this, create a new file on your server and insert this code into it. Upload the file to your server with FTP.

```
<?php
echo 'server name is: '.$_SERVER["SERVER_NAME"];
?>
```

To test your domain, access this file from your browser using its http://www.yourdomain.com URL address.

To test your SSL domain, access this file from your browser using its https://www.yourdomain.com URL address.



These values must match the ones located in your license file.

## 10.4 Template Problems

We highly recommend using the prototype templates to conduct a test of the system's operation after your installation. This test will tell you if the phpCart installation was successful.

If you modify a template and the cart stops working, we recommend that you re-install the prototype template to determine if the problem is located in the phpCart software or the new template. By using the prototype template, you can quickly identify the location of the failure.



## 11 Copyright Notice

Please note that all files in this distribution are copyrighted by the Author. You may not re-distribute any portion of the distribution without the expressed permission of the Author.