



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

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUCTION .....</b>                       | <b>5</b>  |
| 1.1      | INTRODUCTION .....                              | 5         |
| 1.2      | THEORY OF OPERATION .....                       | 6         |
| 1.2.1    | Cart page.....                                  | 6         |
| 1.2.2    | Checkout page.....                              | 8         |
| 1.2.3    | Confirmation page .....                         | 10        |
| 1.3      | SYSTEM REQUIREMENTS.....                        | 14        |
| <b>2</b> | <b>INSTALLATION .....</b>                       | <b>15</b> |
| 2.1      | INSTALLATION .....                              | 15        |
| 2.2      | LICENSE FILE .....                              | 17        |
| <b>3</b> | <b>CONFIGURATION .....</b>                      | <b>19</b> |
| 3.1      | LOG INTO THE ADMINISTRATION CONTROL PANEL ..... | 19        |
| 3.2      | CHANGE ADMIN PASSWORD .....                     | 20        |
| 3.3      | PHPCART 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.....                               | 29        |
| 3.3.5    | Currency Setup.....                             | 32        |
| 3.3.6    | Payment Setup.....                              | 34        |
| 3.3.7    | Shipping Setup.....                             | 36        |
| 3.3.8    | Cart Tax Setup .....                            | 38        |
| 3.3.9    | State/Region Tax Setup .....                    | 39        |
| 3.3.10   | Country Tax Setup.....                          | 40        |
| 3.3.11   | Required Items Setup .....                      | 41        |
| 3.3.12   | Maintenance.....                                | 42        |
| 3.4      | COUPONS .....                                   | 43        |
| 3.5      | ORDERS.....                                     | 46        |
| <b>4</b> | <b>INITIAL SYSTEM TEST.....</b>                 | <b>48</b> |
| <b>5</b> | <b>THE TEMPLATE SYSTEM.....</b>                 | <b>49</b> |
| 5.1      | THE 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        |
| <b>6</b> | <b>PREPARING YOUR STORE.....</b>                          | <b>53</b> |
| 6.1      | DEFINING PRODUCTS .....                                   | 53        |
| 6.1.1    | <i>Required Product Variables</i> .....                   | 54        |
| 6.1.2    | <i>Optional Product Variables</i> .....                   | 54        |
| 6.2      | PRICING.....  | 57        |
| 6.3      | OPTIONS.....  | 58        |
| 6.3.1    | <i>Select (drop down) Box Option</i> .....                | 59        |
| 6.3.2    | <i>Check Box Option</i> .....                             | 60        |
| 6.3.3    | <i>Radio Button Option</i> .....                          | 60        |
| 6.3.4    | <i>Text Box Option</i> .....                              | 61        |
| 6.3.5    | <i>Making an Option Mandatory</i> .....                   | 61        |
| 6.4      | SHIPPING.....   | 62        |
| 6.5      | TAXES .....   | 63        |
| 6.5.1    | <i>Tax Tables</i> .....                                   | 64        |
| 6.5.2    | <i>Pre-defined Tax Amount</i> .....                       | 65        |
| 6.6      | EXTRA VARIABLES .....                                     | 65        |
| <b>7</b> | <b>CUSTOMIZING PHPCART .....</b>                          | <b>67</b> |
| 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        |
| 7.8.1    | <i>Data Arrays</i> .....                                  | 75        |
| 7.8.2    | <i>Sessions</i> .....                                     | 76        |
| 7.8.3    | <i>Outputting Arrays to a Database</i> .....              | 76        |



# 1 Introduction



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.





Coupon

Validate 

| ID  | Name  | Price   | QTY                            | Total          |
|---|---|---------|--------------------------------|----------------|
|  BEAT-01 | Music CD/Beatles<br><a href="#">Details</a> | \$14.99 | <input type="text" value="1"/> | \$14.99        |
| <a href="#">Update Quantities</a>   |   |         |                                |                |
| <b>Sub Total</b>  |   |         |                                | \$14.99        |
| <b>Shipping</b>   |   |         |                                | \$10.00        |
| <b>Tax</b>  |   |         |                                | \$0.00         |
| <b>Total</b>  |   |         |                                | <b>\$24.99</b> |

[Continue Shopping !\[\]\(376aec4e860706c057eb995ec5b07949\_img.jpg\)](#)
[Clear Cart !\[\]\(c865c1cfbf4372e95eb3c0648a6f83ca\_img.jpg\)](#)
[Checkout !\[\]\(7dea90c3b410ba855b14e5946ab4a7db\_img.jpg\)](#)

Copyright © 2006 Carmosa. All Rights Reserved  
phpCart.net - php Shopping Cart

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:



**Please verify your order.**

Complete the form below and click on the button below to complete this order.

**Note:** You will be taken to our Secure 128Bit Payment Gateway Server.


|                |                      |
|----------------|----------------------|
| First Name     | <input type="text"/> |
| Last Name      | <input type="text"/> |
| Email Address  | <input type="text"/> |
| Company        | <input type="text"/> |
| Address        | <input type="text"/> |
| Address Line 2 | <input type="text"/> |
| City           | <input type="text"/> |
| State          | Choose a State ▼     |
| State Other    | <input type="text"/> |
| Zip Code       | <input type="text"/> |
| Country        | Choose a Country ▼   |
| Phone          | <input type="text"/> |











### Please verify your order

|                |                    |
|----------------|--------------------|
| First Name     | Perfect            |
| Last Name      | Customer           |
| Email Address  | perfect@domain.net |
| Company        |                    |
| Address        | 14 Oak Street      |
| Address Line 2 |                    |
| City           | New York City      |
| State          | NY                 |
| State Other    |                    |
| Zip Code       | 06578              |
| Country        | United States      |
| Phone          | 212-555-1212       |
| Payment Method | Check/Money Order  |

[View Cart](#)
[Edit Billing](#)

Click To Order

|           |         |
|-----------|---------|
| Sub Total | \$14,99 |
| Discount  | \$0,00  |
| Tax       | \$0,00  |
| Shipping  | \$12,95 |
| Total     | \$26,00 |

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:



- 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.





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***.
2. 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***





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:

```
1  <?
2      $License = array
3      (
4          "product"          => "PHPCart Single License",
5          "domain"           => "webrigger.com",
6          "version"          => "4.60",
7          "expires"          => "Never",
8          "supportexpires"   => "2008-04-05 00:00:00",
9          "items"            => "0",
10         "accounts"         => "0",
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.*







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 preceded 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 Store Time:: Feb 27th, 2013 7:18:40 am

**Company Name:**

My Company's Name

**Company Address:**

123 Oak Street  
Suite 101A  
Box 444  
My Town, CA 91234

**Company Website:**

[www.mycompany.com](http://www.mycompany.com)

**Company Phone:**

123-456-7890

**Company Email:**  
(Displayed on client emails, etc)

sales@yourdomain.com

**Company Admin Email:**  
(Email to receive receipts,  
orders, etc)

sales@yourdomain.com

**Additional Order Emails:**  
(Orders will be emailed to these also.  
Separate with a comma and space)

**Homepage URL:**

[index.html](#)**Time Zone Offset:**

Default (GMT -5 hours) ⌵

**Maintenance Mode:**  
(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.









**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.





Add New Currency

Currency Name: \*

Currency Code: \*  
(used in product definition)

Currency Display: \*

Primary Currency:

☐

Conversion Rate:

PayPal Code:

Select PayPal Code ▾

Save the currency

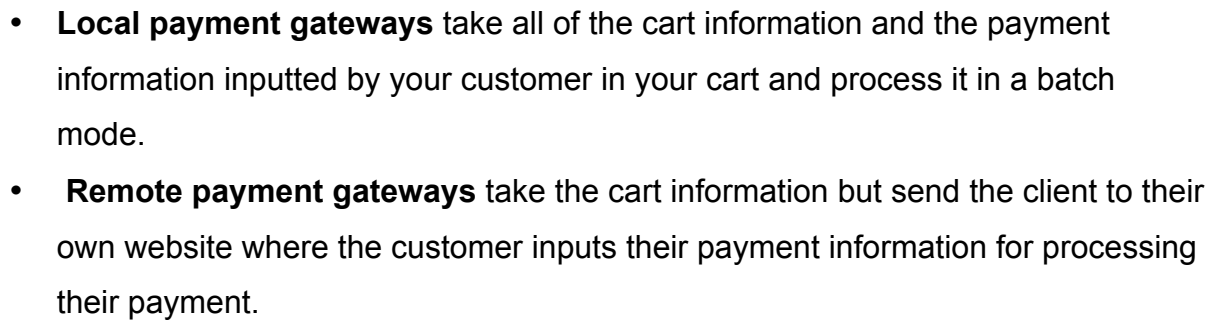
(Items with \* are required)

Note:

**Conversion Rates:** Leave conversion rate blank if you provide separate prices for each currency in your product definition. Use the conversion rate when you provide a single price and a currency code in your product definition. phpCart will do the conversion automatically for you.

**PayPal Code:** Select the PayPal code to match your country in order to use the PayPal gateway.

### 3.3.6 Payment Setup



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.

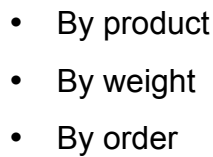
### 3.3.7 Shipping Setup

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:

**Shipping/Handling Per Order:**

113

11

11

11

11

11

11

\_\_\_\_\_

(Example: 0:2.50;5:5.00;10:8.75  
The above will charge 2.50 if the weight is over 0, 5.00 if the weight is over 5 and 8.75 if the weight is over 10.)

11

11

11

11

Save the settings

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.



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.

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



## States/Regions Tax Configuration

| ID | State/Region Name                                 | State/Region Abbreviation       | Taxrate 1                      | Taxrate 2                      | Delete                              |
|----|---|---------------------------------|--------------------------------|--------------------------------|-------------------------------------|
| 1  | <input type="text" value="Alberta"/>              | <input type="text" value="AB"/> | <input type="text"/>           | <input type="text" value="5"/> | <input checked="" type="checkbox"/> |
| 2  | <input type="text" value="British Columbia"/>     | <input type="text" value="BC"/> | <input type="text" value="7"/> | <input type="text" value="5"/> | <input checked="" type="checkbox"/> |
| 3  | <input type="text" value="Alabama"/>              | <input type="text" value="AL"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 4  | <input type="text" value="Alaska"/>               | <input type="text" value="AK"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 5  | <input type="text" value="Arizona"/>              | <input type="text" value="AZ"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 6  | <input type="text" value="Arkansas"/>             | <input type="text" value="AR"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 7  | <input type="text" value="California"/>           | <input type="text" value="CA"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 8  | <input type="text" value="Colorado"/>             | <input type="text" value="CO"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 9  | <input type="text" value="Connecticut"/>          | <input type="text" value="CT"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 10 | <input type="text" value="Delaware"/>             | <input type="text" value="DE"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 11 | <input type="text" value="District of Columbia"/> | <input type="text" value="DC"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |
| 12 | <input type="text" value="Florida"/>              | <input type="text" value="FL"/> | <input type="text"/>           | <input type="text"/>           | <input checked="" type="checkbox"/> |

### 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.







## Required Items 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.



## Routine Maintenance

**Be careful as these are permanent changes and cannot be reversed**

## Clear Customer Carts

- ☒ Older than 1 Month  
☐ Older than 1 Week  
☐ Older than 1 Day  
☐ All

### Clear Customer Carts

**Clear Order History**

- ☒ Older than 1 Year  
☐ Older than 1 Month  
☐ All

[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.









Selecting the “X” icon will delete the Order from the list.









- 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 (`<td>%%DATA%%</td>`) 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 [phpCart Template System \(TS\)](#) documentation.

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.





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 Now 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.





- 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&taxrateid=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.







```
</select>
<select name="option[]">
<option value="Small">Small</option>
<option value="Medium:3:">Medium - $3.00</option>
<option value="Large:5.95:">Large - $5.95</option>
</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:

```
<select name="option[]">
  <option value="Size Small:.">Small</option>
  <option value="Size Large:2.50:.">Large - $2.50</option>
</select>
```

Or

```
<select name="option1">
  <option value="Size Small:.">Small</option>
  <option value="Size Large:2.50:.">Large - $2.50</option>
</select>
```



### 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.









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|shipping|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">
```

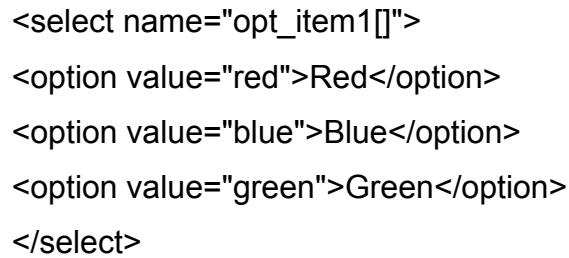
```
<input type="textbox" name="qty_item1" value="1">
```

```
<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:



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.

## 7.2 Adding Variables to the Billing Form

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` Definition: 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.









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.

**Version 1.4**



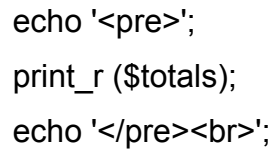
- Use a mySQL database for listing products and creating product pages
- Use a mySQL database for storing records about customers and their orders.
- Use a mySQL database to manage inventory.

### 7.8.1 Data Arrays

- 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)

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

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



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

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.

### 7.8.3 Outputting Arrays to a Database

```
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)

## Version 1.4







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.

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!

### 1. Hyperlink buttons using a text link (no options)



## Hyperlink 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

**The below variables are all optional.**

**Currency:**

(Without selection, cart will use primary currency)

Select Currency

**Weight:**

**Shipping:**  
(each item)

**Shipping:**

(each item)

**Shipping1:**

(first item)

### Shipping2:

(each additional item)

**Tax Rate ID:**

None

Make Button

Reset











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



&lt;!-- END SEPARATETAX1 --&gt;

&lt;!-- BEGIN SEPARATETAX2 --&gt;

%%SEPARATETAX2%%

&lt;!-- END SEPARATETAX2 --&gt;

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.







- 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](mailto:sales@phpcart.net)



## 10 Troubleshooting Techniques



## PHP Version 5.2.17



|                   |  |
|-------------------|--|
| System            | Linux hornet.elinuxservers.com 2.6.27.18-25 #1 SMP Sat Mar 6 21:35:34 UTC 2010 i686  |
| Build Date        | Jul 8 2011 21:45:11  |
| Configure Command | <code>./configure '--prefix=/usr' '--enable-cli' '--enable-cgi' '--enable-fastcgi' '--with-oci8=shared,instantclient,/usr/lib/oracle/11.2/client/lib' '--enable-zend-multibyte' '--enable-bcmath' '--enable-json' '--with-bz2' '--enable-calendar' '--with-curl' '--enable-dbase' '--enable-exif' '--enable-ftp' '--with-gettext' '--with-gmp' '--with-iconv' '--enable-mbstring' '--with-mcrypt' '--enable-memory-limit' '--with-mhash' '--enable-hash' '--with-ming' '--with-ncurses' '--with-openssl' '--with-pspell' '--enable-shmop' '--enable-sockets' '--enable-sysvmsg' '--enable-sysvsem' '--enable-sysvshm' '--enable-wddx' '--with-xmlrpc' '--with-zlib' '--disable-debug' '--enable-dba' '--with-cdb' '--with-db4' '--with-flatfile' '--with-gdbm' '--with-inifile' '--enable-dbx' '--with-freetype-dir=/usr' '--with-t1lib' '--enable-gd-native-ttf' '--with-jpeg-dir=/usr' '--with-png' '--with-gd' '--with-imap=shared' '--with-unixODBC=shared,/usr' '--with-readline' '--with-xsl=shared' '--enable-sqlite-utf8' '--enable-soap=shared' '--enable-pdo=shared' '--with-sqlite=shared' '--with-pdo-sqlite=shared' '--with-pdo-mysql=shared' '--with-pdo-pgsql=shared' '--with-mssql=shared' '--with-mysql=shared' '--with-mysqli=shared' '--with-ffmpeg=shared' '--with-ssh2=shared' '--enable-memcache=shared' '--with-pgsql=shared' '--enable-mbstr-enc-trans' '--enable-mbregex' '--enable-magic-quotes' '--enable-discard-path' '--with-pear' '--enable-safe-mode' '--enable-track-vars' '--with-ttf' '--enable-zip=shared' '--enable-hts=shared' '--enable-htscanner=shared' '--enable-</code> |

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 Ioncube module is available on your server by using the phpinfo (above). Look for the Zend Scripting Language Engine block. It will look like this:



|                                     |   |
|-------------------------------------|---|
| IPv6 Support                        | enabled   |
| Registered PHP Streams              | https, ftps, compress.zlib, compress.bzip2, php, file, data, http, ftp              |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, sslv3, sslv2, tls   |
| Registered Stream Filters           | zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, convert.*, consumed |

This program makes use of the Zend Scripting Language Engine:  
 Zend Engine v2.2.0, Copyright (c) 1998-2007 by Zend Technologies  
 with the ionCube PHP Loader v4.0.1, Copyright (c) 2002-2010, by ionCube Ltd., and  
 with Zend Extension Manager v1.2.2, Copyright (c) 2003-2007, by Zend Technologies  
 with Zend Optimizer v3.3.9, Copyright (c) 1998-2009, by Zend Technologies

Powered By 

## Zend Optimizer Version

## PHP Credits

In this example both Zend and Ioncube 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](http://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.

**Version 1.4**