



The Simple E-Commerce Solution

User Guide

Your guide to using the phpCart E-Commerce System.

phpCart Version: 4.7

Date: 12/29/11



Table of Contents

INTRODUCTION.....4

INTRODUCTION4

THEORY OF OPERATION5

SYSTEM REQUIREMENTS11

INSTALLATION.....13

INSTALLATION13

LICENSE14

CONFIGURATION.....16

LOG INTO THE ADMINISTRATION CONTROL PANEL.....16

CHANGE ADMIN PASSWORD17

CART CONFIGURATION17

GATEWAY CONFIGURATION31

INITIAL SYSTEM TEST.....34

THE TEMPLATE SYSTEM.....35

THE STANDARD TEMPLATE35

CREATING YOUR OWN TEMPLATE35

CART TEMPLATES36

EMAIL TEMPLATES.....37

GATEWAY TEMPLATES37

THANK YOU TEMPLATES38

MINICART TEMPLATE.....38

PREPARING YOUR STORE.....39

DEFINING PRODUCTS39

PRODUCT OPTIONS.....43

EXTRA VARIABLES.....46



<u>CUSTOMIZING PHPCART.....</u>	<u>48</u>
ADDING VARIABLES TO THE BILLING FORM	48
COUPONS.....	48
SHIPPING	52
TAX	52
VALIDATION.....	52
MISC.....	53
ADVANCED PHP USERS	55
<u>ADDITIONAL CART FEATURES</u>	<u>57</u>
USING SSL CERTIFICATES FOR SECURING YOUR CART	57
INTERNATIONALIZATION	58
DISPLAY THE MINI-CART ON OTHER PAGES	58
<u>FREQUENTLY ASKED QUESTIONS.....</u>	<u>60</u>
COMMON INSTALLATION PROBLEMS.....	60
COMMON USAGE QUESTIONS	60
HOW TO GET HELP?	61
<u>TROUBLESHOOTING TECHNIQUES.....</u>	<u>62</u>
USING PHPINFO()	62
ENCODING PROBLEMS	63
LICENSE PROBLEMS.....	64
TEMPLATE PROBLEMS	66
<u>COPYRIGHT NOTICE.....</u>	<u>67</u>



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 product pages that you create using the product definition (below) or the automated 'Code Generator'. When a prospective customer visits your web site, they will see all the items that you make available for sale.

The phpCart software is first activated when your customer selects one of your products by pressing a "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 Details	\$14.99	1	\$14.99
Update Quantities				
Sub Total				\$14.99
Shipping				\$10.00
Tax				\$0.00
Total				\$24.99

[Continue Shopping](#)
[Clear Cart](#)
[Checkout](#)

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:

- SubTotal
- Shipping
- Taxes
- Total



The customer will then be able to:

- Change the quantity of items to purchase
- Click the trashcan 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"/>



- Customer address
- A list of the cart items including quantity and price
- 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
- Order pricing summary
- 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 phpCart software from the phpCart web site. You will need to be able to FTP the files from your local machine to the server in BINARY mode and unzip the files either on your local machine or your server.

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.



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.

PHPCart
The Simple E-Commerce Solution

Current Version: **4.7**
Latest Version: **4.7**

Admin Home

[Support Forums](#) - [Code Generator](#) - [Documentation](#)

Login Username:

Password:

Login

Copyright ©2000 -2011- [Carmosa](#).
[PHPCart.net](#)

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!

Using the drop down box labeled **Admin Home**, select **Set Password**. You will then see this screen:

The screenshot shows the PHPCart Admin interface. At the top left is the PHPCart logo with the tagline 'The Simple E-Commerce Solution'. At the top right, a box indicates 'Current Version: 4.7' and 'Latest Version: 4.7'. Below the logo is a navigation bar with a dropdown menu labeled 'Admin Home' and links for 'Support Forums', 'Code Generator', and 'Documentation'. The main content area displays the message 'All fields must be filled out.' followed by three input fields: 'Set Admin Username:' with the value 'admin', 'Password:', and 'Repeat Password:'. A 'Save the settings' button is located below the fields. At the bottom, a footer contains the copyright information: 'Copyright ©2000 -2011- Carmosa. PHPCart.net'.

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 Cart Configuration

From the drop down box, select **Cart Configuration**. This screen will allow you to set the core values of phpCart. They include values associated with:

- **Company Configuration**
- **Email Configuration**



- **Tax Configuration**
- **Shipping Configuration**
- **Layout Configuration**
- **Misc Configuration**
- **Validation Configuration**

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

3.3.1 Company Configuration

The Company Configuration 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**

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.

Company Configuration	
Company Name:	<input type="text" value="My Company's Name"/>
Company Address:	<input type="text" value="123 Oak Street's"/> <input type="text" value="Suite 101A"/> <input type="text" value="My Town, CA 91234"/>
Company Website:	<input type="text" value="www.mycompany.com"/>
Company Phone:	<input type="text" value="123-456-7890"/>
Company Email: (Displayed on client emails, etc)	<input type="text" value="sales@yourdomain.com"/>
Company Admin Email: (Email to receive receipts, orders, etc)	<input type="text" value="sales@yourdomain.com"/>
Additional Order Emails: (Orders will be emailed to these also. Separate with a comma and space)	<input type="text"/>
Homepage URL:	<input type="text" value="/index.php"/>
Time Zone Offset:	<input type="text" value="Default (GMT -2 hours)"/>

3.3.2 Email Configuration

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.



You can also set a pre-calculated tax amount for each product as it is added to the shopping cart by adding the "tax" variable to the product link.

In order to tax a customer, that customer must be considered taxable. Taxability is determined based on the state/region and country the customer is in. You must edit the **States/Regions Tax Settings** and/or the Country Tax **Settings** by selecting them with the drop down box in the Admin Control Panel. You can add/modify states/regions and countries and then assign the two tax rates to each state/region or country.

If a product is assigned a taxrateid of 1 then any taxable customer will be charged the taxrate1 tax rate as assigned to that state/region or country.

Instead of assigning a taxrateid to each product you can set a flag in the configuration of phpCart that sets all products to be taxed if the customer is taxable. You can also set a flag in the configuration to determine if shipping should be taxed. If the customer is taxable and shipping is taxed, it then taxrate1 is used.

If you include the "tax" variable when adding the product to the cart the customer must still be taxable before the tax amount will be added to the cart. If the "tax" variable is included it will override any calculated taxable amount.

Tax Configuration	
Tax All Items: (If you want to charge tax for all taxable customers and all products)	No ▾
Tax Shipping: (If selected, only customers that are taxable will be taxed for shipping at the taxrate1 rate.)	No ▾
Default Tax Rate 1: (Prior to any state/region or country being entered)	<input type="text"/>
Default Tax Rate 2: (Prior to any state/region or country being entered)	<input type="text"/>



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.

By Order:

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.
- 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.



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.

Currency Character: You can display any currency character that you would like by inputting it into this field. We find it best to use the numerical currency code. Here are some examples:

US Dollar	$
British Pound	£
Japanese Yen	¥
Euro	€
Franc	₣

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.



Validation Configuration	
Text for First Name:	<input type="text" value="First Name"/>
Text for Last Name:	<input type="text" value="Last Name"/>
Text for eMail:	<input type="text" value="Email"/>
Text for Address:	<input type="text" value="Address"/>
Text for City:	<input type="text" value="City"/>
Text for State:	<input type="text" value="State"/>
Text for Zip Code:	<input type="text" value="Zip"/>
Text for Country:	<input type="text" value="Country"/>
Text for Phone:	<input type="text" value="Phone"/>

3.4 Gateway Configuration

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.



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



- **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.



6.1.1 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).
- quantity – the quantity of product to put in basket

6.1.2 Optional Product Variables

- 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 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 for more information
- taxrateid - Tax group of this item. See the Taxes section for more information.
- 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.
- option[] - 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
- option2 - Just 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



```
<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 Product Options

phpCart supports an unlimited number of options for each product. In addition, each option can have a price (adder) associated with it.

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. 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.



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

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

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

6.2.4 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.

Example:

```
<input type="text" name="option1[]">
```

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

6.3 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.



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 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%%.

7.2 Coupons

You can use coupons to provide sales incentives and specialized discounts to your customers.

7.2.1 Create Coupons

You can create coupons to:

1. Discount a dollar amount



2. Discount a percent of the total
3. Discount shipping (including providing for free shipping)
4. Limit discounts to order with a minimum value
5. Provide an expiration date for a coupon

7.2.2 Coupon Configuration

To use coupons, the following steps must be completed:

1. Coupons must be activated in the Admin Control Panel.
2. In your template, cart.php, you must have “COUPON” tags and the coupon form included in order for the coupon form to show up for your customers to enter their coupon code. When your customer enters a coupon code and presses the validate button, the cart will check to see if their coupon exists and will test to see that it meets the parameters that you set when you created the coupon.

It will look something like this:

```
<!-- BEGIN COUPON -->  
<form action="phpcart.php" method="post" name="couponform">  
<input type="hidden" name="action" value="coupon">Coupon  
<input type="text" name="couponcode">&nbsp; <br>  
<input type="image" src="%%TEMPLATEPATH%%images/validate.gif" border=0  
style="border-width: 0">  
</form>  
<!-- END COUPON -->
```

3. In your template, cart.php, you must have “DISCOUNT” tags included in order for the coupon value to show up in the totals section of the form.

It will look something like this:

```
<!-- BEGIN DISCOUNT -->  
<tr>  
  <td height="20" colspan="4" align="right" valign="middle">  
    <span class="username ">Coupon (%%COUPON%%)</span>
```




7.2.3 Adding a New Coupon:

To add a new coupon, follow these steps:

1. Insert a Coupon Code. It is best to use a code that is simple for your customers to remember and something that may relate to the purpose or function of the discount.
2. Choose the type of discount that you would like to provide.
3. Insert the amount of discount that you are offering.

To create a “Free Shipping” discount, just make this amount larger than your highest shipping charge. Your customer’s discount will only equal their actual shipping charge; thereby giving them free shipping.

4. Insert the minimum amount that the customer must purchase to qualify for the discount. If you want the discount to apply to every order without limitation, then just enter ‘0’ here.
5. Enter your expiration date; year, month and day.
6. The coupon will become active as soon as you press the “Save the Coupon” button.

7.2.4 Changing a Coupon:

Coupons cannot be changed at this time. It is best to delete one that you want to change and just recreate it.

7.2.5 Deleting a Coupon:

Press the delete link in the Coupon List that is associated with the coupon that you want to delete. It will be removed from the list permanently.



7.2.6 Customizing the Coupon Functionality:

The modules/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.

7.3 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, there is a file called shippingweight.inc.php which has some excellent code if you want to change your shipping options to ship by state or country or to change the ship by weight options.

7.4 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 determined in your cart.

7.5 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 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

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

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




- 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



9.3 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



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) 1998-2009, 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 

IonCube Loader Version

Zend Optimizer Version

PHP Credits

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

10.3 License 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:



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.