Vibracart Paypal™

V1.8

Vibracart Paypal Manual

Copyright 2010-2013 Vibralogix. All rights reserved.

This document is provided by Vibralogix for informational purposes only to licensed users of the Vibracart product and is provided on an ‘as is’ basis without any warranties expressed or implied.

Information in this document is subject to change without notice and does not represent a commitment on the part of Vibralogix. The software described in this document is provided under a license agreement. The software may be used only in accordance with the terms of that license agreement. It is against the law to copy or use the software except as specifically allowed in the license.

It is the users responsibility to ensure the suitability of Vibracart before using it. In no circumstances will Vibralogix be responsible for any loss or damage of data or programs as a result of using Vibracart. Your use of Vibracart implies acceptance of these terms.

Contents

Chapter 1 Introduction....4
What is Vibracart Paypal?....4
How does it work....5
Chapter 2 Installing Vibracart....7
Check your add to cart buttons are compatible....7
Check your page DOCTYPE....8
Installing Vibracart Paypal....8
Adding Vibracart to a page (popup cart mode)....8
Adding Vibracart to a page (embedded cart mode)....9
Adding Vibracart to a page (embedded mini cart mode)....10
Chapter 3 Discounts and coupons....12
Cart wide discounts settings....12
Item based discounts....14
Coupon Codes....17
Multi currency discounts....19
Chapter 4 Other Features....20
Display item images in the cart....20
Force agreement before Checkout....20
Limit quantity of each item....21
Limit number of items in the cart....21
Fade the page when the cart is displayed....22
Store and restore cart content for visitor....22
Animation when button clicked....22
Clearing the cart contents after an order....23
Linking cart items to product page....23
Chapter 5 Customization....25
Settings in settingsstyle.js....25
Settings in cart.php....32
CSS Styling....32
Chapter 6 Support....34

Chapter 1 Introduction

What is Vibracart Paypal?

As you probably know Paypal provides an easy to setup service for receiving payments for your products worldwide. They provide two main types of buttons that you can use on your site. The first is the buy now button which takes the user directly to Paypal to checkout. The second adds a product to Paypal's cart which normally opens in a new browser window. This cart works fine but doesn't usually keep the look and feel of the website using it. Also because it opens in a new browser window it can be confusing for the buyer. Vibracart Paypal has been designed to replace Paypal's cart with an elegant Ajax powered cart that fits neatly into your site. On most sites it is simply a drop in replacement using the existing Paypal add to cart buttons and only requiring a few lines to be added to the page to activate it.

- Simple to install into any page (doesn't need to be .php)
- Works with existing Paypal add to cart buttons if they are not Paypal hosted and not encrypted. If they are then the buttons can be converted very easily using Paypal's button wizard making them Vibracart Paypal compatible.
- Works as either a popup cart (not in a new window) or an embedded cart inserted in your page.
- Works seamlessly with Linklok Paypal for delivery of downloadable products.
- Supports all of Paypal's button features such as options, discounts and weight calculations.
- Works with all modern browsers including iPhone Safari. If a browser doesn't support the cart then it automatically switches back to Paypal's own cart.
- Cart works across any number of pages on your site.
- Total cost, number of items in cart and links to display the cart and checkout can be displayed anywhere on your page.
- Can require terms and conditions to be agreed before checkout
- Can display an image for each item
- Cart and items based discounts and coupon codes
- Supports Paypal sandbox

How does it work

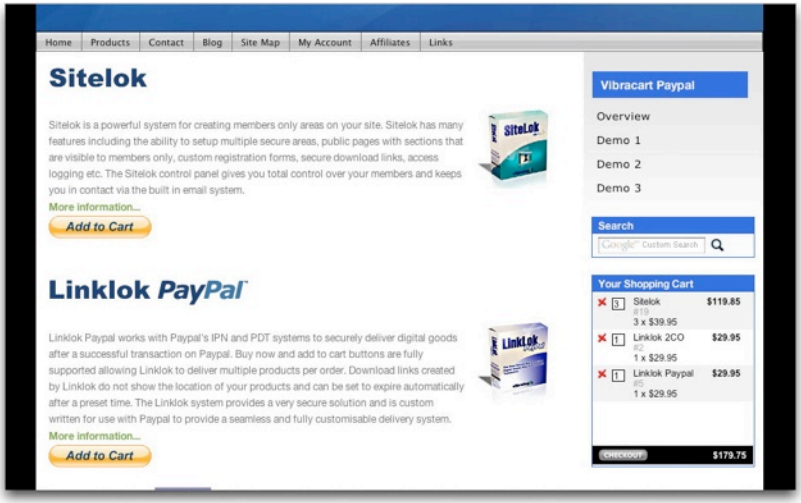
Vibracart Paypal can work as a popup cart that appears when items are added or as an embedded cart in your page that is always visible. In each mode you can adjust colors and sizes and various other settings. Here are some examples.

Popup cart (the default)

In this mode Vibracart displays a small floating widget showing the number of items currently in the cart, the total and links to display the cart or checkout. When an add to cart button is clicked the widget is updated and the main cart appears above the page (no new page access required). This can be closed as required. There are various options such as hiding the widget when the cart is empty and also keeping the main cart hidden until the user chooses to view it. Here is a screenshot of a typical installation showing the floating widget on the left and the popup cart displayed..

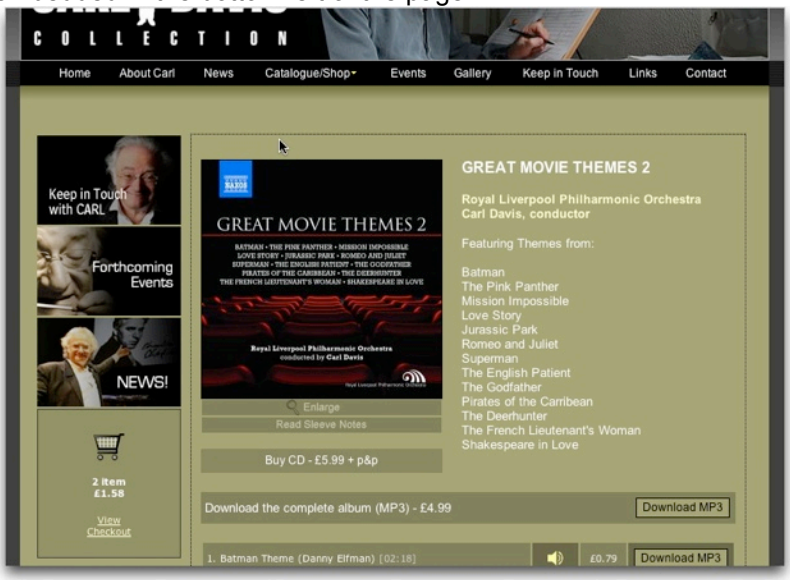
Embedded Cart

The embedded cart style allows you to insert the cart into your page layout instead of using the popup cart and widget. When add to cart buttons are clicked the cart automatically updates without any new page access being required. Here is an example with the cart embedded in the page on the right hand side.



Embedded Mini Cart

This is similar to the popup cart except that instead of using the widget you can embed the cart total, number of items and links to display the cart or checkout directly in your page. Here is an example with the mini cart embedded in the bottom left of the page.



Chapter 2 Installing Vibracart

In this chapter we will guide you through a simple default installation that will get you up and running as quickly as possible. Once Vibracart is working you can read further about tweaking things and changing the style etc.

Don’t worry if you know nothing about HTML or PHP scripts as this is not important. As long as you can use your page editor to add html to your pages where required then it is fine.

Check your add to cart buttons are compatible

Your site will need to use Paypal add to cart buttons that are not Paypal hosted buttons and not encrypted. A non hosted, non encrypted button normally looks something like this

```
<form target="paypal" action="https://www.paypal.com/cgi-bin/webscr" method="post">
<input type="hidden" name="cmd" value="_cart">
<input type="hidden" name="business" value="9N7WK6768PCCA">
<input type="hidden" name="lc" value="GB">
<input type="hidden" name="item_name" value="My Product">
<input type="hidden" name="item_number" value="id1">
<input type="hidden" name="amount" value="39.95">
<input type="hidden" name="currency_code" value="USD">
<input type="hidden" name="button_subtype" value="products">
<input type="hidden" name="no_note" value="1">
<input type="hidden" name="no_shipping" value="2">
<input type="hidden" name="add" value="1">
<input type="hidden" name="bn" value="PP-ShopCartBF:btn_cart_LG.gif:NonHosted">
<input  type="image"  src="https://www.paypal.com/en_GB/i/btn/btn_cart_LG.gif"  border="0"  name="submit"
alt="PayPal - The safer, easier way to pay online.">
<img alt="" border="0" src="https://www.paypal.com/en_GB/i/scr/pixel.gif" width="1" height="1">
</form>
```

If the button doesn't look similar then you will probably have to recreate the button html code using Paypal's button wizard. To do this make the button as normal but in step 2 make sure the 'Save button at Paypal' check box is **not** checked and in the final step click the 'Remove code protection'.

There is no security risk creating buttons like this. It is true that the price could be tampered with when the button is not encrypted but this is only a problem when selling downloadable products. If so we recommend using Linklok Paypal to securely automate delivery of the downloads and to ensure the amount cannot be changed.

Check your page DOCTYPE

Vibracart doesn't normally care which DOCTYPE is used but I&& and IE8 do require one of the following DOCTYPEs to support the fixed floating popup cart. This is only required if you use the popup cart or the widget. If you use the embedded cart then you can ignore this step. They seem to have fixed the issue in IE9.

For html pages use

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

For xhtml pages use

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Installing Vibracart Paypal

To install Vibracart Paypal on your server just upload the vibracart folder from the zip file to the root of your site. So for example it should be at

http://www.yoursite.com/vibracart/

Now that Vibracart is installed on your server you need to choose one of the cart modes to use, popup, embedded cart or embedded mini cart. You can only use one mode per page but you can mix different modes on different pages of your site. We recommend trying the popup mode first of all as this is the easiest so will get you up and running quicker.

Adding Vibracart to a page (popup cart mode)

Find a page on your site that contains one or more compatible Paypal add to cart buttons (not Paypal hosted or encrypted). Open it in your editor and follow these simple steps.

1) In the HEAD section of your page add these lines

```
<link href="/vibracart/vibracartstyle1.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="/vibracart/settingsstyle1.js"></script>
<script type="text/javascript" src="/vibracart/sarissa.js"></script>
<script type="text/javascript" src="/vibracart/vibracart.js"></script>
```

2) Scroll down to the bottom of your page and insert these lines just before the </body> tag

```
<script language="javascript" type="text/javascript">
startcart()
</script>
```

Now republish your page and Vibracart Paypal should be working and appear when you click the add to cart button.

Adding Vibracart to a page (embedded cart mode)

Find a page on your site that contains one or more compatible Paypal add to cart buttons (not Paypal hosted or encrypted). Open it in your editor and follow these simple steps.

1) In the HEAD section of your page add these lines

```
<link href="/vibracart/vibracartstyle2.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="/vibracart/settingsstyle2.js"></script>
<script type="text/javascript" src="/vibracart/sarissa.js"></script>
<script type="text/javascript" src="/vibracart/vibracart.js"></script>
```

2) Scroll down to the bottom of your page and insert these lines just before the </body> tag

```
<script language="javascript" type="text/javascript">
startcart()
</script>
```

3) At the point in your page where you want the cart to be displayed on your page add the following lines. This could be inside a table cell or a div on your page for example.

```
<script language="javascript" type="text/javascript">
insertcart()
</script>
```

Now republish your page and Vibracart Paypal should be displaying on your page which updates when you click an add to cart button.

Most likely the cart won't be the correct size so you can adjust the width and height in settingsstyle2.js (inside the vibracart folder) using the following settings (in pixels).

```
var cart_height=200
var cart_width=188
```

Adding Vibracart to a page (embedded mini cart mode)

Find a page on your site that contains one or more compatible Paypal add to cart buttons (not Paypal hosted or encrypted). Open it in your editor and follow these simple steps.

1) In the HEAD section of your page add these lines

```
<link href="/vibracart/vibracartstyle3.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="/vibracart/settingsstyle3.js"></script>
<script type="text/javascript" src="/vibracart/sarissa.js"></script>
<script type="text/javascript" src="/vibracart/vibracart.js"></script>
```

2) Scroll down to the bottom of your page and insert these lines just before the </body> tag

```
<script language="javascript" type="text/javascript">
startcart()
</script>
```

3) Now in your page you can add the following items related to the cart.

Cart total

To display the cart total in your page add this html.

```
<span id="showcarttotal">$0.00</span>
```

You can change the $ symbol as required.

If you need to display in a second place on the page use showcarttotal2.

Number of items in cart

To display the number of items in the cart in your page add this html.

```
<span id="showitemcount">0</span>
```

If you need to display in a second place on the page use showitemcount 2.

Show cart link

To display a link to show the cart add the following to your page.

```
<a href="" onclick="showCart(); return(false)">Show Cart</a>
```

Checkout Link

To display a checkout link add the following to your page.

```
<a href="" onclick="checkOut(); return(false)">Checkout</a>
```

Now republish your page and you should see the total and number of items update as you click add to cart buttons.

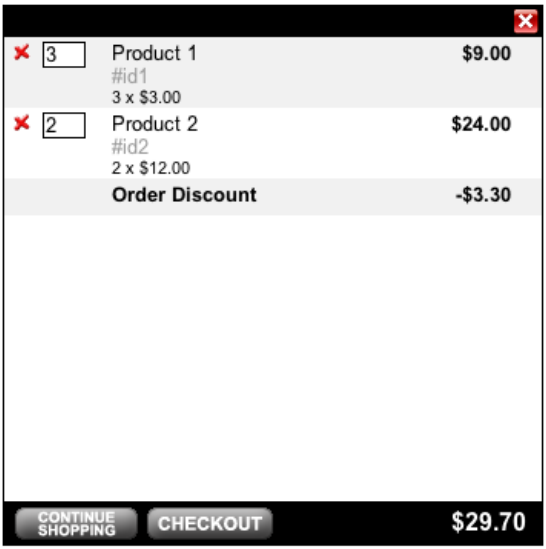
Chapter 3 Discounts and coupons

Vibracart has a powerful discount system allowing you to apply cart or item based discounts based on quantities or total amount ordered etc. Each discount specified can be set so that it is only applied if a matching coupon code has been entered. Each discount is defined on a line in discount.php (in the vibracart folder). You can define any number of discounts. Here is an example with three discounts defined offering 10% on orders of $20+, 15% for $40+ and 20% for $60+. The last 20% discount is only applied if the coupon code PROMO1 has been entered (see later for details)

```
<?php
$ordertotaldiscountpercent[]="10,20";
$ordertotaldiscountpercent[]="15,40";
$ordertotaldiscountpercent[]="20,60,PROMO1";
?>
```

Cart wide discounts settings

Cart wide discounts are applied to the cart total when the specified total cost or quantities are purchased. The total discount amount for cart wide discounts is displayed as an entry in the cart as follows



and is passed to Paypal as a discount amount. Where multiple discount settings are defined Vibracart will apply the one that has the largest discount applicable.

$ordertotaldiscountpercent

Applies a percentage discount to the cart total if the order total is the specified value or higher. For example

$ordertotaldiscountpercent[]="10,30.00";

Would apply a 10% discount to the cart total if the order total is $30 or more.

$ordertotaldiscountamount

Applies a discount amount to the cart total if the order total is the specified value or higher. For example

$ordertotaldiscountamount[]="5.00,30.00";

Would apply a $5.00 discount to the cart total if the order total is $30 or more.

$orderqtydiscountpercent

Applies a percentage discount to the cart total if the total number of items purchased is the specified value or higher. For example

$orderqtydiscountpercent[]="10,5";

Would apply a 10% discount to the cart total if the order contains 5 or more items. This would apply if 1 of five different products were ordered or if 3 of one product and 2 of another were ordered.

$orderqtydiscountamount

Applies a discount amount to the cart total if the total number of items purchased is the specified value or higher. For example

$orderqtydiscountamount[]="5.00,5";

Would apply a $5.00 discount to the cart total if the order contains 5 or more items. This would apply if 1 of five different products were ordered or if 3 of one product and 2 of another were ordered.

Item based discounts

Item based discounts are used to adjust an items unit price based on the quantity purchased of that item or a group of items. Where multiple discounts apply to an item Vibracart will apply the largest discount for that item so that they are not all applied at the same time.

$itemqtydiscountpercent

Applies a percentage discount to the specified product item price if the quantity purchased of that product is the same or more than specified. For example

```
$itemqtydiscountpercent[]="id1,10,5";
```

Would apply a 10% discount to the item price of product id1 when 5 or more are purchased. You can list multiple products (separated by colons) if you wish to apply the same rule to each. For example

```
$itemqtydiscountpercent[]="id1:id2,10,5";
```

Would apply a 10% discount to product id1 if 5 or more are purchased, and a 10% discount to product id2 if 5 or more are purchased. Saves having to have a line for each item with the same discount structure.

$itemqtydiscountamount

Applies a discount amount to the specified product item price if the quantity purchased of that product is the same or more than specified. For example

```
$itemqtydiscountamount[]="id1,0.25,5";
```

Would apply a $0.25 discount to the item price of product id1 when 5 or more are purchased. You can list multiple products (separated by colons) if you wish to apply the same rule to each. For example

```
$itemqtydiscountamount[]="id1:id2,0.25,5";
```

Would apply a 0.25 discount to product id1 if 5 or more are purchased, and a 0.25 discount to product id2 if 5 or more are purchased. Saves having to have a line for each item with the same discount structure.

$multiqtydiscountpercent

Applies a percentage discount to the unit price of a group of items where more than a certain number from the group are purchased. For example if you have a selection of mp3 tracks you can offer 10% off of those tracks if any 3 from the selection are purchased.

$multiqtydiscountpercent[]="id1:id2:id3:id4:id5,10,3";

Would apply a 10 percent discount to any of id1, id2, id3, id4 or id5 if 3 or more are purchased. This would include buying three id1 or one id1 and two id4 for example.

$multiqtydiscountamount

Applies a discount amount to the unit price of a group of items where more than a certain number from the group are purchased. For example if you have a selection of mp3 tracks you can offer $0.25 off of each track if any 3 from the selection are purchased.

$multiqtydiscountamount[]="id1:id2:id3:id4:id5,0.25,3";

Would apply a 0.25 discount to any of id1, id2, id3, id4 or id5 if 3 or more are purchased. This would include buying three id1 or one id1 and two id4 for example.

$multitotaldiscountpercent

Applies a percentage discount to the unit price of a group of items where more than a total sales value from the group is purchased. For example if you have a selection of mp3 tracks you can offer 10% off of those tracks if the user spends $10 on them. For example

$multitotaldiscountpercent[]="id1:id2:id3:id4:id5,10,10.00";

Would apply a 10 percent discount to any of id1, id2, id3, id4 or id5 if $10.00 or more is spent on those items.

$multitotaldiscountamount

Applies a discount amount to the unit price of a group of items where more than a total sales value from the group is purchased. For example if you have a selection of mp3 tracks you can offer $0.25 off of each track if the user spends $10 on them. For example

$multitotaldiscountamount[]="id1:id2:id3:id4:id5,0.25,10.00";

Would apply a $0.25 discount any of id1, id2, id3, id4 or id5 if $10.00 or more is spent on those items.

$combiqtydiscountpercent

Applies a percentage discount to the unit price of a group if items where more than a certain number of each item listed are purchased. For example if you have a selection of mp3 tracks you can offer a 10% discount off of those tracks if all 7 from an album are purchased. For example

$combiqtydiscountpercent[]="id1:id2:id3,10,1";

Would apply a 10 percent discount on each item if at least one id1 **and** one id2 **and** one id3 are purchased.

$combiqtydiscountamount

Applies a discount amount to the unit price of a group if items where more than a certain number of each item listed are purchased. For example if you have a selection of mp3 tracks you can offer a $0.25 discount off of those tracks if all 7 from an album are purchased. For example

$combiqtydiscountamount[]="id1:id2:id3,0.25,1";

Would apply a 0.25 discount on each item if at least one id1 **and** one id2 **and** one id3 are purchased.

$bogodiscountpercent

This can be used to offer buy one get one free or buy two get one half price type discounts. For example

$bogodiscountpercent[]="id1:id2:id3,1,100";

sets a buy one get one free (100% discount) when any combination of id1, id2 or id3 are purchased. The 1 setting defines the number to be purchased to get the discount (buy **one** get ...) and the 100 defines the percent discount. The discounts are applied to the cheapest items first.

Note that this function also works with multiples of the same item. In that case we adjust the unit price so that the item total is correct. You may find that a penny can be lost in rounding when adjusting the unit price in that case.

$bogodiscountamount

This can be used to offer buy one get one free or buy two get one cheaper type discounts. For example

```
$bogodiscountamount[]="id1:id2:id3,1,0.50";
```

sets a buy one get one 0.50 off when any combination of id1, id2 or id3 are purchased. The 1 setting defines the number to be purchased to get the discount (buy **one** get ...) and the 0.50 defines the discount amount. The discounts are applied to the cheapest items first.

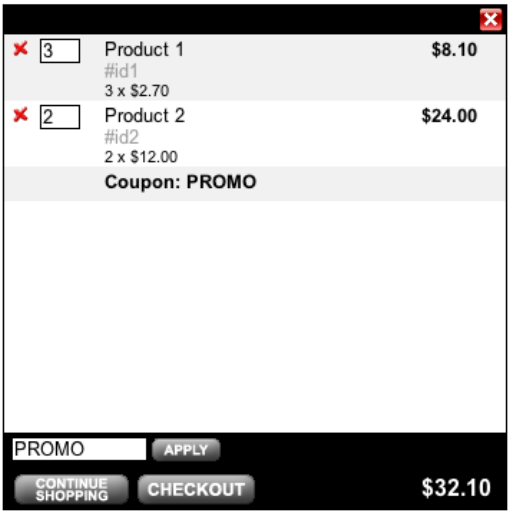
Note that this function also works with multiples of the same item. In that case we adjust the unit price so that the item total is correct. You may find that a penny can be lost in rounding when adjusting the unit price in that case.

Coupon Codes

Normally each discount defined will be applied to all qualifying orders. It is also possible to define some or all of the discounts so that they are applied only if a matching coupon code is entered by the buyer as well as meeting the conditions of the discount. Only one coupon code can be entered by the user per order and the coupon code is not case sensitive.

Enabling the coupon code system

Once the coupon code system is enabled a coupon field and apply button is added to the cart.



The coupon part of the cart is defined in the following settings in settingsstyle1.js, settingsstyle2.js or settingsstyle3.js depending on which style you are using.

```
// Coupon code
var showCoupon=true
var cart_height_coupon=25
var couponinputtext="Enter Coupon"
var coupontext="Coupon: "
var msg_couponnotvalid="Coupon not valid"
```

showCoupon	Set to true or false to show or hide the coupon field on the cart.
couponinputtext	Defines the text shown in the field itself.
coupontext	Defines the text displayed next to the coupon code in the cart.
msg_couponnotvalid	Defines the message displayed when an invalid coupon code is entered.

Requiring a coupon code for a discount

You can set any discounts you have defined so that they are only applied when a valid coupon code is entered. To do this just add the coupon code to the end of the discount like this

```
$itemqtydiscountpercent[]="id1,10,5,PROMO";
```

This would apply the item discount only if PROMO is entered as the coupon code.

You can add multiple coupon codes that are all valid by separating them with colons as follows

```
$itemqtydiscountpercent[]="id1,10,5,PROMO:GOLD123:GETITNOW";
```

Multi currency discounts

If you have buttons in different currencies on your site you may want to define discounts and coupons differently for each one. By default Vibracart will use the file called discounts.php to check for discounts. However if it first finds a file called discountsUSD.php (if the cart currency is USD) this will be used instead. So for example if you have buttons in USD and EUR you can make two discount files called discountUSD.php and discountsEUR.php.

Remember the currency used by the cart is defined by the currency of the first item added to the cart for that session.Paypal does not allow items with prices in different currencies to be added to the same order.

Chapter 4 Other Features

Display item images in the cart

Although Paypal doesn't support images in their buttons you can have Vibracart display a small image next to the item in the cart which will appear like this

To do this add the following line to your buttons

```
<input type="hidden" name="vibracart_image" value="productimage.png">
```

and enable cart images by setting var showItemImage=true

The image file can be .png, .gif or .jpg and the size is not too important as Vibracart will adjust this as required.

Force agreement before Checkout

If you require a buyer to agree to your terms and conditions before checkout then set the following settings in settingsstyle1.js, settingsstyle2.js or settingsstyle3.js depending on which style you are using.

```
var showTerms=true
var cart_height_terms=20
var termspage="terms.html"
var termstarget="_blank"
var termstext="I agree to the terms and conditions"
var msg_terms="You must agree to the terms before checkout"
```

This will add a new section to the cart looking like this

You can change the text of the link by setting termstext and the page pointed to by setting termspage. If the user tries to checkout without agreeing to the terms then they will see the error set in msg_terms.

Limit quantity of each item

On some sites you may want to limit how many of each item can be purchased. To do this just set

```
var itemquantitylimit=1
var msg_itemquantity="No more of this item can be purchased"
```

as required. Set itemquantitylimit=0 to allow unlimited items.

If you set itemquantitylimit=1 (often the case when selling MP3 tracks for example) you may also like to remove the quantity field and the update button from the cart using these settings.

```
var showItemQuantity=false
var updateButton=""
```

The above will apply to all products. You can if you wish set this specifically for each productinstead by adding this line to the button to set the maximum number that can be ordered.

```
<input type="hidden" name="vibracart_maxquantity" value="5">
```

Limit number of items in the cart

If you want to limit how many items can be purchased (added to the cart) then set the following

```
var cartquantitylimit=10
var msg_cartquantity="Only 10 items can be purchased"
```

as required. Set cartquantitylimit=0 to allow unlimited items.

Fade the page when the cart is displayed

If you wish to fade (gray out) the page when the popup cart is displayed set the following in settingsstyle1.js (depending on the style you use).

var pageFade=true;

Store and restore cart content for visitor

If you wish for a users cart contents to be stored between visits using a cookie just need to enable that feature by setting the following in settingsstyle1.js (depending on the style you use).

var savecart=true

This will attempt to store the cart contents in a cookie which can be read on their next visit. There is a limit on the number of items that can be stored in a cookie so Vibracart will store as many as it can.

Animation when button clicked

You can enable a simple animation which moves between the Paypal button and the cart in the embedded mode or the minicart in the minicart mode when the button is clicked. The popup cart mode is not supported as the cart will appear where the user is looking anyway.

To enable this just set the following in settingsstyle2.js (depending on the style you use).

var animateitem=true

You can change the image used by replacing flyingitem.png and flyingitem.gif (required by IE6) in the vibracart folder.

You can adjust the speed of the animation by adjusting the values in

var animateitemspeed=5
var animateitemstep=25

The number set in animateitemspeed is the milliseconds between each move and animateitemstep sets the size of each step in pixels.

Clearing the cart contents after an order

By default the cart contents are not cleared after an order as there is actually no easy place to do that . We can't clear the cart when the checkout button is clicked and the user is sent to Paypal as they may in fact change their mind and return to add more item to the cart.

The best solution is to clear the cart after the user has paid and been returned to your sites thank you page. There are two ways to do this.

1) If you use Linklok Paypal then you can add this setting to linklokipn.php near your other settings to clear the cart automatically after payment.

   ```
   $ClearVibracart=true;
   ```

2) If you don't use Linklok Paypal then you can add the following html to your thank you page. Anywhere will do as it will just display a 1 pixel transparent image.

   ```
   <img src="/vibracart/clearcart.php?mode=all" width="1" height="1"/>
   ```

   The above will clear the cart contents and any stored cart (if you enabled that feature).

   You can instead use mode=cart to clear just the cart or mode=stored to clear just the stored cart (the cookie) but leave the current sessions cart intact.

Linking cart items to product page

On some sites it may be useful to have links inside the cart which a user can click to return to the product page (where the add to cart button was). When this feature is enabled the product id in the cart will become a hyperlink for this purpose. Vibracart will automatically store the page where the product was added to the cart enabling the user to easily return to the it. To use this feature just set the following in settingsstyle1.js (depending on the style you use).

```
var autoProductPageLink=true
var productPageTarget=""
```

If you prefer to manually force a link back page instead of automatically using the page containing the button then you can add this line to the Paypal button to set the page you want.

```
<input type="hidden" name="vibracart_pagelink" value="http://www.yoursite.com/page.html">
```

If you with to have the page open in a new browser window then you can change the productPageTarget setting to

```
var productPageTarget="_blank"
```

Chapter 5 Customization

In most cases the cart should be fine using the default settings. However for those that like to tweak things here are some further details.

Settings in settingsstyle.js

Most of the settings for the cart are in the files settingsstyle1.js, settingsstyle2.js or settingsstyle3.js. Actually each of these three files are the same but have some predefined settings most suitable for the three examples described earlier. You can in fact use any one of the settings files and adjust the settings as needed. The default settings in the three files are suitable for use as follow:-

settingsstyle1.js Popup cart with widget
settingsstyle2.js Embedded cart with no widget
settingsstyle3.js Popup cart with minicart and no widget

var cart_url="/vibracart/"

This is the folder where Vibracart is installed. By default this is in a folder called vibracart in the root of your site. If you need to move it then adjust this setting and also in the calls to Vibracart in your page.

var showcartatstart=false

This setting determines if the popup cart is initially displayed on the page on entry. Set to true to display immediately on entry or false to display only when an add to cart button is clicked.

var showcartonadd=true

Set to true to display the popup cart each time an item is added. Set to false to have the cart displayed only when the user clicks a link to display it, either in the widget or on your page.

var hidecartifempty=false

Set to true to hide the cart when it is empty. This is probably more useful when using the embedded cart.

var cartposition="fixed"

This sets the CSS position method of the cart. When using a popup cart this must be set to fixed. When using an embedded cart this must be set to static.

var cart_left=-1

This sets the horizontal position of the popup cart (pixels) relative to the left. You can use -1 to centre the cart and negative values to position relative to the right side. For example -20 would position on the right leaving room for a scroll bar.

var cart_top=-1

This sets the vertical position of the popup cart (pixels) relative to the top. You can use -1 to centre the cart and negative values to position relative to the bottom. For example -20 would position at the bottom leaving room for a scroll bar.

var cart_height=350
var cart_width=350

These set the height and width in pixels of the cart. For the embedded cart you will need to adjust those to match your site layout.

var cart_height_header=20
var cart_height_footer=28

These set the height in pixels of the header section and footer section of the popup cart. For an embedded cart you would normally set the header height to 0 as it is not needed.

var animate_style="none"
var animate_pixels=20
var animate_speed=3

These settings should be left as is as they are not user adjustable currently.

var showItemRemove=true
var showItemQuantity=true
var showItemDescription=true
var showItemTotal=true

These settings determine whether the item remove icon, item quantity field, item description and item total fields are displayed. Set to true to show the item and false to hide it.

var optionValueSeparator=":"
var optionSeparator=""

The characters used to separate options and option values in the cart.

var idPrefix="#"
var idSuffix=""

The characters used as a prefix and suffix on the item id (item_number) in the cart.

var discountPriceSeparator=", "
var discountQuantityOperator=" x "

The characters used when displaying discounts in the cart.

var msg_cartempty="The cart is empty"
var msg_button="Invalid button"
var msg_currency="You cannot mix items in different currencies"
var msg_items="items"

The text used to display warning messages in the cart. You can change these as required. Useful when you need to translate the cart.

var emptyCartImage="emptycart.png"
var deleteButton="delete.png"
var closeButton="closecart.png"
var continueButton="continue.png"
var updateButton=""
var checkoutButton="checkout.png"
var busyImage="ajax-loader-transparent.gif"

These settings define the images used for the buttons etc on the popup cart. Normally these are fine but if you use an embedded cart you may prefer to use some smaller buttons. We have some smaller examples in the vibracart folder that are used in settingsstyle2.js. Also when using the embedded cart you would normally not have a close cart image or continue shopping button so set those to

var closeButton=""
var continueButton=""

The update button was used on earlier version of Vibracart but is not really needed now as the prices update automatically when a user changes the quantity.

Notice that we normally use .png images (except for the busy image) which provide better transparency support. However IE6 doesn't support this so Vibracart will switch to .gif versions of the images when it detects IE6. For this reason if you use any custom images you should create both .png and .gif versions. You can enable or disable this IE6 .gif support using the setting var IE6usegif=true

var use_widget=true

Set to true to use the widget and false to disable it. Normally when using an embedded cart you would set this to false.

var initiallyshowemptywidget=true

Determines if the widget is initially displayed on the page when it is empty. Set to true or false.

var hidewidgetifempty=false

Set to true to hide the widget if it is empty. Set to false to always display the widget.

var widget_left=0

This sets the horizontal position of the widget (pixels) relative to the left. You can use -1 to centre the widget and negative values to position relative to the right side. For example -20 would position on the right leaving room for a scroll bar.

var widget_top=-1

This sets the vertical position of the widget (pixels) relative to the top. You can use -1 to centre the widget and negative values to position relative to the bottom. For example -20 would position at the bottom leaving room for a scroll bar.

var widget_height=120
var widget_width=50

Set the height and width of the widget.

var widgetbackground="widgetgradient.png"
var widgetimage="widgetcart.png"

The images used as the background and image in the widget.

Notice that we normally use .png images (except for the busy image) which provide better transparency support. However IE6 doesn't support this so Vibracart will switch to .gif versions of the images when it detects IE6. For this reason if you use any custom images you should create both .png and .gif versions. You can enable or disable this IE6 .gif support using the setting var IE6usegif=true

var IE6usegif=true

Set to false if you don't want Vibracart to switch to .gif versions of images when it detects IE6. See other image settings for further details.

var sandbox=""

If you want to use a Paypal sandbox button you need to tell Vibracart by setting the sandbox account in this setting. So for example you could set var sandbox="paypal@yoursite.com"

You can leave your live Paypal buttons on your page as they are and use this setting to instantly switch all of the buttons to be sandbox enabled. This makes testing easy as you don't have to update your buttons. To disable the sandbox just set back to var sandbox=""

var messagedelay=2000
var errordelay=2000

Sets the time in milliseconds that messages or errors are displayed in the cart.

var usejavascriptalert=false

If set to true messages and errors will be displayed using a javascript alert. Set to false to display messages inside the cart.

var showTerms=false
var cart_height_terms=20
var termspage="terms.html"
var termstarget=""
var termstext="I agree to the terms and conditions"
var msg_terms="You must agree to the terms before checkout"

These settings allow you to display a checkbox and link forcing buyers to agree to your terms and conditions before checkout. showTerms should be set to true to enable this feature. You can set the url and target window for the link, the link text and the error message text.

var itemquantitylimit=0
var msg_itemquantity="You cannot order more of this item"

These settings allow you to limit the maximum quantity of each item that can be purchased. Set to 0 to allow unlimited items. So for example if you set var itemquantitylimit=1 then only one of each item can be added to the cart. You can set the error message displayed if the quantity is exceeded or set msg_cartquantity="" to not display any message.

You can if you wish remove the quantity field and the update cart button (if displayed) in this case by setting the following.

var showItemQuantity=false
var updateButton=""

var cartquantitylimit=0
var msg_cartquantity="Only 3 items can be purchased"

These settings allow you to limit the maximum quantity of each item that can be purchased. Set to 0 to allow unlimited items. So for example if you set var cartquantitylimit=10 then only the total items allowed in the cart would be 10. You can set the error message displayed if the quantity is exceeded. If you wish you can set msg_cartquantity="" to not display any message.

var showItemImage=false

Although Paypal doesn't support images in their buttons you can have Vibracart display a small image next to the item in the cart. To do this add the following line to your buttons

<input type="hidden" name="vibracart_image" value="productimage.png">

and enable cart images by setting var showItemImage=true

The image file can be .png, .gif or .jpg and the size is not too important as Vibracart will adjust this as required.

var msg_checkout="Please wait while we connect to Paypal"

This is the message that is displayed when the user clicks checkout. It is just a reminder that the connection to Paypal has started (Paypal is sometimes slow to load). You can set var msg_checkout="" if you don't want this message.

var discounttext="Order Discount"

Sets the text displayed in the cart for the discount.

var showCoupon=true
var cart_height_coupon=25
var couponinputtext="Enter Coupon"
var coupontext="Coupon: "
var msg_couponnotvalid="Coupon not valid"

Define the coupon settings for the cart. You should set

var showCoupon=true

to enable the coupon code field on the cart.

var pageFade=true

Setting to true will fade the page behind the cart when it appears. Set to false to disable.

var savecart=true

Setting to true will enable storing of a users cart in a cookie for later use.

var animateitem=true
var animateitemimage="flyingitem.png"
var animateitemspeed=5
var animateitemstep=25

Enables the animation when a Paypal button is clicked. See earlier in the manual for details.

var escapeclosescart=true;

Allows the escape key to close a popup cart if set to true.

var autoProductPageLink=false
var productPageTarget=""

Enables the product page return link and target window if required.

Settings in cart.php

There are a few setting in cart.php that you can adjust, These relate to currency symbols and decimal places etc. Normally they don't need adjusting but read on if you need to.

Near the top of cart.php you will see settings for each of the currencies supported by Paypal. For example here are the settings for USD.

```
$currencyDetails['USD']['prefix']="$";
$currencyDetails['USD']['suffix']="";
$currencyDetails['USD']['code']="USD";
$currencyDetails['USD']['decimals']=2;
$currencyDetails['USD']['fraction']=".";
$currencyDetails['USD']['thousands']="";
```

Here you can set the currency symbol prefix (displayed before the value), the suffix (displayed after the value), the number of decimal places, the character to use for a decimal point (fraction) and also the character to use for a thousands separator. Don't change the currency code though as this must match Paypal.

CSS Styling

The CSS styling for the cart and widget are in the files vibracartstyle1.css, vibracartstyle2.css or vibracartstyle3.css. Actually each of these three files are the same but have some predefined styles most suitable for the three examples described earlier. You can in fact use any one of the CSS files and adjust the settings as needed. The default settings in the three files are suitable for use as follow:-

vibracartstyle1.css Popup cart with widget
vibracartstyle2.css Embedded cart with no widget
vibracartstyle3.css Popup cart with no widget

CSS styling is a complex area so it is best to experiment with the settings to get the look you wish. We recommend only modifying styles that affect color or size as others may affect the operation of the cart.

Chapter 6 Support

Hopefully if you have followed this manual carefully everything will be working fine. However sometimes things don't go quite so smoothly so if you have any questions or problems then please check the FAQ on the support page or contact us.

Support and updates for Vibracart Paypal are free of charge for one year after purchase. Shortly after purchase you should receive a username and password that provide you with access to the latest downloads, FAQ and other resources.

Support area: http://www.vibralogix.com/support/