



HTML Integration Guide

Version 0.93
Release: 29/05/2009

Copyright © RMP Protection Limited 2008-2009

Contents

Introduction.....	3
What you need.....	3
How it works.....	3
Preventing fraud.....	4
Step by Step guide.....	5
Stage 1 - Registering for a CarrotPay merchant account.....	5
Stage 2 - Creating Protected Links and secure file names.....	6
Stage 3 - Creating the button.....	9
Improving the look.....	11

Introduction

This document gives a technical guide to integrating your Website with the CarrotPay payment system using simple HTML buttons. If you have a dynamically driven Website using PHP, ASP etc. we recommend you use the methods outlined in the “CarrotPay: Script Integration Guide”.

You should also read the “CarrotPay: Overview for Merchants” document so you are familiar with the general concepts of CarrotPay payment before reading this guide.

What you need

We're assuming you have, or are building, a Website where you want to sell content for immediate on-line delivery to your customers. This method of integration isn't designed for use when you are delivering physical goods – for that, please read (or get your Web developer to read) the “CarrotPay: Script Integration Guide”.

We will assume you have the ability to edit basic HTML pages on your site, and to change the name of your content files. Beyond that, you don't need any programming skills or special hosting packages to start to accept payments through CarrotPay.

To collect your payments, you'll need to have a CarrotPurse installed, as will your customers who want to pay with Carrot-WebCoins. You can download a CarrotPurse for free from www.carrot.org.

How it works

As described more fully in the “CarrotPay: Overview for Merchants” document, the basic process for setting up for CarrotPay payments is as follows:

1. You register for a CarrotPay merchant account. This is quick and anonymous, and gives you three things:
 - i. A CarrotPay merchant ID, which looks like this: “BJGC-QZPM-YQBC-GJQP”
 - ii. A 'hash seed', which looks like this: “mjqbpzcyjwmbpzcb”
 - iii. A 'secret key', which is never shown to you

All three of these are kept securely in a special wallet in your CarrotPurse. You only need the first two for CarrotPay integration; the secret key is only used to collect your received Carrot-WebCoins from CarrotPay.

2. For each file you want to sell, you get the CarrotPay 'Button Wizard' to give you a special 'secure' name for it, and rename the file to that name. These names are designed so people can't guess them and get the content without paying.
3. For each file, you then create a CarrotPay button in HTML. You can either write this yourself or go through the 'Button Wizard' which will create one for you. The Buy button (more correctly an HTML FORM), contains details of the item you are selling, including a description (in UTF-8 format), price, and a modified version of the original filename know as a 'Protected Link' (see later), but – not the 'secure' name mentioned above.
4. Then you upload all your files (under their secure names), and your new Web pages with CarrotPay buttons on them, and you're open for business.

When a customer visits your site and wants to buy something, the following happens:

1. They click on the CarrotPay button for the item they want to buy.
2. The button points to the CarrotPay Website, but it doesn't open up a new Web page; rather it generates special Javascript commands which send a request for payment to the customer's CarrotPurse.
3. The customer's CarrotPurse asks them to authorise the payment, or, if the requested amount is less than their automatic “zero-click” payment limit, will just notify them that a payment has been made.
4. The customer's CarrotPurse then sends the correct value of Carrot-WebCoins to CarrotPay, where they are checked and are stored ready for you to retrieve later.
5. CarrotPay recreates the 'secure' URL using the 'Protected Link' in the payment form which then points to the secure name of the file, and the user's browser fetches the content from there.

Some time after one or more of your products have been purchased, you may use your CarrotPurse merchant wallet to 'Collect' the received Carrot-WebCoins so you have them available locally, either to spend on other on-line services yourself, or to transfer to a Bank Account or PayPal etc. CarrotPay applies a small transaction charge (which is what pays to keep the CarrotPay service running) as you collect your Carrot-WebCoins.

Preventing fraud

The process of renaming your content files to 'secure' names means that people can't just look at your HTML FORM to find out the URL of your content and fetch it for free. Because they don't know the secret 'seed' which generates the special secure name of the file they have to go through the CarrotPay process which generates the secure URL for them. Also, the way the secure name is worked out includes the price, so they can't change the price on the form and get the content for less than they should.

The only problem with this very simple HTML only system is that a secure URL will work for anyone who gets hold of it. Hence, if after paying for content the first time, that person posts the secure URL to a forum or sends it to their friends, the recipients could immediately fetch the content without going through the payment process a second time. There are ways to prevent this, but they are more involved and usually require server-side scripting, as described in the "CarrotPay: Script Integration Guide". CarrotPay also provides one client side solution called 'Protected Content', which is described in "CarrotPay: Protected Content Integration Guide". For the rest of this guide we focus on the simple HTML solution.

Here are a few things you can do to reduce fraud:

1. Change the filenames frequently. This involves changing the original name of the file in the payment form (perhaps by adding a number or date on the end), using CarrotPay's 'Button wizard' to regenerate the secure name and renaming the file. How often you do this is up to you, but obviously it is only really worth doing for small numbers of relatively high-value content.
2. Make sure no-one can get the directory listing of the directory with the files in, since that will reveal all the secure filenames! You can usually prevent this by making sure there is an "index.html" (or similar) file in the directory.
3. Block search engines from indexing your content files by including a "robots.txt" file in the directory – see your Web host's instructions for details.

This said, we don't think you should be too obsessive about this – after all, bear in mind that unless you are using Digital Rights Management, anyone could buy the content once, then copy and repost the files themselves in any case. This type of system works best for fairly small values where it is just easier for someone to pay a small amount to get the content legally from you rather than search around trying to find an illegal copy from someone else.

Note: It is a condition of use of CarrotPay that it is not used for any illegal purpose, including distribution of copyrighted works (e.g. music) or illegal material.

Step by Step guide

There are three stages to using CarrotPay on your website:

1. Register for a CarrotPay merchant account
2. Rename your files to special secure filenames
3. Create your "Pay with Carrot-WebCoins" buttons

Stage 1 - Registering for a CarrotPay merchant account

Anyone can use a CarrotPurse to buy things using Carrot-WebCoins, but to sell things you have to have a CarrotPay merchant account. This isn't nearly as complex as a Credit Card merchant account! Really all it is is a store for your Carrots until you collect them, and a shared 'seed' which allows the secure filename modification 'trick' to work. It's free, completely anonymous if you choose and you don't have to provide any personal information to get one.

To register for a merchant account in under 1 minute, go to the following URL:

<http://www.carrot.org/carrotpay>

As a minimum, you must select a currency, read and accept our terms and conditions and you can then register. You may like to initially gain confidence by working with our test currency called XXX. In this case we will give you 100 units of XXX to test-buy your first products so that you don't need to spend anything to get started. However, we don't recommend you use the XXX currency much after the creation of your first couple of products, because you will need to change all you filenames at the point when you finally use a merchant account for a real currency.

CarrotPay Merchant Registration page

Welcome to the growing community of merchants who are accepting **Carrot-WebCoins** for payment of on-line goods and services.

NOTE: The currency "XXX" is for testing only. This is also know as "Sandbox" testing. When you register for this currency you will also receive 100 units of XXX so that you may purchase your own products during development. You should use XXX in your products exactly as you would USD or GBP etc. Once your development and testing are complete, you will need to register for a real currency and substitute your real merchant ID into your products.

Three simple steps	Provide Optional data
1. Select a currency CNY ▾	Please tell us a little bit about yourselves and we will promote your site in our directory.
2. Agree to our terms and conditions <input type="button" value="View Terms"/>	Enter a contact email address. <input type="text"/>
3. Review data and confirm	Enter your web site url. <input type="text"/>
	Enter the major categories for your site (e.g. Games). <input type="text"/>
	Enter a url for your logo. <input type="text"/>

The choice of currency at this stage sets the currency of all future requests for this merchant. Therefore, if you need to bill in more than one currency you must register once for each currency needed. Note however, buyers will automatically have your prices converted into the currency of their own wallets, so multi-currency registration is not really necessary.

On the right of the form you will notice that there are a number of fields which you may optional fill in. If you do provide this data, CarrotPay will use it to promote your site to its customers without charge to you – so its probable a good idea to enter accurate data.

When you click on the “Review details” button, you will be directed to a page to review your information. You may change your details if you need to before finally configuring your CarrotPurse.

CarrotPay new merchant ready

Merchant ID	KWDW-LQQF-CGGQ-SDPQ
Select a currency	USD
Enter a contact email address.	john.smith@biggamesite.com
Site URL	www.biggamesite.com
Enter a url for your logo.	www.biggamesite.com/images/logo.png
Enter the major categories for your site (e.g. Games).	Games

Use the Merchant ID in your HTML payment forms.

You must now configure your CarrotPurse to enable the collection of WebCoin payments. All essential data will be stored in a new Merchant wallet in your purse.

Configuring your CarrotPurse causes a special 'wallet' to be created which has merchant functions enabled. A merchant ID, seed and secret key (as described above), are stored in this wallet and you are now registered with CarrotPay and your CarrotPurse has everything it needs to interact with the CarrotPay 'Button Wizard' and to collect the Carrot-WebCoins as they come in.

Stage 2 - Creating Protected Links and secure file names

Once you have registered as a CarrotPay merchant, your new wallet can help you to create a Protected Link for each of your content files. But first we need to delve a little deeper into exactly how this works...

Both the CarrotPay service and your merchant wallet are set up with a secret 'seed' value which is used when modifying filenames. This seed and the requested price are both used in the modification process of the original name into a secure name. The modification process is arranged so that if someone externally doesn't know the seed, they can't guess what the original filename will turn into, and neither can they change the price to get it cheaper.

Notice that the filename is only part of a URL and other parts need to remain unmodified; such as the domain name, and any directory structure you have on the site. So how does CarrotPay know which bits to modify and which to let alone?

The rule is simple: For a Protected Link, any text inside square brackets “[...]” is modified, and the rest is left alone. So the Protected Link

```
http://www.mysite.com/photos/beach_[beach].jpg
```

might get turned into the secure filename

```
http://www.mysite.com/photos/beach_bjmzqpxg.jpg
```

Note how the square brackets have been removed as well, and what you end up with is a slightly odd looking but standard format URL.

But how do you know what the secure modified text will be? Your CarrotPurse can help you here: On the Merchant Information page, which you can reach by clicking "Information" on the merchant wallet, there is a button which sends you to a "Button Wizard" Webpage with 4 steps.

Step-1: Your merchant information will be pre-filled in this form, so all you need to do is select the button type 'Protected Link' (which is the default), then click Next >> to move to Step-2.

Select wizard type:

- Protected Link (Simple Buy Button)
- Protected Content
- Go to www.51carrot.com to sell products

Step 1 of 4 - Enter Merchants Details

All the data required for these fields can be found in the Information window of your Merchant wallet. We recommend that you use the auto-fill facility provided by CarrotPurse: 'CarrotPurse » Home » Information » Button Wizard'. Start your CarrotPurse, select 'Home', click 'Information' then 'Button Wizard'. This Wizard page will be loaded into your browser and all the fields will be filled from your Wallet data. Alternatively, you may copy and paste the data individually from the Wallet to this Web form.

Merchant ID:	<input type="text" value="HHPQ-PVWQ-WSKV-KCI"/>	Example: "LBVJ-LWKV-JWCJ-RGSS".
Currency:	<input type="text" value="XXX"/>	Example: "RMB".
Hash Seed:	<input type="text" value="gVlsqpchzvmrhdmw"/>	Example: "mjqpbczjwxmbpzb". Your Hash Seed is like a password and must be kept secret.
Service URL:	<input type="text" value="http://payment.carrot.org/"/>	Example: "http://www.carrot.org/payment".

Step-2: Enter the original product filename (without square brackets), and price followed by the content URL (which is your domain name and directory structure). Finally, you may enter the URL of a page to send the buyer to if an error occurs.

Notice that the 'Secure data' field changes as you type the filename and price.

Protected Link (Simple Buy Button) Wizard

Step 2 of 4 - Enter Product Details

Enter your product details here. You must enter the following information about the product you want to sell and the price charged in the currency associated with your merchant ID.

Product Filename:	<input type="text" value="coolpic.jpg"/>	The name of the file containing this product (including file extension).
Price (XXX):	<input type="text" value="1.50"/>	The amount to be charged (From 0.001 to 999.999).
Secure Data:	<input type="text" value="phgkczd"/>	Product/Filename hash.
Content URL:	<input type="text" value="http://www.mysite.com/co"/>	This is the address to where you will upload your file.
Failure URL:	<input type="text" value="http://www.mysite.com/fai"/>	The page to be displayed if a purchase is canceled or there is a problem during the payment.

Short Product Description:

Note that the format of the price is critical at this stage. You must type it in exactly the same format when you come to create the HTML later (actually the wizard does this automatically). So you can't put "10.00" here and "10" later – that will generate different secure data in the URL and your purchase system won't work. You can try this by changing the price and you can see the secure data changing as you do it.

Step-3: Choose a button style that suits your site taking into consideration the size and shape of the button.

Step-4: Review and copy the HTML form data and filename. Notice that the file name now looks something like this 'beach_bjmqpxg.jpg'. An underscore character and the new secure data part have been inserted between the original filename and the file extension. We found this to be a handy way to secure the file but still leave it recognisable for easy identification.

Protected Link (Simple Buy Button) Wizard

Step 4 of 4 - Review and Copy Form or Link Data

Original File Name:	coolpic.jpg
Price:	1.50
Description:	Cold as ice
New Filename:	coolpic_phgkczd.jpg
Content URL:	http://www.mysite.com/content/
Return URL:	http://www.mysite.com/content/coolpic_[coolpic].jpg
Failure URL:	http://www.mysite.com/fail.html

HTML Form

Copy and past the HTML code into your website.

```
<form action="http://payment.carrot.org/payment" method="post">
<input type="hidden" name="merchant" value="HHPQ-PVWQ-WSKV-KCLW">
<input type="hidden" name="total" value="1.50">
<input type="hidden" name="description" value="Cold as ice">
<input type="hidden" name="return_url"
value="http://www.mysite.com/content/coolpic_[coolpic].jpg">
<input type="hidden" name="fail_url"
value="http://www.mysite.com/fail.html">
```

Buy Now

Email Link

If you want to just use a simple text link or give a link via E-Mail, you can use the URL below. This will not show the button or an image and should just appear as plain text or as a link.

```
https://www.carrot.org/payment?merchant=HHPQ-PVWQ-WSKV-
KCLW&total=1.50&description=Cold as ice&return_url=http%3A%2F%
2Fwww.mysite.com%2Fcontent%2Fcoolpic_%5Bcoolpic%5D.jpg&fail_url=http%
```

[Click here to buy now](#)

You can try out your new button right on the Wizard page. Just click the 'Buy Now' and your CarrotPurse should ask you to authorise the payment (assuming your zero-click is not set to at least that value).

You can also use the link form of the payment request to send in emails etc., just copy and paste that same as for the HTML FORM.

Stage 3 - Creating the button

The final stage to enabling CarrotPay payments on your site is to add “Pay with Carrot-WebCoins” buttons. These are simple HTML forms which you can either use a 'wizard' to create (as was just described), or create manually yourself.

In both cases, there are some pieces of information that need to be included in the form:

1. The CarrotPay payment service URL: **http://payment.carrot.org/payment**
2. Your merchant ID (XXXX-XXXX-XXXX-XXXX)
3. The price you want to charge, in the currency of your merchant account, *in the same format as when you generated the secure URL*.
4. A description of the product, which will be shown to the customer when they authorise payment. **NOTE:** *For non-latin languages, you must use the UTF-8 encoding standard. To do this ensure that your HTML editor is set to save with UTF-8 encoding.*
5. The original, unmodified URL of the content, including the parts in square brackets.
6. A failure URL to return the customer to if payment authorisation fails.

Using the wizard

The final stage of the Button Wizard produces all the data you need to copy and paste into your Webpage and to rename your content files.

Writing the form manually

The HTML of the form is also quite simple to create manually. It is a standard HTML FORM with a single submit button and number of hidden inputs:

Hidden input name	Description	Example
merchant	Merchant ID (16 letters with dashes)	"KBCS-ZMSW-PDRZ-DVRH"
total	Price to charge, in the currency of your merchant account	"10" → Range: 0.001 to 999.99
description	Short (one line) description of the product	"Image from Trehaverne Image Library"
return_url	Unmodified URL with square brackets	"http://www.trehaverne.org/photos/[beach].jpg"
fail_url	Failure URL	" http://www.trehaverne.org/failed.html "
target	Javascript window/frame to receive return or fail URLs (optional)	"top"

Here is the code for a simple text button which you can fill in yourself:

```
<form action="http://payment.carrot.org/payment" method="POST">
  <input type="hidden" name="merchant" value="(your merchant ID)">
  <input type="hidden" name="total" value="(price)">
  <input type="hidden" name="description"
    value="(product description)">
  <input type="hidden" name="return_url"
    value="(unmodified URL with square bracket sections)">
  <input type="hidden" name="fail_url"
    value="(URL for failure page)">
  <input type="submit" value="Pay with Carrot-WebCoins">
</form>
```

Here it is filled in with the above example data:

```
<form action="https://payment.carrot.org/payment" method="POST">
  <input type="hidden" name="merchant" value="KBCS-ZMSW-PDRZ-DVRH">
  <input type="hidden" name="total" value="10">
  <input type="hidden" name="description"
    value="Image from Trehaverne Image Library">
  <input type="hidden" name="return_url"
    value="http://www.trehaverne.org/image/beach_[beach].jpg">
  <input type="hidden" name="fail_url"
    value="http://www.trehaverne.org/failed.html">
  <input type="submit" value="Pay with Carrot-WebCoins">
</form>
```

You will likely find that it is quite a simple task to copy and paste the form data and to then change certain of the input fields in a text editor. If you find this method suitable for you, then you will only have to use the Button Wizard to determine the secure filenames. You may find it more efficient to proceed to Step-2 of the wizard and then process a batch of filenames by sequentially copying and pasting the original filenames and copying the resulting secure names from the wizard to your HTML pages.

Improving the look

When a simple payment form (as above), is submitted to CarrotPay, the service generates an HTML page which contains some 'magic' Javascript to request payment from the customer's CarrotPurse, and then proceeds to the (now modified) , 'return_url' if it is successful, or the 'fail_url' if not. The problem with this simple approach is that the page containing the Javascript is otherwise completely blank, so the customer won't know what's going on. The solution is to make the form post into a hidden IFRAME, like this:

```
<form action="http://payment.carrot.org/payment" method="POST"
      target="working">
  ... form inputs as above ...
</form>

<iframe name="working" src="" style="display: none"></iframe>
```

Note: The iframe is hidden because its style attribute is set to "display: none".

The thing to recognise here is that the 'target' attribute in the form element is set to the same value as the 'name' attribute of the iframe element (i.e. "working"). You may use any value in the quotes of course – as long as the target and name are identical. Using this method, the blank page is hidden from the user, but still allows the Javascript to run. However the fact that the result of the payment is hidden is a problem because the final objective (i.e. to view the content), is naturally loaded into the invisible iframe!

The solution to this problem is to set a 'target' input in the form. You can set the target to the name of any Javascript window or frame and the Javascript will then force the new page to be loaded into that window or frame (using a statement like "<target>.location.replace(<url>)"). In the example above, if you were to add in an input with the name 'target' and a 'value' set to "top", the return or fail page will be loaded over the top of the original page containing the form.

Hence the completed button form becomes:

```
<form action="http://payment.carrot.org/payment" method="POST"
      target="working">

  <input type="hidden" name="merchant" value="KBCS-ZMSW-PDRZ-DVRH">
  <input type="hidden" name="total" value="10">
  <input type="hidden" name="description"
        value="Image from Trehaverne Image Library">
  <input type="hidden" name="return_url"
        value="http://www.trehaverne.org/image/[beach].jpg">
  <input type="hidden" name="fail_url"
        value="http://www.trehaverne.org/failed.html">
  <input type="hidden" name="target" value="top">
  <input type="submit" value="Pay with Carrot-WebCoins">

</form>

<iframe name="working" src="" style="display: none"></iframe>
```