



# Upgrading Your Store to ProductCart v3

---

## ***Table of Contents***

Overview .....	2
Preparing Your Store for the Upgrade .....	2
Uploading Files to Your Server .....	3
Running the Upgrade Wizard .....	3
Updating Header.asp .....	5
Dynamic Meta Tags in ProductCart v3 .....	6
Editing Store Elements Added to the Interface.....	7
Adding or Editing a Search Box .....	8
Adding or Editing Category Navigation .....	8
Links to Content Pages and Other Key Storefront Pages .....	9
Adding or Editing a Shopping Cart Summary .....	10
Adding or Updating a Login Box .....	11
Updating store links (301 Permanent Redirects) .....	12
ProductCart v3 and CSS .....	12
Overview .....	12
Loading the style sheets.....	12
pcStorefront.css .....	13
CSS resources.....	13

---



## Overview

Congratulations on your decision to upgrade your store to ProductCart v3.

ProductCart v3 is the biggest upgrade to our shopping cart software that we have ever released. It will make your store look more professional and run more efficiently. You will find dozens of new store management features and completely redesigned storefront pages to make your products and services sell even faster!

Because ProductCart v3 represents such a substantial upgrade of your e-commerce store, you will have to carefully review the following Upgrade Instructions to make sure that the process is a smooth one. We recommend that you print this document and keep it with you throughout the upgrade process.

Upgrading to ProductCart v3 will require several few steps.

You should expect to spend 1 to 3 hours upgrading your store. Apart from the time necessary to upload the new files to your Web server, the total time spent upgrading your store will depend on a number of factors including how much your interface (pc/header.asp and pc/footer.asp) will need to be edited, how many other Web site pages need to be updated, and more. In addition, you should expect to spend some time familiarizing with some of the great new features included in ProductCart v3

Since your store will be offline while you perform the upgrade, you might want to begin the process at a time when traffic to the store is limited.

If a Web designer or developer helped you get your store up & running, we strongly recommend that you hire them again to perform this upgrade. This is a must if the ProductCart source code was customized. But even if the source code was not customized, you should consider hiring a professional if your store interface contains dynamic elements (e.g. dynamic navigation) and you are not sure that you will be able to update them. This document does provide the information you need to make those changes.

## Preparing Your Store for the Upgrade

Follow these steps to prepare your store for the upgrade.

- **Turn the store off**  
You can do so from the General Settings > Store Settings page in the Control Panel. For example, you could enter the following message: *" We are currently updating our e-commerce system. The store will be offline for a few hours. We appreciate your patience."*
- **Back up the database**  
To backup the database:
  - If you are using an Access database, simply download a copy of the database to your local system.
  - If you are using a SQL database, depending on how the database has been setup, you may be able to either perform the backup yourself or have your Web hosting company do it for you. Note: your SQL database is likely being automatically backed up every day: confirm that this is the case by asking your Web host when the last back occurred.
- **Back up the store files**  
FTP the entire store to your desktop.



### **Uploading Files to Your Server**

Use your favorite FTP client to upload the v3 files to your Web server. Several hundred files will be uploaded, so the process will likely take a few minutes. Before you get started, please note:

- **Parent Paths Enabled vs. Disabled**  
Make sure to select the correct files depending on whether your store is running on a server that has Parent Paths Enabled or Disabled. If Parent Paths are Disabled, then when you log into the Control Panel, the version number should include the letters "PPD".
- **The PCADMIN Folder**  
The Control Panel folder - by default named "pcadmin" - is always renamed on a ProductCart store, for security purposes. So when you FTP the files, make sure to move the contents of the "pcadmin" folder that is part of the upgrade files into the right folder on your Web server. Otherwise a new, "pcadmin" folder will be created on the server, and your store will not be upgraded to v3. A useful tip:
  - Rename the "pcadmin" folder on your desktop using the name used on the Web server
  - This way you can select all folders that are part of the Upgrade files and upload them all to the Web server at the same time.
- **Overwritten Files**  
Make sure to back up any files that were customized, so that you can later integrate your custom code with the new files. Among other files that will be overwritten by the v3 Upgrade is **pc/orderComplete.asp**. This is a file that is often customized by ProductCart users as they add tracking code for advertising systems like Google AdWords or Yahoo! Search Marketing. Remember to edit the new version of the file to re-add that code.
- **Default Meta Tags**  
Back up the file *include-metatags.asp* located in the *pc* folder. This file will be overwritten as part of the upgrade. You will need to re-enter the default Meta Tags once you are done with the upgrade, and you can do so by copying them from your back up copy of the file. See [Dynamic Meta Tags in ProductCart v3](#) for more details.

### **Running the Upgrade Wizard**

Once you have uploaded the v3 Upgrade files to your server via FTP, login into the Control Panel. You will be automatically redirected to the ProductCart v3 Upgrade Wizard. This Wizard performs the following tasks:

- **Update Store Database**  
A script will be run to add new tables and fields to your database. After the database has been updated, we will update the store settings, including the version number. You will be able to review all new store settings at the end of the Wizard.
- **Convert Product Options**  
ProductCart v3 allows you to assign unlimited options to your products. Existing product options will be converted to the new database structure. On stores with thousands of products and/or orders, this task may take some time to execute.
- **Convert Checkout Options**  
ProductCart v3 allows you to create unlimited "Special Customer Fields". These used to be limited to 2 fields and were set in the "Manage Checkout Options" section. A script will convert them to the new database structure. On stores with thousands of customers, this task may take some time to execute.

## Upgrading your store to ProductCart v3



- **Update Store Settings**  
Your store settings will be updated, including the version number.
- **Review New Features & Update Related Settings**  
Changes have been made to many areas of ProductCart. There are a few areas where you need to review and update your store settings.

Please follow the instructions provided in the Wizard for more information. Note that when you are done running the Wizard, you are not done with the upgrade. You should review and perform the following, additional tasks before re-opening your store to the public.

A few, particularly important things to note:

- **Payment Options**  
Regardless of which payment option(s) you are using, review your settings and especially the new "Process orders when they are placed" and "When orders are placed, set the payment status to..." features. Details for both features are included in the new Control Panel online help system.
- **FedEx®**  
ProductCart v3 contains a completely new integration with the FedEx API. You will be able to use a number of FedEx services, including the ability to print shipping labels right from the Control Panel. The v3 Upgrade Wizard will save your FedEx shipping settings have been saved and temporarily deactivate the shipping provider. To re-activate FedEx you will need to obtain a FedEx® API *meter number*. ProductCart will guide you through the registration process.
- **UPS®**  
The integration with UPS has also been substantially improved. ProductCart now uses additional UPS Online Tools and connects to the UPS Shipping API to allow you to print shipping labels right from the Control Panel. The v3 Upgrade Wizard will disable UPS in your store (if it's active) and alert you to re-register to take advantage of the new features. ProductCart will guide you through the registration process.
- **New Buttons**  
ProductCart v3 uses additional buttons in the storefront. Make sure to upload new buttons to your store to match your existing ones. Updated, editable Fireworks (PNG) graphics are included in the *pc/images/sample* folder of the v3 Upgrade files.
- **Store and Display Settings**  
Should AJAX be enabled on your storefront? And what's AJAX anyway? This and a number of other, new store and display settings have been added to your store and you should review them right after upgrading to v3. Details for each of them are provided through the new online help system in the Control Panel.



## Updating Header.asp

The code at the top of *header.asp* will need to be slightly updated for v3.

In many areas of this document we refer to the "default" *header.asp* and *footer.asp* files that are included in ProductCart v3. When you first install ProductCart, these are the files that make up the graphical interface of the storefront.

Since you are upgrading your store, these files are not included in the upgrade as they would overwrite your own. If you want to use them as a reference, you can download them here.

[http://www.earlyimpact.com/productcart/support/ProductCart\\_v3\\_header\\_footer.zip](http://www.earlyimpact.com/productcart/support/ProductCart_v3_header_footer.zip)

The very top of your *header.asp* in a ProductCart v2.76 store typically reads as follows:

```
<!--#include file="include-metatags.asp"-->
<html>
<head>
  <%GenerateMetaTags()%>
  <%Response.Buffer=True%>
  <%
    Set conlayout=Server.CreateObject("ADODB.Connection")
    conlayout.Open scDSN
    Set RSlayout = conlayout.Execute("Select * From layout Where layout.ID=2")
    Set rsIconObj = conlayout.Execute("Select * From icons WHERE id=1")
  %>
```

You will need to replace these lines of code with the following:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--#include file="include-metatags.asp"-->
<html>
<head>
  <%if pcv_PageName<>" then%>
    <title><%=pcv_PageName%></title>
  <%end if%>
  <%GenerateMetaTags()%>
  <%Response.Buffer=True%>
  <%
    Set conlayout=Server.CreateObject("ADODB.Connection")
    conlayout.Open scDSN
    Set RSlayout = conlayout.Execute("Select * From layout Where layout.ID=2")
    Set rsIconObj = conlayout.Execute("Select * From icons WHERE id=1")
  %>
  <link type="text/css" rel="stylesheet" href="pcStorefront.css" />
```

**Note:** If there are extra lines or other characters in the code, ProductCart may not function properly. Sometimes pasting from a MS Word or PDF document directly into your HTML editor can create problems. Instead of copying and pasting the code from this document, we recommend that open in your HTML editor the default *header.asp* and *footer.asp* files located in the *pc* folder, and copy the code from there. Or you can find the same code in [this ZIP file](#).

If you update your *header.asp*, load a ProductCart page (after you have completed the upgrade process), and receive an error, then probably the culprit is extra characters that were added to the



code when you copied it and pasted it. Make sure that you are using the correct code. You can find it in [this ZIP file](#).

The lines located at the bottom of *footer.asp* do not need to be updated. Just make sure that they are correct. They should read as follows.

```
<%  
conlayout.Close  
Set conlayout=nothing  
Set RSlayout = nothing  
Set rsIconObj = nothing  
>  
</body>  
</html>
```

### ***Dynamic Meta Tags in ProductCart v3***

Just like in previous versions of ProductCart, the code that you have added to *header.asp* will take care of automatically creating the Title, Description, and Keyword Meta Tags for every page in the storefront. However, ProductCart v3 contains additional features to give you more flexibility and control.

Specifically, in ProductCart v3 Meta Tags are generated as follows:

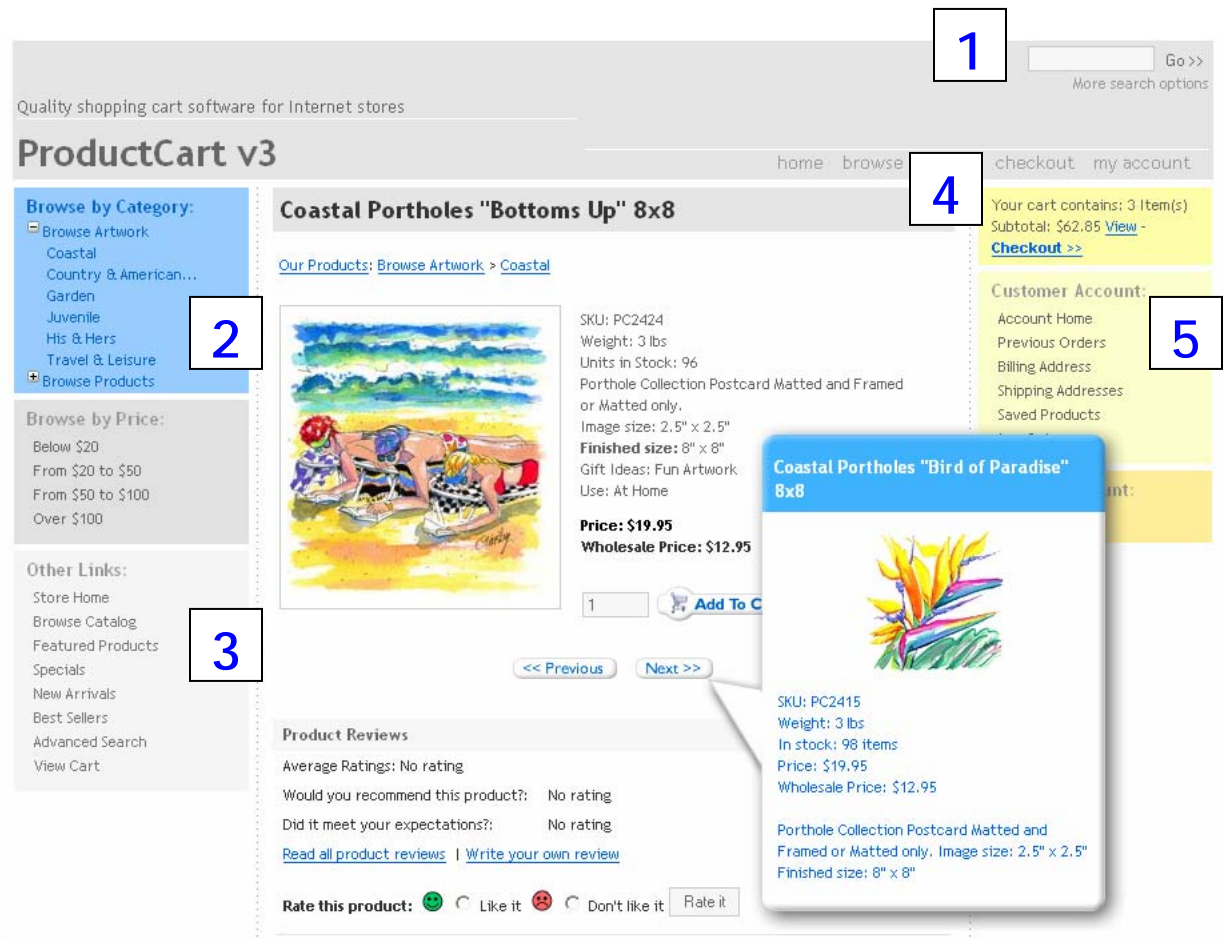
- **Store Manager-Specified Meta Tags.** If Meta Tags have been specified for a product or a category using the ProductCart Control Panel (when adding or modifying a category or a product), then that information is used on the corresponding product or category page. This is a new feature in version 3.
- **Automatically Generated Tags.** If Meta Tags have not been specified for a product or a category, ProductCart automatically generates them for those products or categories, using the category/product name for the Title, and a portion of the category/product description for the Description tag. If you have thousands of products and therefore are unable to manually enter optimized Meta Tags for each of them, this feature provides a sensible solution.
- **Default Meta Tags.** For pages other than product and category pages (e.g. "View shopping cart", "Search", etc.), ProductCart uses the default Meta Tags that are found in the file *pc/include-metatags.asp*. **This file was overwritten during the upgrade** because the ASP source code needed to be updated. This means that **you need to re-enter your default Meta Tags**. To edit this file, do the following:
  - Download *pc/include-metatags.asp* to your desktop using your favorite FTP program.
  - Open it in Notepad or an HTML editor, and edit the default tags at the top of the page.
  - Save it and upload it back to your Web server.

## Editing Store Elements Added to the Interface

To further integrate your Web store's graphical interface with ProductCart, many stores add a number of static and dynamic elements to *header.asp* and *footer.asp*. By static, we mean an interface element that doesn't change depending on what the user or the store manager does (e.g. a search box). By dynamic, we refer to an interface element that does change.

Take a look at the picture shown below: this demo store uses the default *header.asp* and *footer.asp* files that come with ProductCart. These files contain a number of these interface elements. Let's look at what they are and how you can use them in your version of *header.asp* and *footer.asp*.

1. **Search box.** You can let your customers run a keyword search from anywhere on your site.
2. **Category Navigation.** You can use the Control Panel to dynamically generate a category tree that is then shown in your storefront.
3. **Content Pages and Other Links.** You can automatically show a link to any page generated via the "Manage Content Pages" feature, and use the "Generate Links" feature to create static links to other pages (e.g. New Arrivals, Best Sellers, etc.).
4. **Shopping Cart Total.** You can automatically display the shopping cart total, when something has been added to the shopping cart. The cart summary is hidden if the cart is empty.
5. **Customer Login/Account Links.** You can automatically switch between a login link and an account menu (shown once customers have logged in). You can do the same with Affiliates.





Let's take a closer look at each of these 5 interface elements. If you are looking for other ideas, checkout the ProductCart forums at <http://www.earlyimpact.com/forum> or visit the Developer's Corner at <http://www.earlyimpact.com/productcart/support/developers.asp>.

## Adding or Editing a Search Box

You can easily display a "search box" in your HTML code by including into *header.asp* the file *SmallSearchBox.asp*, which is located in the *pc* folder. To do so, simply add the following line of code to the section of your HTML code where you want the search box to appear (e.g. a table cell):

```
<!-- #Include file="SmallSearchBox.asp"-->
```

If you need to edit the way the search box is presented in your storefront, either open the file *SmallSearchBox.asp* itself in your favorite HTML editor, or edit the main style sheet used by the storefront (*pcStorefront.css*) and edit the classes that pertain to the search box.

If you decide to edit the file itself, you will see that the code reads as follows:

```
<form action="showsearchresults.asp" name="search" method="get" class="pcForms">
<input type="hidden" name="pageStyle" value="<%=bType%>">
<input type="hidden" name="resultCnt" value="10">
<input type="Text" name="keyword" size="14" value="">
<input type="submit" name="submit" value="Go &gt;&gt;" id="submit">
<div class="pcSmallText">
  <a href="search.asp">More search options</a>
</div>
</form>
```

Compare the code above with the one used on your store, if any, and make the required changes.

A couple of quick edits can help you change the way the search box behaves:

- Change the *value* of the *pageStyle* hidden field from *<%=bType%>* (this code is retrieving the default page style used in your storefront) to one of the following values to change the way search results are presented:
  - h = search results are shown horizontally
  - l = search results are shown in a list
  - m = search results are shown in a list, with the ability to add multiple products to the cart
  - p = search results are shown vertically
- Change the *value* of the *resultCnt* hidden field from "10" to any other integer to change the number of results shown on the search results page (e.g. "12").

## Adding or Editing Category Navigation

In ProductCart v3 you can add category navigation to your store interface by simply adding the following "include" statement.

```
<!--#include file="inc_catsmenu.asp"-->
```

You should remove the code that currently handles dynamic navigation in your store interface and replace it with this simple line of code. ProductCart will take care of the rest. Specifically...





- Using the **Settings > Generate Navigation** feature in the Control Panel, you can create the category tree. You can recreate the tree whenever you add, remove or rearrange your categories, to keep it updated. This is a new feature in ProductCart v3. It provides a huge performance boost on stores that contain a large number of categories and products, as the category tree is not built on the fly whenever a customer visits the storefront, but rather built once in the Control Panel, saved to a text file, and then used over and over again in the storefront, without performing any calls to the database.
- By adding the line of code shown above to the section of your HTML document where you want the navigation to appear (e.g. a table cell), you will tell ProductCart to display the category tree there. You can assign a CSS class to the code that is generated by this feature to maintain control on the way the navigation links are displayed.

### How it works

ProductCart will generate two static files to store your category and/or product names and locations: one for retail customers, and one for wholesale customers (this is because the Control Panel contains a feature that allows you to hide some categories from retail customers, but show them to wholesale customers: so the navigation might be different for the two customer types).

These files will be used to generate the navigation. This approach (using static files vs. dynamically generating the category tree every time) is used to eliminate redundant queries to the store database to collect the same information over and over (i.e. navigation links remain the same until you change the category tree, and it is a waste of server resources to generate the category tree on the fly every time it is used in the storefront). This also means that the navigation will not change automatically when you add/edit products and/or categories. Remember to rerun this feature whenever you update your product catalog (e.g. you add a new category).

### To Create or Update your navigation

Log into the Control Panel, select *Settings > Generate Navigation*, and choose one of the following three options:

1. Include categories and sub-categories, but no products. Customers will be able to expand/collapse the category tree to see the sub-categories. No product links are shown in the category tree.
2. Include only top-level categories and their products. Only the categories listed in the root are shown in the navigation.
3. Include categories, sub-categories, and their products.

If you choose (2) or (3), you will also need to specify the *Maximum number of products per category*. A "More products..." link is automatically added if there more products in the category than the number specified here.

For added flexibility, you can add a CSS class to the navigation code: it is applied to the table cells that are generated when the category tree is built. Simply enter the name of the class in the corresponding input field.

### Links to Content Pages and Other Key Storefront Pages

"Content Pages" are generated by using the "Manage Content Pages" feature in the Control Panel, which is a basic content management system included in ProductCart. For example, you can use ProductCart to create a "Customer Service" or "Frequently Asked Questions" page right from within the Control Panel.

Let's take a quick look at the code needed for showing these links. Compare with the one used on your store, if any, and make the required changes.

```
<% ' Show List of Content pages
sdquery="SELECT pcCont_IDPage,pcCont_PageName FROM pcContents where
pcCont_InActive=0;"
set rsSideCatObj=conlayout.execute(sdquery)
do while not rsSideCatObj.eof
%>
<li><a
href="viewcontent.asp?idpage=<%=rsSideCatObj("pcCont_IDPage")%>"><%=rsSideCatObj("pcCo
nt_PageName")%></a></li>
<%
rsSideCatObj.MoveNext
loop
set rsSideCatObj=nothing
%>
```

This code retrieves from the database the list of active Content Pages, and then creates a list of links to them. As mentioned elsewhere in this document, if you want to use this code, we recommend that you copy and paste it directly from *header.asp* using by opening the file in Notepad or your HTML editor. If you copy it from here, the ASP code might not function properly in your file due to extra characters or line breaks that might be created during the “copy & paste” process.

If you are wondering why the code uses the HTML “unordered list” tag (<li>), the reason is that with CSS you can manipulate a list to work great in a navigation area. Here, a cascading style sheet is used to remove the bullets from the HTML list (<li>). The styles that perform this task are included in the CSS file “pcHeaderFooter.css”, which is used by the default *header.asp* and *footer.asp*.

Many stores also link to include a links to pages such as Specials, New Arrivals, etc. You can generate the correct URLs to these pages using the **Generate Links** feature in the Control Panel. For example, some of these links are:

```
<a href="home.asp">Featured Products</a>
<a href="showspecials.asp">Specials</a>
<a href="shownewarrivals.asp">New Arrivals</a>
<a href="showbestsellers.asp">Best Sellers</a>
<a href="search.asp">Advanced Search</a>
<a href="viewcart.asp">View Cart</a>
```

Note that some of these files have been renamed in v3.

### Adding or Editing a Shopping Cart Summary

Your customers will appreciate having a quick summary of what they have added to their shopping cart shown on every storefront page, so that they know at all times what the total of their order is at that time.

This useful code snippet shows how many items have been added to the cart, the current order total, a link to view the shopping cart details, and one to start the checkout process.

In ProductCart v3, you can add this dynamic interface element to your store interface simply by including the following line of code (you will find it in the default *footer.asp* file):

```
<!--#include file="SmallShoppingCart.asp"-->
```

## Adding or Updating a Login Box

Many stores use a “login box” on their *header.asp*, *footer.asp*, or even on other Web site pages. We are referring to a small form that contains a user name field, a password field, and a login link or button. Since the login procedure has changed in v3, you will need to update this code.

A login box in a store running ProductCart v3 should contain the following code:

```
<form name="login" method="post" action="checkout.asp?cmode=1">
  User Name <input type="text" name="LoginEmail">
  Password <input type="password" name="LoginPassword">
  <input name="PassWordExists" type="hidden" value="YES" >
  <a href="JavaScript:document.login.submit();" ></a>
  <input type="hidden" name="SubmitCO.y" value="1">
</form>
```

Compared to the code that is currently on your page (if any):

- “PassWordExists” is a new field and needs to be passed with a value of “YES”
- The login form should point to “checkout.asp?cmode1” instead of “custva.asp”
- The field names are now “LoginEmail” and “LoginPassword” instead of “email” and “password” .
- The “SubmitCO.y” field should remain the same.





### Updating store links (301 Permanent Redirects)

Some of the files used by ProductCart v2.76 are no longer used in v3. In many cases, multiple files have been consolidated into one, more flexible file. This can create two issues:

- There might be links in your Web site that now point to “old” pages. Although many of these pages will load without error, they are not v3 pages, and therefore may not behave properly (e.g. if you are using *Pricing Categories*, a new feature in v3, the correct prices would not be shown).
- Search engines might have spidered your old pages. How can you tell them to switch to the new pages, without penalizing your search engine rankings? The appropriate way to do it is by using “301 Permanent Redirects”. Google specifically endorses and recommends this technique.

Therefore, we have created a number of files that you can upload to your storefront and that will “permanently” redirect to the corresponding file in the v3 store, preserving the display type (e.g. products shown vertically vs. horizontally) and category ID (“viewCat” pages).

Once again: the code uses the redirect procedure that is recommended by all of the major search engines (301 Permanent Redirect). It communicates to the search engine spider that the page has been permanently relocated to a new location.

You can download a ZIP file that contains these pages from the following URL.

[http://www.earlyimpact.com/productcart/support/ProductCart\\_v3\\_Upgrade\\_301\\_Redirects.zip](http://www.earlyimpact.com/productcart/support/ProductCart_v3_Upgrade_301_Redirects.zip)

These pages are not contained in your upgrade files as some ProductCart-powered stores may have heavily modified some of these files and may not wish to overwrite them during the upgrade.

## ProductCart v3 and CSS

### Overview

New in ProductCart v3 is the use of cascading style sheets throughout the storefront. Virtually all <FONT> tags were removed and replaced with a number of different classes. Styles were also added to many other elements of the interface.

All styles are stored in a file called *pcStorefront.css*, which is located in the *pc* folder. The file has been heavily commented to help you understand how it affects the look and feel of your store.

All ProductCart users will be able to easily apply basic interface changes to their stores (e.g. setting the font family to “Verdana” or the font color to a certain shade of blue), without needing to be proficient in the use of cascading style sheets. How? All of the interface properties that were previously set through the Control Panel’s Display Settings page (e.g. the main font type and color) have been turned into CSS styles and are grouped together at the top of *pcStorefront.css*, as mentioned below.

Web designers and other advanced users will find that this new version of ProductCart gives them a much larger degree of flexibility in altering the look of their stores, thanks to the use of CSS.

First, let’s make sure that the cascading style sheet used by ProductCart is correctly loaded in your store.

### Loading the style sheets

The first line of code in *header.asp* must always be the following:

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

This line is very important: it defines the type of document that you are asking a browser to render and it also tells the browser how to interpret the cascading style sheets used on the store. Unless this statement is present - and correctly positioned at the very top of your header.asp file - the cascading style sheets used by ProductCart will not function properly.

Next, make sure that pcStorefront.css is loaded in your *header.asp* file by verifying that the following line of code appears before the closing `</head>` tag.

```
<link type="text/css" rel="stylesheet" href="pcStorefront.css" />
```

## pcStorefront.css

Let's now take a quick look at the top part of the CSS file to understand how basic interface changes can be made to your storefront.

The styles included under "MAIN Styles" are the ones that used to be controlled by the *Display Settings* area of the ProductCart Control Panel in previous versions of the software. These are the only styles that you will need to edit if you want to change the type of font used in the storefront, its size, its color, etc. Unless you want to, you DO NOT need to change any of the other styles.

```
#pcMain {
    font-family: "Trebuchet MS", Verdana, Arial, sans-serif;
    font-size: 11px;
    color: #333333;
    text-align: left;
    background-color: #FFFFFF;
    width: 100%;
}
```

This first section of the CSS file sets most of the store-wide interface properties that most users will need to edit. Any CSS editor will help you easily change the font family, size, color, etc. It is important that you use a CSS editor rather than trying to edit the file manually as CSS is very "picky" when it comes to interpreting the syntax of the file.

```
#pcMain a:link, a:visited {
    color: #0066FF;
}

#pcMain a:hover {
    color: #0000FF;
    text-decoration: none;
}
```

This other section of the file sets the color for store links. This is also something that was set using the Control Panel in previous versions of ProductCart.

By using CSS, you now have much more flexibility in setting the appearance of links. For example, you could edit the CSS document to specify a different appearance for all different states that a link can be in.

## CSS resources

If you are interested in learning more about Cascading Style Sheets, Wikipedia's section on the topic can be a good start. Here are this and other resources:

- [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- <http://www.w3schools.com/css/default.asp>

## Upgrading your store to ProductCart v3



Your favorite HTML editor will allow you to properly edit a *pcStorefront.css*. If you don't own an HTML editor such as Dreamweaver® or FrontPage® and are looking for one, here are a couple of inexpensive options:

- <http://www.blumentals.net/rapidcss/?refid=g1>
- <http://www.highdots.com/css-editor/>
- <http://www.newsgator.com/NGOLProduct.aspx?ProdID=TopStyle>