



The CartIt Commerce System Site Design Guide

Version 8.0

February 3, 2003

INTRODUCTION TO CARTIT CONCEPTS	3
WHAT ARE CFOS (CARTIT FORM OBJECTS)?.....	3
PRIMARY	3
EXTENSION.....	3
SYSTEM.....	3
CUSTOM.....	3
<i>What Are the CFOs' Names?</i>	4
HOW TO CREATE PRODUCT PAGES.....	4
<i>Basic Elements of a Product Page</i>	4
<i>The cartitadd CFO Can Be in a SELECT List</i>	4
<i>How Do I Let My Shopper Change the Quantity?</i>	5
<i>How Do I Add Options to a Product?</i>	5
<i>How Do I Add Text Options to a Product?</i>	6
<i>How Do I Create Up-Sell Items (Multiple Item Ordering)?</i>	7
<i>You Can Use the addit CFO in Many Ways</i>	7
<i>addit with Quantities</i>	9
<i>How to Make a "View Your Shopping Cart" Link</i>	9
ADDITIONAL CFO VALUES FOR 4.X+ GENERATION CARTIT.....	9
Taxable	9
Smart Tax.....	10
GETTING A BETTER UNDERSTANDING OF IT ALL.....	10
Thinking CFO.....	10
FORM Elements are FORM Elements.....	11
DEBUGGING PAGES.....	12
The Missing Form Container.....	12
The Out of Place Form Element.....	12
Using Improper Page Design.....	13
CARTIT CFO REFERENCE	14
CARTITADD.....	14
QUANTITY.....	14
EOP.....	14
TEXTADD	15
ADDIT.....	16
CARTITQA.....	17
CARTIT_DYNAMIC_FILE.....	17
CARTIT_SET_COOKIE	18
CARTIT_ITEM_REQUIRES_COOKIE.....	19
CARTIT_DISCOUNT.....	20
BOUNCE.....	21
BOUNCETO.....	21
DYNAMIC_CARTIT	22
CARTITQD.....	23
CARTITQDC	24

Introduction to CartIt Concepts

In order to use CartIt, it is strongly recommended that you have a general understanding of HTML forms. While many users have some knowledge of forms, most rely on Microsoft® FrontPage® or other HTML editors for site design. However, those users who know the most about HTML forms make the most of CartIt.

CartIt was originally designed as a product for professional e.commerce design firms. Since this time, CartIt has become known as an easy to use product for everyone. However, the more you know, the more you can do and this is the true power of CartIt!

What are CFOs (CartIt Form Objects)?

CFOs (CartIt Form Objects) are FORM elements of a specific NAME. CartIt has many CFOs. This is what makes CartIt so powerful to the developer; you have virtually unlimited methods for adding items to the shopping cart. This may seem a little confusing to you if you are new to CartIt, but once you see some examples, it will all seem very simple.

PRIMARY

The Primary CFO is required for adding a product to the shopping cart. Only one Primary (product) CFO can be added to the shopping cart at once. The name of CartIt's Primary CFO is 'cartitadd.'

EXTENSION

Extension CFOs extend the capabilities of the Primary CFO. A Primary CFO may add a T-shirt to the shopping cart, but an Extension CFO can add options such as shirt size, color and even custom imprinting.

SYSTEM

A System CFO is not related to a product. A System CFO usually tells CartIt to perform some sort of action. Please see the CFO Reference for more information on System CFOs.

CUSTOM

A Custom CFO is usually a derivative of a Primary, Extension or System CFO. A Custom CFO usually acts like one of those CFOs, but has some limitation or special feature that does not allow it to fall into those categories.

What Are the CFOs' Names?

It is important to note that CFOs have specific names; they are simple HTML form elements that have specific names that mean something to CartIt. If you do not use the correct CFO name then the CFO is no longer a CFO, it will simply be a generic form element that CartIt will ignore.

cartitadd	The Primary CFO
eop	Extension CFO
quantity	Extension CFO
addit	Custom CFO

How to Create Product Pages

Throughout this manual you will be introduced to the different CFOs. Your pages can be complex or simple; how you design your pages is completely up to you providing you adhere to the CFO standards defined here.

Basic Elements of a Product Page

1. Create a standard form container that points to CartIt.
2. Use the cartitadd Primary CFO.
3. Add a Submit button or image.

The following example assumes that we are selling a product called "ACME Explosives" for \$19.95 that weighs 5 pounds:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="ACME
Explosives^19.95^5">
<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
</FORM>
```

That's essentially all you need to know! Next, you would place some text describing the product and optionally add an image to this form. When the shopper clicks on the Submit button, 1 "ACME Explosives" would be added to their shopping cart that costs \$19.95 and weighs 5 pounds.

The cartitadd CFO Can Be in a SELECT List

Using the cartitadd CFO in a select list will allow you to have several products with varying descriptions, prices and weights!

In the following example, we are selling a Pentium 200 and a Pentium 300. The user can select the computer they wish to purchase:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
```

The CartIt Commerce System 8.0 Site Design Guide

```
<SELECT NAME="cartitadd">
<OPTION VALUE="Pentium 200^1200^40" SELECTED>Pentium 200 -- $1200.00
<OPTION VALUE="Pentium 300^3500^42">Pentium 300 -- $3500.00
</SELECT>

<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">

</FORM>
```

NOTE: The weight portion of this CFO can be excluded if you are not using a weight-based method of shipping.

The cartitadd CFO is always required when adding an item to the shopping cart! cartitadd is the Primary CFO and other CFOs only extend the capabilities of cartitadd.

How Do I Let My Shopper Change the Quantity?

To allow your shoppers to select a quantity, you simply use the quantity CFO:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
<INPUT TYPE="TEXT" NAME="quantity" VALUE="1" SIZE="3" MAXLENGTH="3">
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="ACME
Explosives^19.95^5">
<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
</FORM>
```

The above will allow shoppers to enter the quantity they wish to purchase. You can also use a SELECT list for the quantity, just provide a list of quantities.

How Do I Add Options to a Product?

To add options to products use the eop (extended option) CFO.

We will use a T-shirt example for the eop CFO. The eop CFO will be used for the shirt size to charge an additional \$2.95 for the XX-Large T-shirt and increase the weight by 1 pound:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
```

Order a T-Shirt for \$19.95

```
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="T-Shirt^19.95^5">
```

Select a Color:

```
<SELECT NAME="eop_Shirt Color">
<OPTION VALUE="Red^00.00^0">Red
<OPTION VALUE="Blue^00.00^0">Blue
<OPTION VALUE="Green^00.00^0">Green
<OPTION VALUE="Black^00.00^0">Black
<OPTION VALUE="White^00.00^0">White
</SELECT>
```

Above, we used the eop CFO, but have set the price and weight to 0. We do not charge for colors. Below, we will do the same, but we will charge \$2.95 for an XX-large T-shirt:

```
Select a Size:
<SELECT NAME="eop_Shirt Size">
<OPTION VALUE="Small^00.00^0">Small
<OPTION VALUE="Medium^00.00^0">Medium
<OPTION VALUE="Large^00.00^0">Large
<OPTION VALUE="X-Large^00.00^0">X-Large
<OPTION VALUE="XX-Large^2.95^0">XX-Large -- Add $2.95
</SELECT>
<INPUT TYPE="SUBMIT" VALUE="Order!">
</FORM>
```

How Do I Add Text Options to a Product?

To allow shoppers to add personalized text to their order use the textadd CFO. Using the T-shirt example, we will let the shopper have their name printed on the front of the shirt and optionally their company printed on the back:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

Order a T-Shirt for $19.95
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="T-Shirt^19.95^5">

Select a Color:
<SELECT NAME="eop_Shirt Color">
<OPTION VALUE="Red^00.00^0">Red
<OPTION VALUE="Blue^00.00^0">Blue
<OPTION VALUE="Green^00.00^0">Green
<OPTION VALUE="Black^00.00^0">Black
<OPTION VALUE="White^00.00^0">White
</SELECT>

Select a Size:
<SELECT NAME="eop_Shirt Size">
<OPTION VALUE="Small^00.00^0">Small
<OPTION VALUE="Medium^00.00^0">Medium
<OPTION VALUE="Large^00.00^0">Large
<OPTION VALUE="X-Large^00.00^0">X-Large
<OPTION VALUE="XX-Large^2.95^0">XX-Large -- Add $2.95
</SELECT>

Your Name printed on the front -- $5.95
<INPUT TYPE="TEXT" NAME="textadd_Your Name_5.95" SIZE="20"
MAXLENGTH="20">
Your Company Name printed on the back -- $10.95
<INPUT TYPE="TEXT" NAME="textadd_Company Name_10.95" SIZE="20"
MAXLENGTH="20">
<INPUT TYPE="SUBMIT" VALUE="Order!">
```

```
</FORM>
```

How Do I Create Up-Sell Items (Multiple Item Ordering)?

We may confuse you a little with this CFO. Remember we said that you could not add an item to the shopping cart without the cartitadd CFO? Well, we lied to you-- well, not exactly. *We did it only to stress the importance of the cartitadd CFO and to save a little confusion.*

The usage of the addit Custom CFO looks exactly like the cartitadd CFO:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="addit" VALUE="ACME Explosives^19.95^5">
<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
</FORM>
```

Although the addit CFO can add products to the shopping cart, it is not a Primary CFO since it does not communicate with Extension CFOs such as eop and quantity. The addit CFO completely ignores them- **but this is a good thing!**

You Can Use the addit CFO in Many Ways

Unlike the cartitadd CFO, you can add as many addit CFOs as you like. You can have a page of 20 or 30 checkboxes with one Submit button. Your shopper can then add all the checked items to their shopping cart at once.

For example, if you are selling a computer, you can include the option of adding a monitor and more RAM (or anything) simply by using the addit CFO:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

<INPUT TYPE="CHECKBOX" NAME="addit" VALUE="32 MB RAM^170.95^2">
32 MB RAM

<INPUT TYPE="CHECKBOX" NAME="addit" VALUE="17"
Monitor^300.00^50">
17" Monitor

etc...

<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">

</FORM>
```

Note: `"` is the HTML for a quotation mark.

You can build a standard form with the cartitadd CFO, include as many Extension CFOs as you like and use an addit CFO to add additional products.

The CartIt Commerce System 8.0 Site Design Guide

Extension CFOs will add options and capabilities to the cartitadd Primary CFO, but the addit CFO works alone and will not interfere with the Primary/Extension CFO relationship.

Here is an example:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

ORDER:
<SELECT NAME="cartitadd">
<OPTION VALUE="Pentium II 500^1200^40" SELECTED>Pentium II 500 --
$1200.00
<OPTION VALUE="Pentium II 750^3500^40">Pentium II 750 -- $3500.00
</SELECT>

<SELECT NAME="eop_Memory">
<OPTION VALUE="32 MB^00.00^0" SELECTED>32 MB</OPTION>
<OPTION VALUE="64 MB^89.95^0">64 MB -- Add $89.95</OPTION>
<OPTION VALUE="128 MB^189.00^0">128 MB -- Add $189.00</OPTION>
<OPTION VALUE="256 MB^384.00^0">256 MB -- Add $384.00</OPTION>
</SELECT>

<INPUT TYPE="CHECKBOX" NAME="addit" VALUE="17"
Monitor^300.00^50">
Check this box to order a 17" Monitor with it!

<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">

</FORM>
```

With this example, the shopper can order a Pentium system, select memory they want and optionally add a monitor to the cart at the same time. This is the power of addit. Why not just make the monitor an eop? Well, you want the shopper to be able to select the appropriate options they want, but you do not want to force them to buy a monitor.

addit with Quantities

addit is a Custom CFO so it does not accept Extension CFOs for quantities and/or options. For this reason, we created cartitqa (CartIt Quantity Add). This CGI is also a Custom CFO, however it is designed to work like addit except that it accepts quantities.

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

<INPUT TYPE="TEXT" NAME="cartitqa_ System Monitor _39.95_1"
VALUE="1">
System Monitor

<INPUT TYPE="TEXT" NAME="cartitqa_ Acme Explosives _29.95_1"
VALUE="1">
Acme Explosives

etc...

<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
```

How to Make a "View Your Shopping Cart" Link

Just create a link to CartIt; it can be a text link or button, it does not matter.

```
<A HREF="/cgi-bin/cartit.cgi">View Your Shopping Cart</A>
```

Of course, you can replace that text with an image.

Additional CFO Values for 4.x+ Generation CartIt

Up until now, you have seen that cartitadd has a value format of description^price^weight. In Version 4.x+ CartIt actually goes beyond this:

```
description^price^weight^taxable^smarttax
```

Taxable

Taxable can be set to either "1", "0" or can be left blank.

Leaving this value blank is the most common usage of this taxable option. If this item is left blank, CartIt will use the default setting you have configured in CommerceBuddy. For example, if in CommerceBuddy you have set the taxable option to "By Default Items Are Taxable," then by default CartIt will consider all items taxable. Thus, if you leave this option blank in your commerce page design, CartIt will consider the item a taxable item, i.e., the CommerceBuddy setting is the default value.

NOTE: This does not mean that CartIt will ALWAYS charge tax for the item, this only means that CartIt will charge tax if the shopper lives in the state or location where the tax applies, i.e., the item is a candidate for tax.

If you have this option set to "1" CartIt will consider this item a taxable item. If the shopper lives in one of the taxable regions, then this tax will take effect.

If you set this option to "0" this item will never be taxable.

Smart Tax

This option is rarely used. It is designed for special tax needs, for example, a cigar store. A cigar store may use taxable on all items such as humidors, pipes, etc., but may need to use Smart Tax for cigars, a tobacco product that has special tax considerations.

If a shopper purchased all the aforementioned items, then CartIt would charge the standard state sales tax to persons who live in the state, but with the Smart Tax you could charge 15% tax on the cigars only.

Getting a Better Understanding of It All

It is recommended that you create a few example pages based on the above material before getting into the following section. This section is designed to provide further explanation and hopefully will not confuse you too much.

Thinking CFO

The most important aspect of CartIt site design is that you can actually think in CFO; what does each CFO do, how does each CFO react to another, etc. The importance of this cannot be stressed enough. Far too often we receive support questions regarding eop and addit: "Why is addit ignoring my eop?" That is what addit was designed to do. It is a Custom CFO that works independently from the Primary (cartitadd) and Extension CFOs.

We have seen our user base do some very amazing things with CartIt. Those who understand CFOs and forms will design the most powerful and effective pages.

The CFO Army

Think of cartitadd and Extension CFOs as an army. They work together, the Primary CFO (cartitadd) is the general and the Extension CFOs are the soldiers. Without cartitadd, the Extension CFOs are lost and they have no purpose. The soldiers will not fight without the general. Again, Extension CFOs will work only with cartitadd.

System and Custom CFOs ignore the army and the army in turn ignores them.

Keep this in mind when designing pages:

1. CFOs have a purpose and type (or classification) such as Extension, Custom, etc.).
2. CFOs have specific names.
3. CFOs have specific values and the format must be adhered to.

FORM Elements are FORM Elements

Other than the obvious, what are we getting at?

Far too often, even advanced users will assume that a form element must be used as a checkbox or perhaps a select list. In our addit example, we used addit as a checkbox and one may make the mistake of assuming that this is required.

For example, you can very easily make eop a checkbox; if it is not checked, then it will be ignored:

```
<INPUT TYPE="CHECKBOX" NAME="eop_Monitor" VALUE="15&quot; Monitor -- $89.95">
Check this box for a 15&quot; Monitor -- $89.95
```

addit can easily be a select list:

```
<SELECT NAME="addit">
<OPTION VALUE="" SELECTED>No Thanks</OPTION>

<OPTION VALUE="15&quot; Monitor^49.95^0^^" SELECTED>15&quot; Monitor
-- 49.95</OPTION>

</SELECT>
```

We threw a curve ball at you with this one... the first option has no value, it is empty. The user sees "No Thanks" but the actual value of the form is empty. If it is selected, then nothing goes to CartIt.

It works like this:

Form *names* are very important to CartIt, form *values* are very important to CartIt, but it does not matter what form *elements* send this information. You are not sending a checkbox to CartIt, you are not sending a select list to CartIt; you are sending a *name* and a *value*, that is all. To look at it another way, when you have a drop-down menu (or select list), you have one name (let's use cartitadd) and you have many values. However, when you submit the button, the only thing that is sent to CartIt is the *name* cartitadd and the *value*, whatever item you had selected.

Keep in mind that CartIt is powerful and intelligent, but relies on you completely. CartIt is a program on the server and you are submitting information to the program in the form of a *name* and a *value*. CartIt has no idea that a Web page exists on the site. CartIt knows nothing. CartIt only sits on the server and waits for a link or a form post and when it gets it, CartIt executes and reads all of the *names* and *values* and performs the action that you have told it to:

```
cartitadd=description^price^weight
```

To CartIt This Means:

Add an item to the cart. The description, price and weight of the item follow. Also, this is cartitadd, so look for Extension CFOs.

Now keep in mind, no matter what form element sends this cartitadd, it is still going to go to CartIt in the same format: a *name* and a *value*.

What if your form contained and submitted the wrong name, let's say, "cartititem"? Well, this is not a CFO, so CartIt would execute and say, "Your shopping cart is empty." CartIt would not do anything else because no valid command was sent.

Debugging Pages

We will touch base on several of the most basic page design errors we see.

The Missing Form Container

The most common page design errors is an improper form design; the Web designer places the form elements on the page, but does not surround the form elements with the required `<FORM>... </FORM>` tags.

The following example will NOT work because although the Submit button exists, it does not have any direction. The button is just a button and it has no purpose without the form container.

Example of Missing Form Element:

```
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="ACME
Explosives^19.95^5">
<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
```

The Out of Place Form Element

It is also common to see a form that has the form container tags and the form elements, but the form elements are not inside of the form container. Below, you can see a properly designed form WITH THE EXCEPTION that the Submit button is not a part of the form. Therefore, when you click the Submit button, the form will not work.

Example of Elements not placed in Form:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="ACME
Explosives^19.95^5">

</FORM>

<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
```

Using Improper Page Design

When designing Web pages, it is important to note that in many versions of CartIt, certain characters should not be sent to the software. In particular, pricing such as \$1,250.00 can be displayed on the page to the shopper, however the VALUE of the form element should be written as 1250.00.

"\$" and "," are human readable characters that have no function to CartIt and in some verions of CartIt, instead of removing these invalid characters, CartIt will consider it to be an invalid number.

Example of Improper Value Usage:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">
```

```
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="ACME  
Explosives^$1250.00^5">
```

```
Buy ACME Explosives for $1,250.00
```

```
<BR>
```

```
<INPUT TYPE="SUBMIT" VALUE="Add Item To Shopping Cart?">
```

```
</FORM>
```

CartIt CFO Reference

cartitadd

```
Type.....: Primary
Usage.....: HIDDEN | SELECT | RADIO
Requirements..: NONE
VALUE FORMAT..: description^price^weight^taxable^smarttax
NAME FORMAT...: cartitadd
```

EXAMPLE:

```
<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="T-Shirt^19.95^3">
```

COMMENTS:

The cartitadd CFO is a replacement for the description/price CFO pair method of adding items to the shopping cart.

The description and price portion of the cartitadd CFO is required, but the weight portion may be omitted if you are not using a weight-based method of shipping.

quantity

```
Type.....: Extension
Usage.....: TEXT | SELECT | RADIO
Requirements..: cartitadd
VALUE FORMAT..: integer
NAME FORMAT...: quantity
```

EXAMPLE:

```
<INPUT TYPE="TEXT" NAME="quantity" VALUE="1" SIZE="3" MAXLENGTH="3">
```

COMMENTS:

The most common usage is as a TEXT input as shown above. Quantity is limited to 999. CartIt will check for valid user input. Any invalid input such as "AAA" or ".02" will be turned into "1".

eop

```
Type.....: Extension
Usage.....: SELECT | RADIO | CHECKBOX | HIDDEN
Requirements..: cartitadd
VALUE FORMAT..: description^price^weight
NAME FORMAT...: "eop_what the option is for"
```

EXAMPLE:

```
<SELECT NAME="eop_Shirt Size">  
<OPTION VALUE="Small^0^0">Small  
<OPTION VALUE="Medium^0^0">Medium  
<OPTION VALUE="Large^0^0">Large  
<OPTION VALUE="X-Large^2.95^1">X-Large add $2.95  
</SELECT>
```

COMMENTS:

The eop CFO will add options to a Primary CFO. It is best used in a SELECT list form element. You can have unlimited eop CFOs in a form. They can also increase the price and weight of the Primary CFO.

The description and price portion of the cartitadd CFO is required, but the weight portion may be omitted if you are not using a weight-based method of shipping.

CHECKBOX NOTES:

If an eop CFO is used as a CHECKBOX, keep in mind that checkboxes do not require a selection. If the option is required such as a "T-Shirt Size," it is recommended that you use a RADIO group or SELECT list.

textadd

```
Type.....: Extension  
Usage.....: TEXT  
Requirements..: cartitadd  
VALUE FORMAT..: User Input limit 60 characters  
NAME FORMAT...: textadd_what the text is for_price
```

EXAMPLE:

```
<INPUT TYPE="TEXT" NAME="textadd_Custom Printing_9.95" SIZE="40"  
MAXLENGTH="40">
```

COMMENTS:

The textadd CFO allows your shopper to enter text for personalization options. To limit the amount of text the user can input, use the MAXLENGTH value.

The textadd CFO should not be used with a TEXTAREA element. The textadd CFO can have up to 60 characters. If no text is entered, CartIt will ignore the option.

Form elements (CFOs) are usually sent to the CGI (CartIt) in the order they appear in the form.

addit

```
Type.....: Custom
Usage.....: SELECT | CHECKBOX | RADIO | HIDDEN
Requirements..: NONE
VALUE FORMAT..: description^price^weight^taxable^smarntax
NAME FORMAT...: addit
```

EXAMPLE:

```
<INPUT TYPE="CHECKBOX" NAME="addit" VALUE="Pentium
400^2000.00^40">Order a Pentium 400 for $2,000.00
```

TYPE INFORMATION:

This Custom CFO adds items to the shopping cart independently of all other CFOs. It can be used alone to add items to the shopping cart (like a Primary CFO) or it can be used like an Extension CFO along with Primary CFOs to add additional products to the cart at the same time.

Unlike Extension CFOs, addit will not add options or capabilities to an item. Instead, it will always add an independent product to the cart.

COMMENTS:

The addit CFO can be used in a list of checkboxes, allowing your shopper to select the items they wish to purchase and submit the form to CartIt. All of the checked items will be added to the cart at once.

The addit CFO is usually used in one of 2 ways:

1. If you are selling an automobile, you can also sell paint sealant with an addit CFO. You could use the eop CFO to make the paint sealant as part of the automobile, but the addit CFO will add the paint sealant as a separate product, allowing your shopper to remove it.
2. If you are selling camping tents, you could use the standard cartitadd Primary CFO for the tent and the shopper could select the tent color with an eop CFO. Just under these options, you could have "Add a matching sleeping bag to your order!" as an addit CFO with a checkbox. The shopper could then add a tent to the shopping cart and optionally add a sleeping bag as a separate product.

HIDDEN FEATURE:

A previously undocumented feature of addit is that you can actually name your addit element addit?, where "?" is any character. This is done because if you want to have more than one radio 'group' of addits, you would not be able to because all groups would have to be named addit. For this reason, we allow addit to be named addit0, addit1, addit2, etc. You can name your first radio 'group' (group of radios) addit and have additional groups named addit1, addit2, etc.

cartitqa

```
Class.....: Custom
Usage.....: TEXT
Requirements.....: NONE
V3 VALUE FORMAT.....: NOT SUPPORTED
V4 VALUE FORMAT.....: Integer
NAME FORMAT.....:
cartitqa_description_price_weight_taxable_smarttax
```

EXAMPLE:

```
<INPUT TYPE="TEXT" NAME="cartitqa_Microsoft Money_95.00_3" SIZE="3"
MAXLENGTH="3">
```

Although the cartitqa CFO can add products to the shopping cart, it is not a Primary CFO since it does not communicate with Extension CFOs like eop and quantity. The cartitqa CFO completely ignores them -- but this is a good thing!

Unlike the cartitadd CFO, many cartitqa CFOs may be added to the shopping cart at once. You can have a page of 30 text inputs that allow your shopper to specify a quantity for each item they want to purchase with one click of the Submit button. Your shopper can add all of the items with a >0 quantity to their shopping cart at once.

cartit_dynamic_file

```
Class.....: System
Usage.....: HIDDEN
Requirements.....: Any add item CFO (cartitadd, addit, cartitqa)
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: cartit_dynamic_file
Value Format.....: filename.html
```

Example:

```
<INPUT TYPE="HIDDEN" NAME="cartit_dynamic_file"
VALUE="printer_accessories.html">
```

This CFO allows you to display a dynamic header at the top of the shopping cart. This header will be placed above the shopping cart contents and below the custom header allowed by CartIt.

The purpose of this CFO is to allow a file with one or more CartIt product interfaces in it.

Example:

You are selling computer supplies and you want to sell printer accessories with all of your printers. So, you create a file called printer_accessories.html and put it in the data directory. You then add the line:

```
<INPUT TYPE="HIDDEN" NAME="cartit_dynamic_file"
VALUE="printer_accessories.html">
```

to all of your printer related product pages. Now every time someone adds a printer product to the shopping cart, they will be prompted with the content of this file above the cart.

CartIt is no longer compatible with the "recommend" CFO; we are using this method in its place. The problem with "recommend" is that if you recommended a product on 10 pages and the price changes for this product, you had to change 10 pages. This method allows you to change only one file, while putting anything in the file like images, links, etc. You can even delete the file and CartIt will ignore the command.

WARNING!

In this file, if you use HTML forms you must remember to ALWAYS close the form. If you fail to close the form you may cause CartIt to malfunction. If your form does not close, it will carry over into CartIt's form.

cartit_set_cookie

```
Class.....: System
Usage.....: HIDDEN
Requirements....: NONE
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: cartit_set_cookie
Value Format.....: name^value^expin^exptype^action^actionvalue
```

Why use it?

You can use this feature for setting any type of cookie, but we implemented it so you can take advantage of two features directly built into CartIt:

The cartit_item_requires_cookie CFO and the cartit_discount system.

Example:

```
<INPUT TYPE="HIDDEN" NAME="cartit_set_cookie" VALUE="Cookie
Name^Cookie Value^1^days^redirect^http://www.site.com/cool.html">

name^value^expin^exptype^action^actionvalue
```

name:

Name of the cookie

value:

Value of the cookie

expin:

The cookie will expire in (directly related to exptype)

exptype:

The method to calculate expin. Possible values are hours, days, months, years

action:

Can be "printfile" or "redirect" (directly related to actionvalue)

actionvalue:

If action is "redirect" a fully qualified URL must be used. If action is "printfile" it must be a file name located in the data directory. The file name may not contain any special characters and must have a .html file extension.

Additional information and examples are available in the cartit_item_requires_cookie CFO and the cartit_discount system sections.

cartit_item_requires_cookie

```
Class.....: System
Usage.....: HIDDEN
Requirements....: Any add item CFO (cartitadd, addit, cartitqa)
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: cartit_item_requires_cookie
Value Format.....: name^value^faildoc
```

This CFO allows you to require the shopper have a cookie for a product or they cannot add that product to their shopping cart.

Example:

```
<INPUT TYPE="HIDDEN" NAME="cartit_item_requires_cookie"
VALUE="ctype^VIP^vip.html">
```

Above, if the shopper does not have a cookie with the NAME "ctype" and a value of "VIP" the file vip.html stored in your data directory will be displayed. If they DO have the cookie, the item will be added to their shopping cart as normal.

Why use it?

You may want to have specific items offered only to wholesale customers, VIP customers and so on. You may be able to think of a few other examples in which you would want only certain people to have access to an item.

How does a customer get this cookie?

You can use any software you like, but CartIt does have a built in CFO called cartit_set_cookie.

To set a cookie for the above, you would perhaps send a valued customer to a Web page that informs them they have become a VIP customer and they can now purchase VIP items!

```
<INPUT TYPE="HIDDEN" NAME="cartit_set_cookie"  
VALUE="ctype^VIP^10^days^redirect^http://www.site.com/cool.html">
```

cartit_discount

cartit_discount is a cookie name that must exist to give a shopper a discount based on cookies. This section explains how to set this cookie and the reason it exists. You can use any program, CGI or otherwise to set this cookie or you can use CartIt's cookie setting CFOs.

CartIt offers several types of discounts. All account discounts and coupon discounts are set in CommerceBuddy. CartIt also offers a cookie discount that requires the cartit_discount cookie to exist. This feature was designed to accommodate Web Sites that deal in resale and wholesale merchandise, allowing a specific discount to be given to a specific group of customers. The cartit_discount cookie must have the following format:

```
cartit_discount=Discount Description::Discount Percentage
```

Example:

```
cartit_discount=Wholesale::0.30
```

The above would give everyone with this cookie a 30% discount and display the text "Total Wholesale Savings: \$??.???" in the shopping cart.

To set a cookie for wholesale customers (or other VIP customers) follow these steps:

Create a Web page explaining to the shopper that you are about to set a cookie identification that gives them an x% discount on all items. You can put the page anywhere on your Web Site and password protect it if desired.

On this page, put the following:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">  
  
<INPUT TYPE="HIDDEN" NAME="cartit_set_cookie"  
VALUE="cartit_discount^Wholesale::0.30^1^months^redirect^http://www.  
site.com/YouAreWholesale.html">  
  
<INPUT TYPE="SUBMIT" VALUE="Get a Cookie">  
  
</FORM>
```

Then, create YouAreWholesale.html (or any name you want to use) and place it on your Web Site-- note the action value above.

bounce

```
Class.....: System
Usage.....: HIDDEN | CHECKBOX
Requirements....: Any add item CFO (cartitadd, addit, cartitqa)
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: NOT SUPPORTED!
V5 Name Format...: bounce
Value Format.....: 1
```

bounce will tell CartIt to bounce back to the page from which the item was added to the cart.

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="Acme Product^19.95^5">

<INPUT TYPE="HIDDEN" NAME="bounce" VALUE="1">

<INPUT TYPE="SUBMIT" VALUE="Order">

</FORM>
```

NOTE: In `cartitcf.cgi` you can configure CartIt to bounce automatically by setting `$CARTIT_BOUNCE=1`; (where 1 is true or on). Note that if no item is added to the cart, in the case that a person may select 0 as a quantity, then CartIt will show the normal CartIt page; this is also true if someone clicks the checkout links.

bounceto

```
Class.....: System
Usage.....: HIDDEN | RADIO | SELECT
Requirements....: Any add item CFO (cartitadd, addit, cartitqa)
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: NOT SUPPORTED!
V5 Name Format...: bounce
Value Format.....: 1
```

This allows you to enter a fully qualified URL to bounce to.

Example:

```
<FORM ACTION="/cgi-bin/cartit.cgi" METHOD="POST">

<INPUT TYPE="HIDDEN" NAME="cartitadd" VALUE="Acme Product^19.95^5">

<INPUT TYPE="HIDDEN" NAME="bounce" VALUE="1">
```

The CartIt Commerce System 8.0 Site Design Guide

```
<INPUT TYPE="HIDDEN" NAME="bounceto"
VALUE="http://www.yoursite.com/product2.html">

<INPUT TYPE="SUBMIT" VALUE="Order">

</FORM>
```

Keep in mind that with this option you can use Radio buttons.

Example:

Tell Me About :

```
<INPUT TYPE="RADIO" NAME="bounceto"
VALUE="http://www.yoursite.com/">Home Page<BR>

<INPUT TYPE="RADIO" NAME="bounceto"
VALUE="http://www.yoursite.com/phones.html">Sales on Phones<BR>

<INPUT TYPE="RADIO" NAME="bounceto" VALUE=" ">Stay on this page<BR>
```

This above will bounce the shopper to the option they select; the phone.html page, the home page, or bounce will be ignored since the final option is blank.

dynamic_cartit

```
Class.....: System
Usage.....: HIDDEN | RADIO | SELECT
Requirements....: Any add item CFO (cartitadd, addit, cartitqa)
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: NOT SUPPORTED!
V5 Name Format...: dynamic_cartit
Value Format.....: dynamic_filename.html
```

This CFO allows you to display a simple shopping cart page to the shopper. It can be any page really, but a simple shopping cart is most common.

Example:

```
<INPUT TYPE="HIDDEN"
NAME="dynamic_cartit"
VALUE="anyfilename.html">
```

It is recommend that your file name start with "dynamic_".

Included with CartIt 5.0 will be several files in your data folder that you should refer to for further examples.

```
/data/dynamic_cartit_frame.html
/data/dynamic_cartit_ref.html
```

In cartitcf.cgi you have the option of setting the variable:

The CartIt Commerce System 8.0 Site Design Guide

```
$CARTIT_DISPLAY = 'dynamic_filename.html';
```

This variable encapsulates all of the functions on `dynamic_cartit`. In essence, setting this variable to a file name will force the cart to always be in `dynamic_cartit` mode.

Please note that CartIt will not display the `dynamic_cartit` page if no items are added to the cart. This allows "View Shopping Cart" links to continue to function, as the only time no item is added to the cart is when the user clicks this link.

HIDDEN FEATURES:

FRAMES SUPPORT: In frames-based pages, if the dynamic file name contains the word "frame" then the frame file will always show and the standard CartIt shopping cart will NEVER show. This is done to prevent the full shopping cart from loading in a small frame window.

You should never hard code a frame file in `cartitcf.cgi`:

```
$CARTIT_DISPLAY = 'dynamic_framefilename.html';
```

This will cause CartIt to NEVER show.

YOU CAN BOUNCE!

Sometimes you want to allow the shopper to bounce back to the same page with this option. If you want them to bounce, you can also set your file name to 'bounce':

```
<INPUT TYPE="HIDDEN"  
NAME="dynamic_cartit"  
VALUE="bounce">
```

This will function the same as 'bounce'.

cartitqd

```
Class.....: Extension  
Usage.....: HIDDEN  
Requirements....: cartitadd  
V3 Name Format...: NOT SUPPORTED!  
V4 Name Format...: NOT SUPPORTED!  
V5 Name Format...: cartitqd  
Value Format.....: from:to:discount [to +]
```

Example:

```
<INPUT TYPE="HIDDEN"  
NAME="cartitqd"  
VALUE="2:5:10#6:10:20#11:+:25">
```

The above example shows the first directive is 2-5 items at a 10% discount, the next directive is 6-10 items at a 20% discount and the final directive is 11+ at a 25% discount. The # separates the directives. The + is a special directive that says "or greater".

cartitqdc

```
Class.....: Custom
Usage.....: HIDDEN
Requirements....: addit or cartitqa
V3 Name Format...: NOT SUPPORTED!
V4 Name Format...: NOT SUPPORTED!
V5 Name Format...: cartitqd
Value Format.....: from:to:discount [to +]
```

Same as cartitqa except it works with addit and cartitqa only.