# DOCUMENTATION



# **Product Options with eCart**

Online stores sell a wide variety of products, and many of those products have options within themselve example, you may be selling shoes, and want the customer to be able to choose the size shoe that they reslightly more complex example would be if you were selling T-Shirts, and needed to charge extra for the purchase X-Large due to the additional fabric.

This tutorial shows you how you can incorporate product options with eCart's shopping cart functionalit including the ability to increment the price for certain options.

**NOTE:** We have provided our <u>Free eCommerce MySQL database</u> to alleviate the need for creating your own custom database tables. It is recommended you use this database to avoid having to make modifications to the recordset provided to you in this tutorial.

#### What you need to start:

- Dreamweaver CS3 or CS4
- Dreamweaver site with eCart checkout functionality already created.
- eCart 5
- WebAssist Free eCommerce MySQL database (Optional)
  - **NOTE:** This tutorial refers to the <u>Free eCommerce MySQL database</u> that is provided by WebAssist. This database contains all the necessary tables and columns to complete this tutorial. If you are not using this database, you will need to review the table and column references to make sure they match up with your database structure.

Before you proceed with adding the product option functionality to eCart, you will need to populate your product catalog, and add the v and option groups to your database.

The <u>Free eCommerce MySQL database</u> contains the necessary database tables for handling these product options and it is recommende database to add product options to your online store. Using your preferred database management tool, you should populate the necessar products and their options. Then continue with this tutorial to add this functionality to your shopping cart.

Refer to the Database Relationship Overview to learn more about how the tables are configured for product options.

In this tutorial we will be creating two types of product options:

- 1. Color options without a price increment.
- 2. Size options with a price increment for certain sizes.

#### 1. Update your eCart Object

The first step you need to take is to update your eCart Object with the necessary columns for handling your product options.

#### **Add New Columns**

- 1. From the eCart Object panel, select your cart and click Edit.
- 2. Choose the Columns tab of your eCart object.

art Object							2
GENERAL	COLUMNS	CALCULATIONS	DISCOUNTS	CHARGES	TAX	SHIPPING	
Create nev	v or edit exis	ting columns					

- 3. Add a new column called **color**.
- 4. From the menu list, choose text for the column.
- 5. Select the unique checkbox.
- 6. Add a new column called size.
- 7. From the menu list, choose text for the column.
- 8. Select the unique checkbox.
- 9. Add a new column called sizePriceInc.
- 10. From the menu list, choose **currency** for the column.
- 11. Select the unique checkbox.

color	Text v	=	I
size	Text V		
sizePriceInc	Currency √ 😽	-	
Name:	Output Type: Unique:		
sizePriceInc	Currency 🔻 🔽		

#### **Adding new Calculations**

1. Choose the Calculations tab.

GENERAL COLUMNS CALCULATIONS DISCOUNTS CHARGES TAX SHIPPING

- 2. Add a new calculation called truePrice.
- 3. For the truePrice calculation, enter the following: [Price] + [sizePriceInc]
- 4. With this new calculation, we will need to update the TotalPrice calculation.
- 5. Modify the TotalPrice calculation to look like this:
  - ([Price] + [sizePriceInc]) \* [Quantity]

Name	Formula		Туре
TotalWeight	[Weight] * [Quantity]		Weight
TotalPrice	([Price] + [sizePriceInc]) * [Quantity]		Currency
FullDetails	[Quantity] . ' ' . [Name] . ' (' . [ID] . ')'		Text
truePrice	[Price] + [colorPriceInc] + [sizePriceInc	]	Currency
Name:	Outp	it Type:	
TotalPrice	Curr	ency	•
Calculation form	la:		+ Add to formula
([Price] + [sizeF	riceInc]) * [Quantity]		

6. Click OK to save your changes and close the eCart Object window.

# 2. Create your Shopping Cart

Now that you have made changes to your eCart Object, you will need to update the shopping cart page to contain the new columns. This remove the Shopping Cart instance from your cart page, and insert a new one using the eCart Display Manager.

- 1. Open your shopping cart page. If you don't already have one, create a new PHP page in your site called cart.php.
- 2. Place your cursor where you wish your new shopping cart display to be inserted.
- 3. From the Insert menu, choose WebAssist > eCart > Display Manager.
- 4. Recreate your eCart Shopping cart. If you have never created an eCart Shopping Cart Display, see the <u>Getting Started Guide</u> for more inform

## 3. Adding Product Options

Now that you have updated your eCart Object and recreated your Shopping Cart Display, you can update your Add to cart behaviors to handle the product options.

- 1. Open your product detail page. This page is typically created by DataAssist that displays the details for a specific item and the Add to Cart button for that item.
- 2. Open the Server Behaviors panel from Window > Server Behaviors.
- 3. From the Server Behaviors panel, choose Add > Recordset.



**NOTE:** When the Recordset dialog box displays, it may open in Simple Mode. For these instructions, we will need to change this to Advanced Mode. If present, choose the Advanced... button from the buttons on the right.

- 4. In the Name field, enter rsColor.
- 5. Highlight the following code, and press Ctrl (Cmd) + C to copy this content. SELECT productoptions.OptionPriceincrement, productoptions.ProductOptionID, options.OptionName, options.OptionGroupID, optiongroups.OptionGroupName FROM options INNER JOIN productoptions ON options.OptionID = productoptions.ProductOptionID INNER JOIN optiongroups ON options.OptionGroupID = optiongroups.OptionGroupID WHERE optiongroups.OptionGroupName = 'color' AND productoptions.ProductID = paramItem
  6. Place your current in the SOL here of the Pacerdent dialog and press Ctrl (Cmd) + V to pasta this content.
- 6. Place your cursor in the SQL box of the Recordset dialog and press Ctrl (Cmd) + V to paste this content.
- 7. Under the Variables section, click Add.
- 8. In the Name field of the Variables dialog, enter paramItem.
- 9. From the menu list choose Integer.
- 10. In the Value field, enter -1.
- 11. Copy and paste the following parameter in the \$row\_WADAproducts['ProductID']

Name:	rsColor		ОК
Connection:	wadb	▼ Define	Cancel
SQL:	SELECT productoptions.Optio options.OptionName, options FROM options INNER JOIN p productoptions.ProductOptio WHERE options.OptionGroup	onPriceincrement, productoptions.ProductOptionID, s.OptionGroupID  roductoptions ON options.OptionID = onID DID = 1 AND productoptions.ProductID = paramItem	Test Simple Help
Variables:	•-		
	paramItem	Name: paramItem Type: Integer Default value: -1 Run-time Value: \$row_rsProducts['Pr	

- 12. Click **OK**.
- 13. From the Server Behaviors panel, choose Add > Recordset to create another recordset. When the New Recordset dialog displays, make sure you are in Advanced Mode.
- 14. In the Name field, enter rsSize.
- 15. Highlight the following code, and press Ctrl (Cmd) + C to copy this content. SELECT productoptions.OptionPriceincrement, productoptions.ProductOptionID, options.OptionName, options.OptionGroupID, optiongroups.OptionGroupName FROM options INNER JOIN productoptions ON options.OptionID = productoptions.ProductOptionID INNER JOIN optiongroups ON options.OptionGroupID = optiongroups.OptionGroupID WHERE optiongroups.OptionGroupName = 'size' AND productoptions.ProductID = paramItem
- 16. Place your cursor in the SQL box of the Recordset dialog and press Ctrl (Cmd) + V to paste this content.
- 17. Under the Variables section, click Add.
- 18. In the Name field of the Variables dialog, enter paramItem.
- 19. From the menu list choose **Integer**.
- 20. In the Value field, enter -1.
- 21. Copy and paste the following parameter in the \$row\_WADAproducts['ProductID']

Name:	rsSize		ОК
Connection:	wadb	▼ Define	Cancel
SQL:	SELECT productoptions.Option options.OptionName, options. FROM options INNER JOIN pro productoptions.ProductOption WHERE options.OptionGroupI	Priceincrement, productoptions.ProductOptionID, OptionGroupID oductoptions ON options.OptionID = NID D = 2 AND productoptions.ProductID = paramItem	Simple
Variables:	<b>+</b> -		
	paramItem	Name: paramItem Type: Integer Default value: -1 Run-time Value: \$row_rsProducts["Pr	

#### 4. Create Add to Cart button

The next step is to create your Add to Cart buttons. If you already have these buttons added to this page, you can reenter the interface to modify them. Otherwise, you can create new ones.

- 1. Place your cursor where you wish to insert your Add to Cart button. If you already have an Add to Cart button on this page, you can launch the Add to Cart button interface by double-clicking the eCart Add from Recordset option in the Server Behaviors panel.
- 2. From the Insert menu, choose WebAssist > e Cart > Add to Cart.
- 3. From the General tab, choose Look up item ID from recordset.
- 4. Choose the DataAssist recordset, WADAProducts.

This recordset was created by DataAssist and displays all the products in the products database table. If you are not using a DataAssist created page, you will want to create this recordset, and then reenter the Add to Cart interface.

eCart Add To	Cart Button
GENERAL	BINDINGS BUTTON
Properties	Cart: eCart1   Add static item to cart
	<ul> <li>Look up item ID from recordset</li> <li>Recordset: WADAProducts</li> <li>ID Column: ProductID</li> </ul>

5. Choose the Bindings tab.

- 6. Select the Color column.
- 7. Select the Updatable checkbox.
- 8. From the menu list, choose Select List.

Price	php echo \$row_WADAProducts none</th
color	php echo \$row_rsColorLookUp[ select</td
colorPriceInc	php echo \$row_rsColorLookUp[ none</td
size	php echo \$row_rsSizeLookUp[" none</td
Default value:	<pre>\$row_rsColorLookUp['OptionName']; ?&gt;</pre>
Updatable:	👽 as: select list 🔻

Click OK to create this add to cart button.
 We will be modifying this add to cart button to handle more information later.

5. Size Lookup Recordset

The next step is to create the recordset that handles the size price increment.

- 1. From the Server Behaviors panel, choose Add > Recordset.
- 2. Make sure you are in Advanced Mode.
- 3. In the Name field, enter rsSizeLookup.
- 4. Copy and paste the following SQL query into the SQL text field. SELECT productoptions.OptionPriceincrement, productoptions.ProductOptionID, options.OptionName, options.OptionGroupID, optiongroups.OptionGroupName FROM options INNER JOIN productoptions ON options.OptionID = productoptions.ProductOptionID INNER JOIN optiongroups ON options.OptionGroupID = optiongroups.OptionGroupID WHERE optiongroups.OptionGroupName = 'size' AND productoptions.ProductOptionID = paramOption AND productoptions.ProductID = paramItem

JUSEL			
Name:	rsSizeLookUp		ОК
Connection:	wadb	▼ Define	Cancel
SQL:	SELECT productoptions.Op options.OptionName, optio FROM options INNER JOIN productoptions.ProductOp WHERE options.OptionGro	otionPriceincrement, productoptions.ProductOptionID, ons.OptionGroupID I productoptions ON options.OptionID = tionID upID = 2 AND productoptions.ProductOptionID =	Simple Help
Variables:	+-	Name: paramOption	
	paramItem	Type: Integer Default value: -1 Run-time Value: \$ POSTFeCart1_1	

- 5. Under the Variables section, click Add.
- 6. In the Name field of the Variables dialog, enter paramOption.
- 7. From the menu list choose Integer.
- 8. In the Value field, enter -1.
- 9. Copy and paste the following parameter in the **Runtime Value** field. \$\_POST['eCart1\_1\_size\_Add']

Edit Variable		<b>—</b> ×-
Name: Type: Default value:	paramOption Integer  ▼	OK Cancel Help
Runtime value:	<pre>\$_POST['eCart1_1_size_Add']</pre>	

- 10. Under the Variables section, click Add.
- 11. In the Name field of the Variables dialog, enter paramItem.
- 12. From the menu list choose Integer.
- 13. In the Value field, enter -1.
- 14. Copy and paste the following parameter in the \$\_POST['eCart1\_1\_ID\_Add']

dit Variable			×
Name:	paramltem		ОК
Type:	Integer	•	Cancel
Default value:	-1		Help
Runtime value:	\$_POST['eCart1_1_ID_Add']		

**NOTE:** In case your select list name differs, make sure that the name entered in the parameter above matches the name you copied earlier.

#### 6. Populate the Select Lists

The Color select list was automatically created for you when you created your eCart Add to Cart button. However, we are going to manually create the Size select list since this list will include a price increment for certain options.

#### **Size List**

- 1. Place your cursor where you wish the size select list to display.
- 2. From the Insert menu, choose Form > List/Menu.
- 3. In the name field, enter eCart1\_1\_size\_Add.

List/Menu	Type 💿 Menu	Height	List Values	Class
Cart1_1_size	🔿 List	Selections Allow multiple	Ø Dynamic	]

**NOTE:** This name is already being used in the recordset you created above. So, in order to ensure this works, you will need to enter the name exactly as it is displayed.

- 4. Select the Size Select List you just created.
- 5. From the Properties Inspector, choose the Dynamic button.

	Type 🔘 Menu	Height		List Values	Class	None	•
_size	🔘 List	Selections	Allow multiple	Ø Dynamic	Ì		
	Initially selected		Select A Size	Bind to Dynamic Sour		rce	

- 6. Click Add for static options.
- 7. Press tab to leave the Value field blank.

- 8. In the Label field, enter Select A Size.
- 9. From the Options from recordset field, choose rsSize.
- 10. From the Value field, choose **ProductOptionID**.
- 11. From the Labels field, choose **OptionName**.

Menu:	eCart1_1_size	Cart1 🔻		ОК	
Static options:	•-			0	Cancel
	Value	Label			Help
		Select A Si	ize		
)ptions from recordset:	rsSize		•		
Values:	ProductOptionIE	• •]			
Labels:	OptionName	•			
Select value equal to:			Ŷ	2	

#### **Color List**

- 1. Select the Color select list that was created when you added the eCart Add to Cart button.
- 2. From the Properties Inspector, choose the Dynamic button.
- 3. Click Add for static options.
- 4. Press tab to leave the Value field blank.
- 5. In the Label field, enter Select A Color.
- 6. From the Options from recordset field, choose rsColor.
- 7. From the Value field, choose **ProductOptionID**.
- 8. From the Labels field, choose **OptionName**.

Menu:	"eCart1_1_cold	or_Add2 <sup>*</sup> in form	n "eCai 🐨	OK
Static options:	÷			Cancel
	Value	Label		Help
		Select A (	Color	
ptions from recordset:	rsColor		•	
Values:	ProductOption	ID 🔻		
Labels:	OptionName	•		
Select value equal to:			Í	

# 7. Update the Add to Cart

Now that we have created and populated the lists for this product's options, we need to update the Add to Cart button.

- 1. From the Server Behaviors panel, double-click the eCart Add From Recordset behavior. *This will launch the Add to Cart interface for modifying the behavior.*
- 2. Select the **Bindings** tab.
- 3. Select the size column.
- 4. Choose the value lightning bolt.
- 5. Expand the rsLookup Recordset.
- 6. Choose OptionName.



- 7. Click OK.
- 8. Select the sizePriceInc.
- 9. Choose the value lightning bolt.

- 10. Expand the rsLookup Recordset.
- 11. Select OptionPriceIncrement.

Dynamic Data		
Field:	Recordset (rsProducts) Recordset (rsSizeLookUp) OptionPriceincrement ProductOptionID OptionName OptionGroupID Recordset (rsColorLookUp)	OK Cancel Help

### What to do next...

Now that you have completed creating your Download Center, you may want to spend some time customizing the appearance of the page, and adding additional Recordset bindings to display information from the orders or products tables. When you are finished with your site, you will want to upload everything to your live server and import your database to your live MySQL database so that your online store is on the web.

Last Updated: April 14, 2010