

### **Introduction**

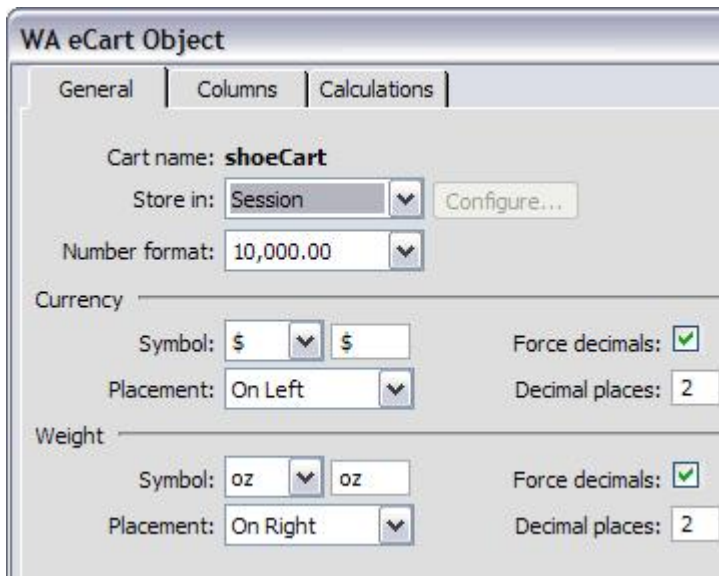
One feature that WA eCart users have at their disposal is the ability to format the way currency is displayed in their WA eCart applications. This TechNote shows how to format your currency display in your eCart application using the built-in functions in WA eCart.

### **General Overview**

The WA eCart Object allows the developer to define several aspects related to currency: the currency symbol, its placement, the number of decimal places, even the number format. Typically the currency settings are shown in the display generated by the WA eCart Display Manager. Often, there will be places in the application where formatted currency needs to be displayed outside of the WA eCart display or where a custom currency value needs to be added to the display. In addition to the controls available through the WA eCart Object, you can use a custom function built-in to WA eCart to properly display currency values.

### **Defining how currency is to be displayed**

The WA eCart Object Server Behavior's General tab is where you define the currency display symbol, placement, whether to force decimals and how many decimal places are to be displayed:

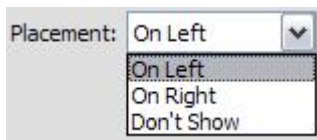


The available symbols for currency display are Dollars, Euros, British Pounds Sterling, Francs and Yen.

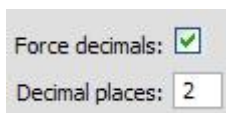
Other currency symbols such as the Swedish Krona (kr) can be added manually in the Symbol text field to the right of the list:



The Placement option allows you to format the symbol display to appear to the left or right of the currency value, or not at all:



If selected, the Force decimals option causes whole number values to appear with the trailing decimal to the set number of decimal places:



With the above settings, if you have a calculation that returns a 5, the value will be displayed as "5.00"; if you have a value that has more than two decimal places (usually from a tax calculation), it will be displayed with only two decimal places. If you examine the generated code, you'll see that the custom WA\_eCart\_FormatNumber() function performs the rounding operation in these situations.

The same tab in the WA eCart Object allows you to change how numbers, including currency values, are formulated. Use the Number Format list to choose between comma-separated thousands with period-separated decimals (i.e., \$10,000.00) and period-separated thousands with comma-separated decimals (i.e., \$10.000,00).

### How to apply currency formatting to values in the eCart application

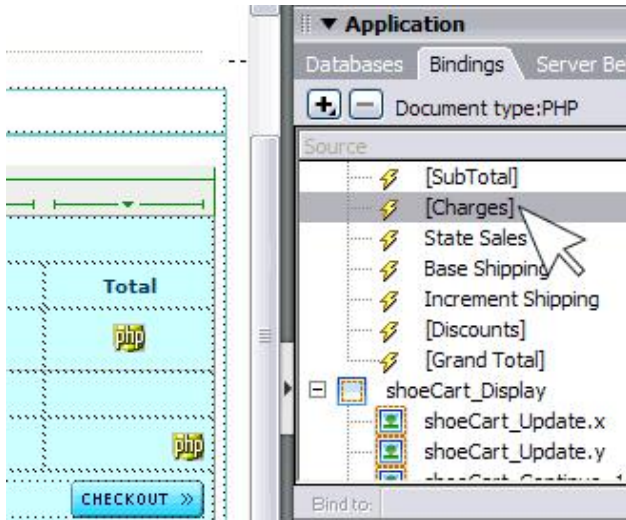
As mentioned above, the WA eCart Display Manager automatically inserts formatted currency values on the page:

Price	Quantity	Delete	Total
\$101.80	<input type="text" value="1"/>	<input type="checkbox"/>	\$101.80
\$89.95	<input type="text" value="1"/>	<input type="checkbox"/>	\$89.95
			<b>\$191.75</b>

You may need to display formatted currencies in other places in your eCart application, or you may need to display custom charges and discounts within the WA eCart display. In these situations, you can bind the value to the page and then wrap it in the `WA_eCart_DisplayMoney()` function to enable the currency to display properly. Here's how to do it:

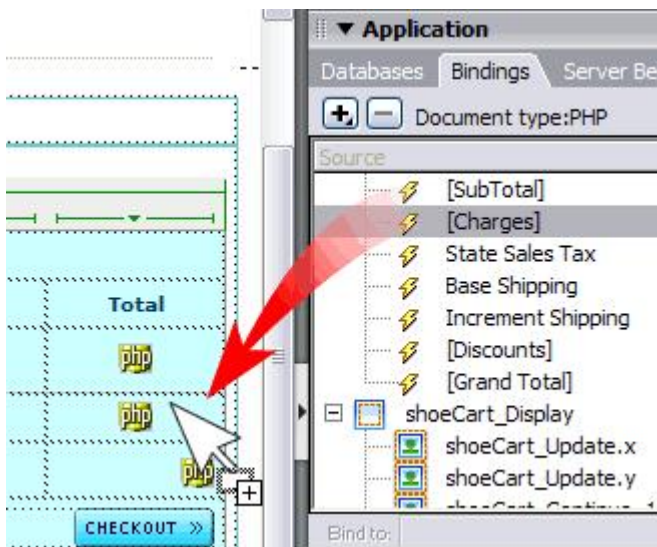
#### Step 1: Bind the value to the page

The Dreamweaver Bindings panel makes it easy to add any value to your Web page. Click the (+) button (PC) or right-pointing triangle (Mac) to the left of any object in the Bindings panel to expand the object and reveal the elements you can bind.



Elements can include recordset fields, Session Variables and eCart Object values. You can even bind form POST elements that have been added to the Bindings panel using the WA Form Data binding and WA Validated Entries from WA Validation Toolkit.

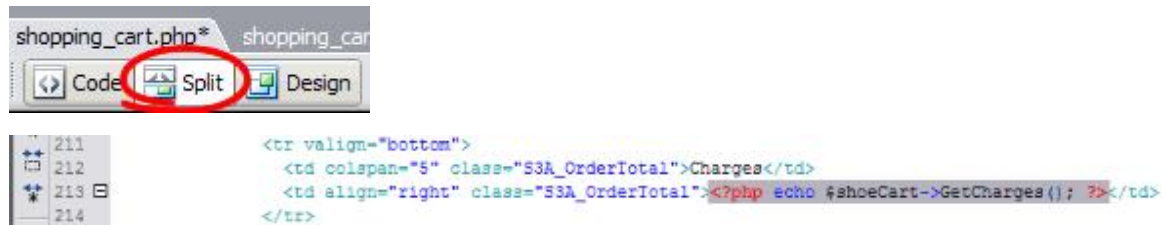
Next, drag the desired element onto the page in the location where you wish the value to appear on the page.



After binding the value to the page, you can add formatting to display the desired currency settings.

## Step 2: Format the value to WA eCart currency settings

After you've placed the binding on the page, switch to Split view in Dreamweaver and locate the line of code where the new value is located:



Here's how the code looks when you bind the WA eCart [Charges] value in each of the supported Server Languages:

[ASP JScript]

```
<%=cartName.GetCharges() %>
```

[ASP VBScript]

```
<%=cartName_GetCharges() %>
```

[ColdFusion]

```
<cfoutput>#cartName_GetCharges() #</cfoutput>
```

[PHP]

```
<?php echo $cartName->GetCharges(); ?>
```

To display the resulting value as formatted currency, apply the `WA_eCart_DisplayMoney()` function with the value as a parameter of the function. Here is how the function is initialized with the parameters it uses in the `WA_eCart_Definition` file for each server language:

[ASP JScript]

```
function WA_eCart_DisplayMoney(theObject, theAmount)
```

[ASP VBScript]

```
function WA_eCart_DisplayMoney(theObject, theAmount)
```

[ColdFusion]

```
function WA_eCart_DisplayMoney(theCart, theAmount)
```

[PHP]

```
function WA_eCart_DisplayMoney($theObject, $theAmount)
```

Notice that the two parameters, **theObject** (or theCart or \$theObject) and **theAmount**, are used by this function. These parameters are the cart object and the value to be formatted, respectively.

To apply the formatting to your value, call the function by its name and pass the cart object name as the first parameter and the value you wish to format as the second parameter (bolded for clarification):

```
[ASP JScript]
<%=WA_eCart_DisplayMoney(cartName, cartName.GetCharges())%>
```

```
[ASP VBScript]
<%=WA_eCart_DisplayMoney(cartName, cartName_GetCharges())%>
```

```
[ColdFusion]
<cfoutput>#WA_eCart_DisplayMoney(cartName,
cartName_GetCharges())#</cfoutput>
```

```
[PHP]
<?php echo
WA_eCart_DisplayMoney($cartName, $cartName->GetCharges());
?>
```

By placing the value you wish to display in the function call as **theAmount** parameter into the function call, the value will now be displayed as formatted currency, according to the settings chosen in the WA eCart Object.

You can also add values in this function as shown in WA eCommerce Recipes – Recipe2: Database-driven Sales Tax and Shipping at:

### Currency Formats in the WA eCart Insert Checkout form

Three of the supported Payment Gateways in the WA eCart Insert Checkout Form object provide a form element that you may bind a currency value to in order to enable the payment gateway to display monetary values using your currency symbol. These are:

- Authorize.Net

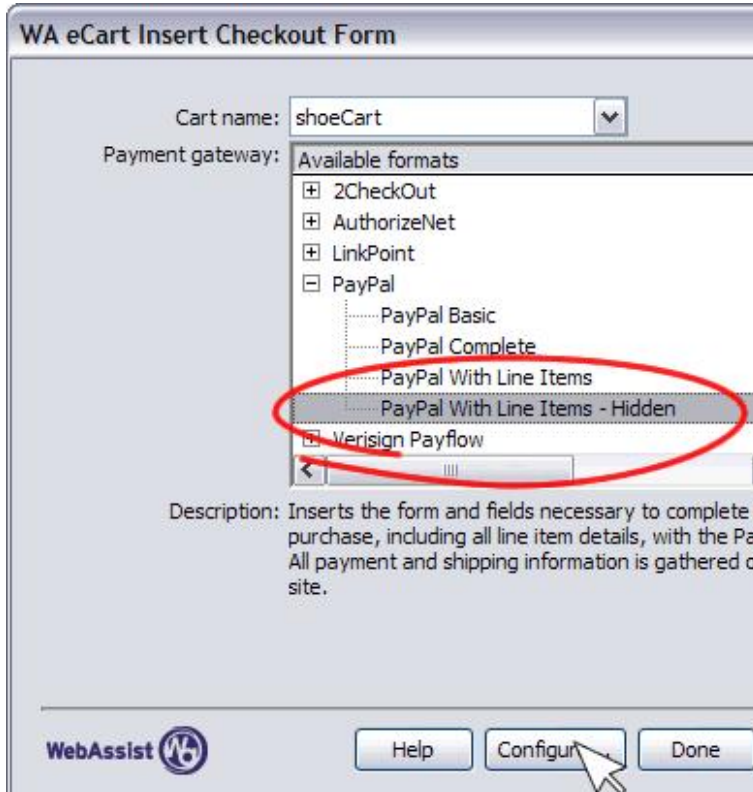
- PayPal
- WorldPay

WA Express Checkout provides an optional form element that you can add to your checkout form named "currency" which can be used to pass the currency symbol to the payment gateway for display. See FAQ Article number QA20050713163906 found in the Merchant Administration area for more information.

To assign a currency value to your Insert Checkout Form configuration, click on the WA eCart Insert Checkout Form object on the WA eCart Insert bar:



Click the + symbol to the left of your selected payment gateway to expand the options and select the desired Checkout Form from the available options. Click on the Configure button:



In the Configuration panel, select the label that represents the currency value to be used on the payment gateway and set the Default value to one of the supported currency codes for your payment gateway:



Configure Checkout Form

Label: Currency

Name: currency\_code

Default: GBP

Display as: hidden field

+ - Update

Label	Name	Default	Display
Paypal Transaction Type	cmd	_cart	hidden
Paypal Transaction Type	upload	1	hidden
Currency	currency_code	GBP	hidden
First Name	first_name		hidden
Last Name	last_name		hidden
Address	address1		hidden

Click OK and then save your configuration. Click Done to insert the checkout form with your currency value set for the proper form element.

If you are working with an existing checkout form, look for the hidden form element that passes the currency value. You can change the value that it passes in Code view:

```
142 <input type="hidden" name="currency_code" value="GBP">
```

Please consult the documentation for your particular payment gateway to determine if they support your desired currency symbol, how they implement it in shopping cart applications and name of the form element used to pass the desired value in the checkout form.