



**eCart 4.0
Help Documentation**

Copyright © 1999-2008 WebAssist.com Corporation
All rights reserved.

Preparing to use eCart

Prior to using eCart, it is important that your system and site be set-up properly.

The following sections outline the additional steps needed to develop a shopping cart application using WA eCart.

These steps includes creating your catalog of items for use with eCart, either on standard, static Web pages, or contained in a database.

Additional information is supplied to help those working with database-driven catalogs set up the needed data source.

- [Building your catalog](#) - Prepare all the information necessary for the items to be stored in your shopping cart. This section covers architecture and necessary info for both static and database-driven catalogs.
 - [Working with a payment gateway](#) - Discusses setting up a relationship with a payment gateway supported by eCart. This link will redirect you to our website and list all providers we directly support.
 - [Setting up for shipping](#) - Discusses setting up a relationship with a shipping provider to take advantage of eCart's real-time shipping quote feature. This link will redirect you to our website and list all providers we directly support.
-

Building your catalog

Shoppers visiting your online store need information about the products carried before they can decide which to buy.

You'll need to gather the product details — product names, SKU numbers, descriptions, images, and so on — prior to creating an online store catalog.

There are two basic types of catalog: static and database-driven.

- [Creating a static catalog](#) - a static catalog requires that item information is manually specified when designing the page. Information displayed for items on the page is added at design-time, both for descriptive pages about the item, as well as in shopping cart application itself.
 - [Creating a database-driven catalog](#) - a database-driven catalog retrieves item information from a data source, like Microsoft Access, MS SQL or MySQL. Information about an item is retrieved using a recordset, and then bound to page areas and within the shopping cart application itself.
-

Creating a static catalog

This section details the necessary considerations in creating a static item catalog for use within eCart. It discusses what information is necessary for your catalog items, the architecture for organizing this information, and how it relates to the development of your shopping cart application.

1. Create a complete inventory list.

This is an important first step in creating your catalog. Items in a catalog do not necessarily contain the same types of information; if you're selling umbrellas, your catalog will be different than if you're selling magazine subscriptions. Creating this list will help in designing the structure for specifying item information within the context of your shopping cart application.

2. Create a unique identification key for each item.

Each item available for placement in the cart needs to have unique identity to distinguish it within the shopping cart application. Typically, this is a SKU number or a similar inventory tracking mechanism. Identify how each of your items is uniquely identified, or if you have no system in place, create a identification system.

3. Determine categories for each item's data.

Once you have your list of items, determine the different categories of information used within the context of the shopping cart. At a minimum, the categories required for use with eCart are:

- the unique id for an item
- the price of the item

Additional categories which are commonly associated with products follows:

- Name
- Short Description, for use in catalog listings and within the shopping cart display
- Long Description, for use in detail pages.
- Weight

If you're planning on offering discounts for items, you should also add a category for discount price and, if a minimum is required to receive the discount, a category for discount minimum.

See [Columns](#) and [Calculations](#) in the [Understanding the eCart Object](#) section to learn more about how this information is used.

Creating a database-driven catalog

This section describes the necessary considerations when creating a database-driven catalog for use with eCart. It discusses what information is necessary for available items, the architecture for organizing this information, and how it relates to the development of your shopping cart application.

1. Create a complete inventory list.

This is an important first step in creating your catalog. Items in a catalog do not necessarily contain the same types of information; if you're selling umbrellas, your catalog will be different than if you're selling magazine subscriptions. Creating this list will help in designing the structure for specifying item information within the context of your shopping cart application.

2. Create a unique identification key for each item.

Each item available for placement in the cart needs to have unique identity to distinguish it within the shopping cart application. Typically, this is a SKU number or a similar inventory tracking mechanism. Identify how each of your items is uniquely identified, or if you have no system in place, create a identification system.

3. Determine categories for each item's data.

Once you have your list of items, determine the different categories of information used within the context of the shopping cart. At a minimum, the categories required for use with eCart are:

- o the unique id for an item
- o the price of the item

Additional categories which are commonly associated with products follows:

- o Name
- o Short Description, for use in catalog listings and within the shopping cart display
- o Long Description, for use in detail pages.
- o Weight

If you're planning on offering discounts for items, you should also add a category for discount price and, if a minimum is required to receive the discount, a category for discount minimum.

See [Columns](#) and [Calculations](#) in the [Understanding the eCart Object](#) section to learn more about how this information is used.

4. Create and populate the database.

Now that you have outlined the necessary architecture and content for your catalog, you may create one or more tables in your database which will provide this information for use in your shopping cart application. Create a table which has unique columns for all of the categories you have outlined and populate rows for each item entry, specifying each item's pertinent details in the appropriate column.

The eCart tutorial contains a database with an excellent example of just such series of tables; the Products table (products in MySQL) is the primary repository of catalog information. Other tables supplement the Products table with a number of categorizations (to easily separate shoes for men and women) and options like color and size.

Using eCart

This section discusses the individual features contained in eCart.

The following sections provide detailed information on building a shopping cart application using each of the available features:

- [Understanding the eCart Object](#)
 - [Using the Add to Cart Button](#)
 - [Creating a shopping cart display](#)
 - [Configuring checkout support](#)
 - [Implementing Shipping Services](#)
 - [Storing and retrieving shopping cart data](#)
 - [Prior version support](#)
-

Understanding the eCart Object

The eCart Object maintains all of the information specific to a shopping cart configuration within your site.

Shopping cart displays, buttons, and server behaviors all depend on the eCart Object to define the global attributes defined for the shopping cart. These attributes range from the way information for items in the cart are displayed to the way charges, discounts, and totals are calculated.

All pages that contain shopping cart functionality must have an eCart Object applied to the page. This provides the fundamental structure for the operation of the shopping cart.

It is possible to create multiple shopping carts within a single website that have different configuration requirements (for example, creating a shopping cart and a separate "wish list"). For this reason, it is possible to define more than one eCart Object within a site, and apply multiple shopping cart instances, if necessary.

Objects specific to a shopping cart instance are maintained in the eCart support folder, which is created in the root of your site when you first add shopping cart functionality to a page within your website (see [Site structure](#) for more information).

The following sections detail how to create, manage, and configure the attributes of an eCart Object:

- [Using the eCart Object panel](#): This section discusses how to access and use the eCart Object panel to add and edit instances of the eCart Object within your website.
 - [Defining the eCart Object](#): This section details configuration of an eCart Object. Configuration of the object is broken into the following categories:
 - [General](#): Sets the persistence option, currency and item weight settings.
 - [Columns](#): Sets the columns utilized within the cart, both default and custom.
 - [Calculations](#): Sets how totals are calculated for columns within the cart.
 - [Discounts, Charges, Taxes, and Shipping](#): Details the process of creating merchandising rules to establish how charges, discounts, taxes, and shipping are applied based on the contents of the cart and your business requirements.
-

Using the eCart Object panel

The eCart Object panel adds and edits the eCart Objects configured for use within your site.

Access

The following locations in Dreamweaver open the eCart Object panel:

- eCart Insert panel



- *Insert > WebAssist > eCart > eCart Object*
- *Server Behaviors > eCart > eCart Object*

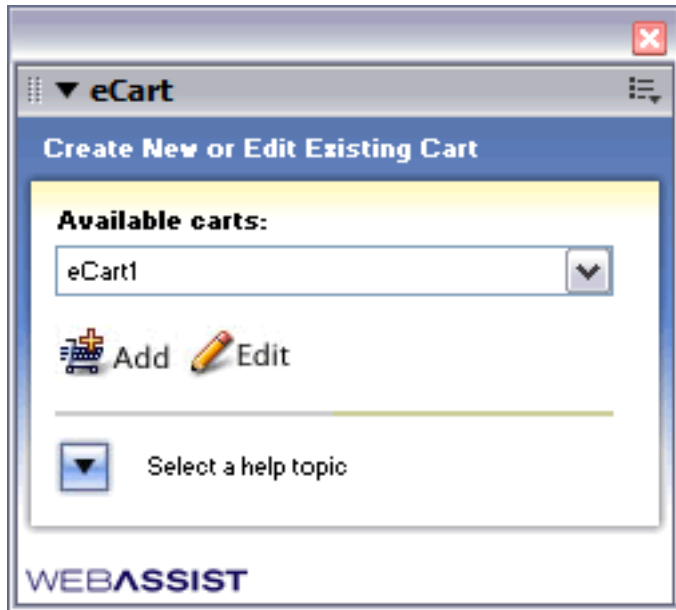
Managing eCart Objects

The following controls are available for adding and editing eCart Objects in this panel:

Available carts: Displays any existing eCart Objects configured for your site. These are stored in a root folder of your site that is created the first time an object is inserted into your site (see [Site structure](#) for more information).

Add: Opens the eCart Object user interface to specify a name and select the configuration options specific to a new eCart Object within your site. *Note:* You cannot change an eCart Object's name when you have completed configuration and exited the interface.

Edit: For an existing eCart Object selected from the Available carts: list, this button opens the eCart Object user interface to edit the current configuration options.

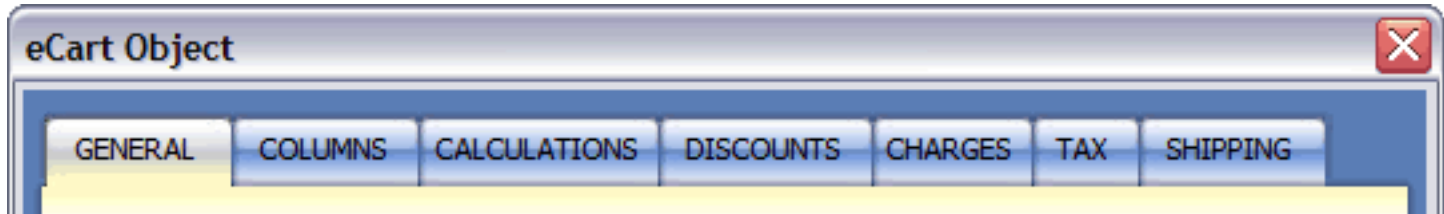


Note: For information on removing an eCart object, please refer to the [Site structure](#) section.

Configuring the eCart Object

The eCart Object dialog controls the configuration of the varying options specific to your shopping cart implementation, ranging from currency type to shipping calculations.

This information has been organized across multiple tabs in this dialog:



The first three tabs define the high level currency and data storage options, the columns for each type of data relevant to items placed in the cart, and possible calculations based on data available within columns in the cart. Information specific to these configuration options can be found in the the following sections:

- [General](#): Sets the persistence option, currency and item weight settings.
- [Columns](#): Sets the columns utilized within the cart, both default and custom.
- [Calculations](#): Sets how totals are calculated for columns within the cart.

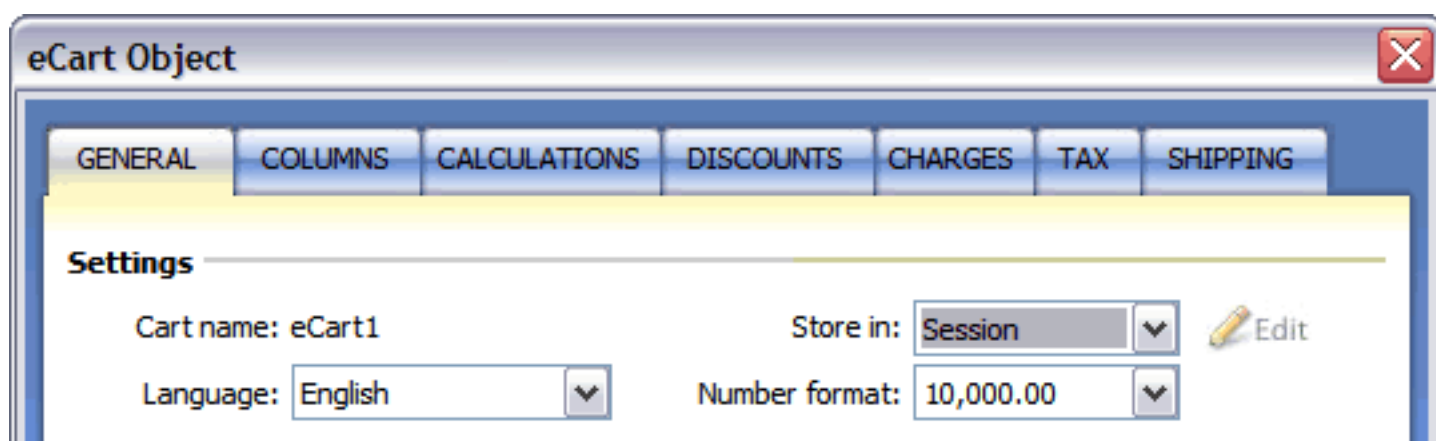
The remaining four tabs detail the configuration of additional merchandising options and are discussed in detail in the following section:

- [Discounts, Charges, Taxes and Shipping](#): Details the process of creating merchandising rules to establish how charges, discounts, taxes, and shipping are applied based on the contents of the cart and your business requirements.
-

General

The General tab of the eCart Object dialog sets the persistence type of the object, as well as the number format, currency and weight attributes.

Settings



Cart name: Identifies the eCart Object currently under configuration in the interface. This cannot be edited.

Language: Determines the language to be used for the descriptive text on the buttons available within the shopping cart. Available options include English, Spanish, French, German, Dutch, and Italian.

Store In: The Store In configuration option sets the persistence of the shopping cart. Persistence is the method for maintaining the contents of the shopping cart across all pages in which the eCart object is applied. There are two methods available for persistence:

- *Session:* Maintains the contents of the cart in the current session held by the user viewing the pages. All information will be available as a global variable for the duration of the session. The session variable is specified automatically when the eCart object is placed on the page.
- *Cookie:* Maintains the contents of the cart by setting a cookie on the client machine containing the current state of the cart. Using Cookie requires defining how that cookie is set on the page through the Edit button.



This opens the above interface and requires configuration of the following attributes:

- *Domain:* Enter the domain for which this cookie will be valid. You must specify at least two periods in order for this attribute to be set correctly (e.g. .domain.com) In this example, the default value has been left in to demonstrate. Leaving this blank will make the cookie valid for any domain which may call a cookie by the same name.

Note: If this is being run on a local web server, DO NOT specify a domain as the cookie object will not work.

- *Expires:* Sets the duration which the cookie will be set for in days.
- *Path:* Specify the subdirectory under the domain for which the cookie is valid. Using "/" will make the cookie valid for all directories off the root of the specified domain. A specified path will only allow the pages under that path and derivative folder names to be used.

(e.g. The path /subdir would allow the cookie to be used for /subdir, /subdirectory, and /subdir/content.htm.)

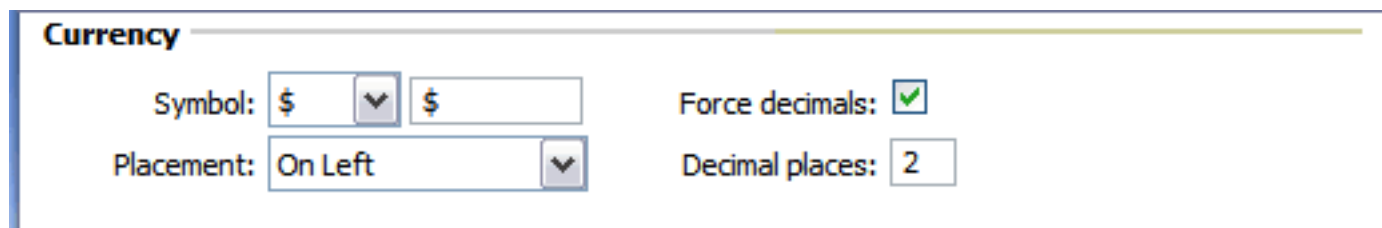
- *Secure:* Sets whether the information in the cookie is passed using https:// security protocols. Use this setting if the cookie is going to be set from a page that is using the https:// protocol.

Number Format: Sets the formatting options for maintaining numbers within the eCart object. The available options are:

- 10,000.00
- 10.000,00
- 10000.00
- 10000,00

Currency

The following configuration options are specific to currency within the scope of the shopping cart:



Symbol: Sets the symbol utilized to identify the currency for a number. You may select Dollars, Euros, Pounds, Francs, or Yen from the list to default in the symbol into the entry box, or specify your own symbol.

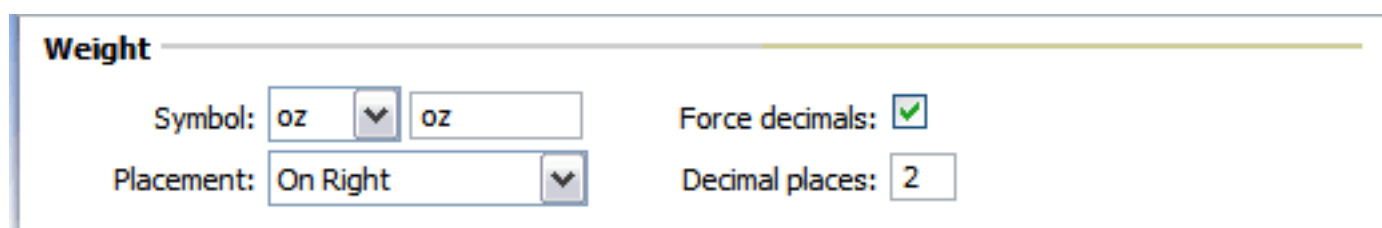
Placement: Sets where the symbol is displayed in relation to the number it modifies.

Force Decimals: Tells the shopping cart display to show decimal places for the currency.

Decimal Places: If a value has a decimal, the currency will maintain the number of decimal places specified.

Weight

The following configuration options are specific to item weight within the scope of the shopping cart:



Symbol: Sets the symbol utilized to identify the weight for a number representing an items weight. You may select lbs., Kg, oz., g, t, or tn from the list to default in the symbol for each, or specify your own symbol.

Placement: Sets where the symbol is displayed in relation to the number it modifies.

Force Decimals: Tells the shopping cart display to show decimal places for the weight.

Decimal Places: If the value has a decimal, the weight will maintain the number of decimal places specified.

Columns

Columns define information types pertinent to each item added to the cart. The Columns configuration tab defines what columns are available for use within your cart.

One column typically contains the unique identifier for a specific product, such as the ID column. A unique column may be selected from all available cart columns, but typically the ID column is used by default.

The following columns are provided by default within the object and are followed by their data output types. Though they do not need to be used or displayed, they cannot be removed from the object and require a zero value if not in use:

- **ID:** Number
- **Name:** Text
- **Description:** Text
- **Weight:** Weight
- **Quantity:** Number
- **Price:** Currency

Creating additional columns

Custom columns can be defined by creating a column name and selecting the appropriate output type. The six available output types are Number, Text, Currency, Date, Integer, and Weight.

Additional columns prove useful in storing data for each item which may be of use within the cart.

Use the **Add** button to insert a new column into the list of available columns. Use the **Delete** button to remove a selected column within the window.

The screenshot shows the 'eCart Object' configuration window with the 'COLUMNS' tab selected. The window title is 'eCart Object' and it has a close button (X) in the top right corner. The tabs are GENERAL, COLUMNS, CALCULATIONS, DISCOUNTS, CHARGES, TAX, and SHIPPING. The main area is titled 'Create new or edit existing columns' and contains an 'Add' button (plus icon) and a 'Delete' button (trash icon). Below this is a table with columns 'Name', 'Type', and 'Unique'. The table lists the following columns: ID (Number, Unique), Name (Text), Description (Text), Weight (Weight), Quantity (Number), and Price (Currency). Below the table are three fields: 'Name:' (text input), 'Output Type:' (dropdown menu showing 'Text'), and 'Unique:' (checkbox). At the bottom left is the 'WEBASSIST' logo, and at the bottom right are 'HELP', 'OK', and 'CANCEL' buttons.

Name	Type	Unique
ID	Number	✓
Name	Text	
Description	Text	
Weight	Weight	
Quantity	Number	
Price	Currency	

To update an existing column:

1. Select an existing column from the list of available columns. The configuration details for the selected column populate the console below the columns listing.
2. Make any edits or changes to the column.
3. Select another column to apply the changes within the display pane.
4. Click OK to apply the changes to the eCart object.

Note: Modifying and specifying additional cart columns requires that you manually add that column to all existing applications of eCart features that are dependent on column configuration, such as [Add to Cart](#) buttons and [shopping cart displays](#).

Calculations

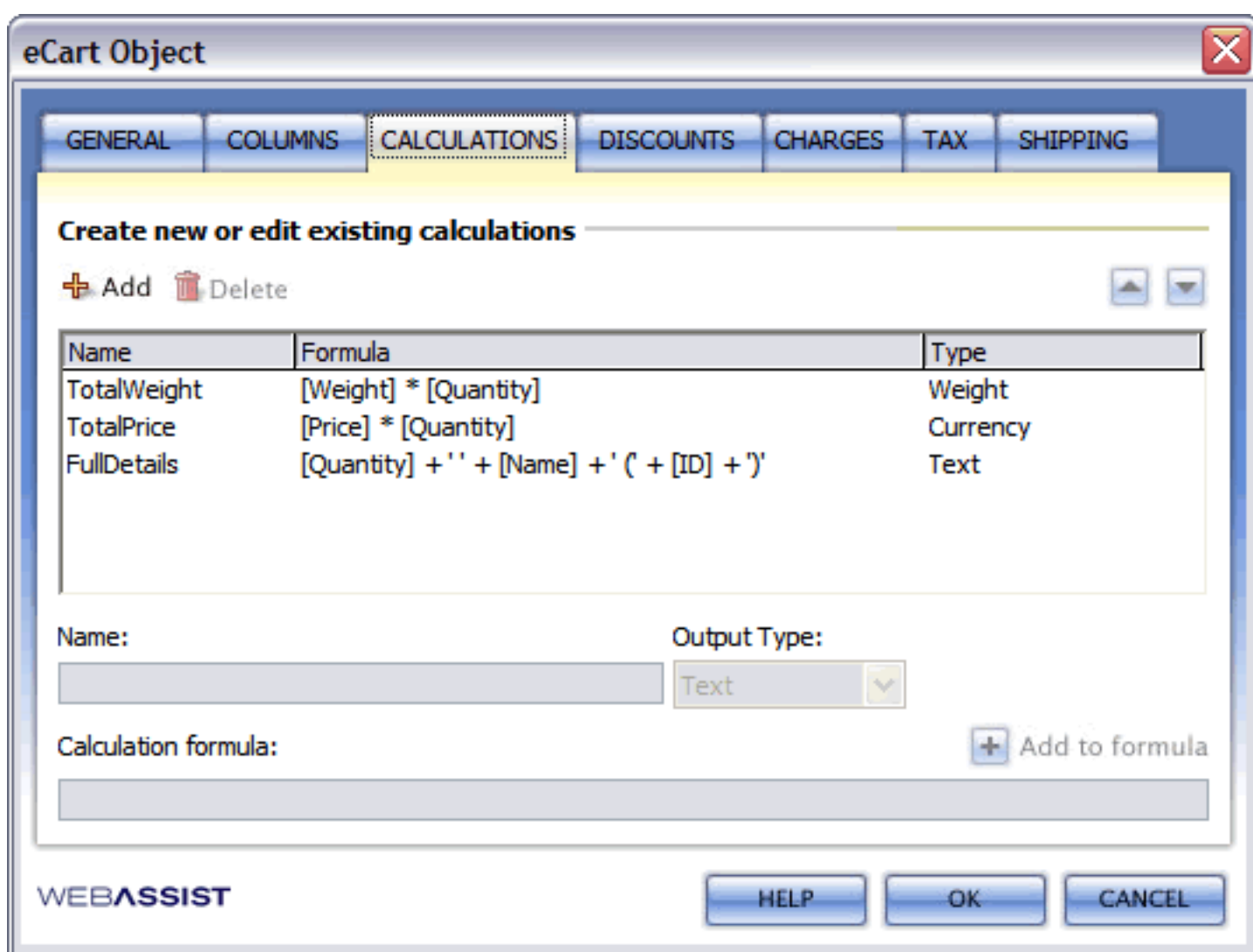
The eCart Object can be configured to calculate values based on column contents of your cart, or custom calculations written in server-side code. Calculations refer not only to numeric values, but the concatenation of text from columns in the cart.

The results of these calculations can then be displayed as custom columns within the shopping cart display.

For example, three commonly used calculations are automatically available upon creation of the object:

- **Total Weight:** for a line item in the shopping cart, this is the weight of the item multiplied by the number of that item in the cart.
- **Total Price:** for a line item in the shopping cart, this is the price of the item multiplied by the number of that item in the cart.
- **Full Details:** this is a string built up for each line item that contains the name of the item, its ID, and its quantity. It is typically used for posting a basic transaction description from a checkout form.

By default, these calculations are available as display columns when creating a [shopping cart display](#).



To create a custom calculation:

1. Click the **Add** button to add a blank calculation to the eCart Object.
2. In the Name field, specify a unique name for the calculation that will identify its result within the Display Manager. Be sure that the name is not the same as any existing column or calculation configured in the eCart Object.
3. From the Output Type list, select one of the six available data types to format the calculation result. Available options are Number, Text, Weight, Integer, Date and Currency.
4. In the Calculations field, specify the calculation by one of the following methods:

- **Numerical column calculations:** you may add, subtract, multiply and divide values available in specified columns of the cart using these corresponding operators: +, -, *, /. Columns are specified within a calculation by placing the column name in brackets - all lower case.

e.g. TotalPrice = [Price] * [Quantity]

English equivalent:

Total Price for a specific quantity of an item in the cart
is equal to

the price of an item in the cart (retrieved from the cart Price column)
multiplied by (the quantity for that item
the quantity of the item in the cart (retrieved from the cart Quantity column)

- **Text calculations:** the text from multiple columns can be displayed within a single column by concatenating the information. Columns are specified within a calculation by placing the column name in brackets - all lower case. Literal text to be displayed must be encapsulated in quotation marks.

e.g. a user sells T-Shirts and has a column of Color defined for the item. To display the item and the color within a single column requires the following calculation:

Display Text: VNeck: Blue

Calculation: [Name]+": "+[Color]

- **Server-side calculations:** Advanced users can write server-side code within the calculation field to write custom calculations based on the global variables available in the site.

Note: To assist in building your calculation string, you may use the Add to formula feature to select from a list of available cart columns or default calculations and place the string corresponding to that value in the calculation field.

5. Select a different calculation in the display pane to apply your changes.
6. Proceed with additional configuration or click OK to complete the eCart Object configuration.

To update an existing calculation:

1. Select an existing calculation from the list of available calculations. The configuration details for the selected calculation populate the console above the calculations listing.
2. Make any edits or changes to the calculation.
3. Click the Update button to apply the changes within the console. Changes will populate the listing for the selected calculation.
4. Click OK to apply the changes to the eCart object.

Calculations and Merchandising

For events and conditions used in creating merchandising rules, calculation results are available as conditions in the cart in the same manner that columns are.

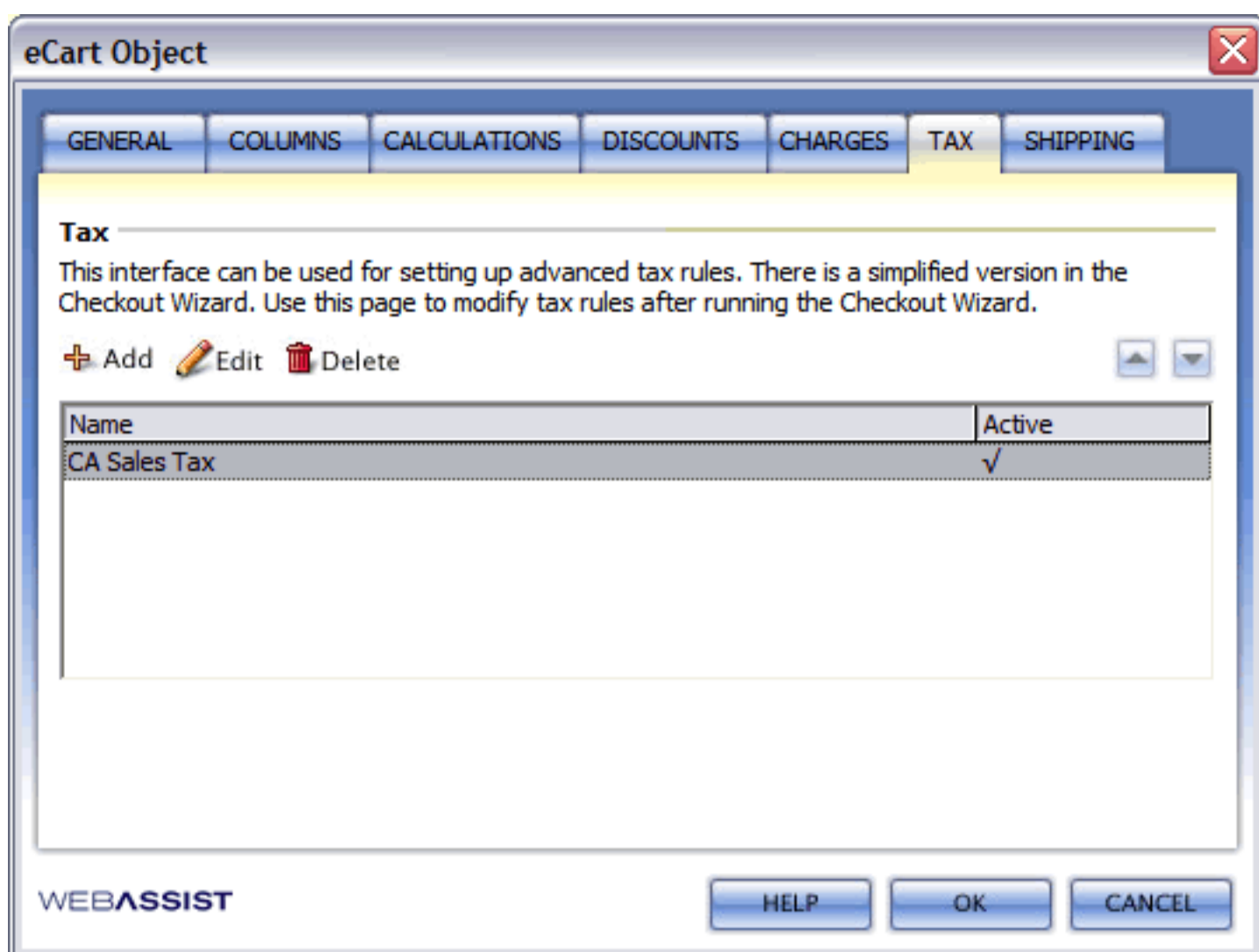
Please see Available Conditions and Events for [Discounts, Charges, Taxes and Shipping](#) for information pertinent to calculations for each of these rule types.

Discounts, Charges, Taxes and Shipping

The eCart Object can be configured to perform additions and subtractions from the subtotal of the items in your cart based on conditions configured within the object. This allows you to create merchandising configurations in your cart that are triggered when the cart satisfies specified conditions.

Aside from the contents of the cart, 4 additional categories of line items can be added and displayed to capture the discounts and charges that meet your business requirements. Discounts decrease the cart total, and charges, tax, and shipping increase the cart total. These categories are broken into the last 4 tabs of the eCart Object dialog:

- **Discounts**
- **Charges**
- **Tax**
- **Shipping**

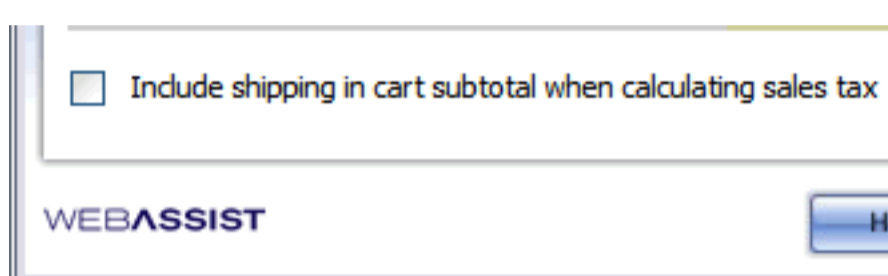


Rules specific to each of these categories can be added, edited, and removed in each of these tabs.

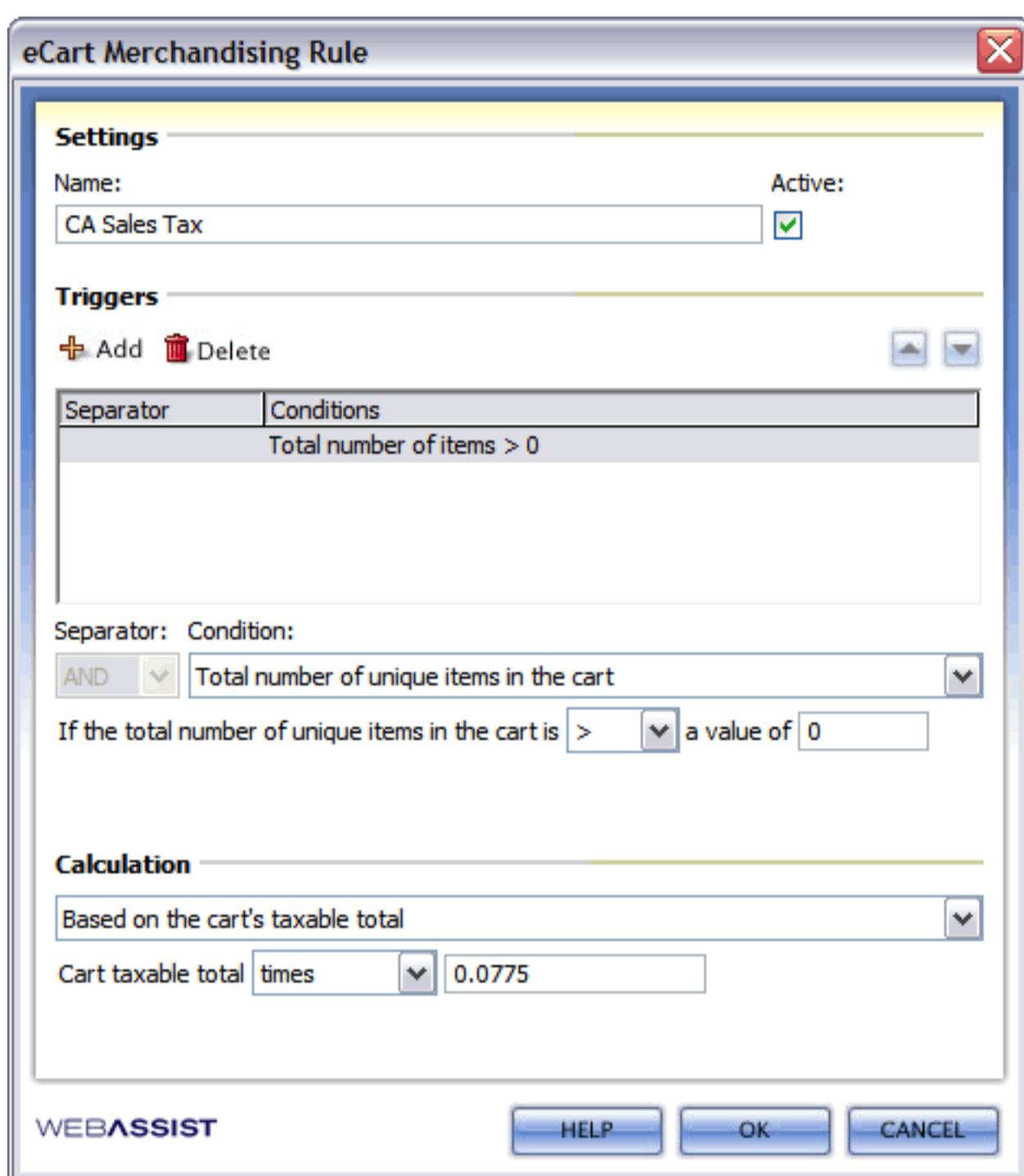
The up and down arrows change the order in which the rules are listed in the shopping cart display within the category.

The name of the rule, and the value calculated for it, is displayed in the corresponding category in your shopping cart display, dependent on how your display is configured (see [eCart Display Manager](#) for more information).

The only difference amongst these tabs is that the Shipping tab provides the option of including shipping costs when calculating sales tax using rules in the Tax tab:



The Add and Edit functions access the eCart Merchandising Rule dialog that configures the settings, triggers, and calculation for a specific rule.



Settings

This section sets the name of the rule, identifying it within the eCart Object dialog, as well as setting the label shown in the shopping cart display for the corresponding value returned. As well, rules can be marked as active or inactive, allowing you to disable and maintain rules for future use without making them currently active within the shopping cart.

Triggers

Triggers are one or more conditions that must be satisfied for the rule to be applied to the shopping cart. Select a condition from the available list and configure the settings specific to your business requirements. Use the Add button to add it to the display.

When multiple conditions are specified, choose either the AND or OR separator to determine how the selected condition interacts with its predecessor in the list. Use the up and down arrows to adjust the order of a condition selected in the display to assist in managing these relationships.

The AND separator requires that the two conditions its separates must both be true. In contrast, the OR separator requires that if either condition is satisfied, the requirement to apply the rule is fulfilled.

The following is a list of all conditions available for use in rule configuration:

- Total number of unique items in the cart
- Number of items with a specific column value
- Presence of an item in the cart
- Quantity of a specific item in the cart
- Taxable total for the cart (Tax configuration only)
- Subtotal for the entire cart contents
- Subtotal for any column
- If today is after a certain day
- If today is before a certain day
- Session variable exists
- Based on a session variable value
- Based on Australia Post shipping quote value
- Based on Canada Post shipping quote value
- Based on DHL shipping quote value
- Based on FedEx shipping quote value
- Based on UPS shipping quote value
- Based on US Postal Service shipping quote value
- Custom Expression evaluates to true

Calculation

The calculation determines the value returned when the conditions set in the trigger are satisfied, and adjusts the cart total accordingly.

Select a calculation from the available list and configure the settings specific to your business requirements to calculate the correct value applied. The basis for each calculation can typically have a specified value added or subtracted, or adjusted by a specified multiplier or divisor.

The following is a list of all calculations available for use in rule configuration:

- Based on the quantity of an item
- Based on an item's column value
- Based on the cart's taxable total
- Based on the cart's subtotal (on the Charges tab, the option exists to determine whether to calculate this before or after Discounts have been applied)
- Based on items with a specific value
- Based on multiple of column subtotal
- Based on column subtotal
- Based on Australia Post shipping quote value
- Based on Canada Post shipping quote value
- Based on DHL shipping quote value
- Based on FedEx shipping quote value
- Based on UPS shipping quote value
- Based on US Postal Service shipping quote value
- Flat rate

Using the Add to Cart Button

The Add to Cart Button adds items to the shopping cart and passes all information pertinent to the configured item to the shopping cart.

Item information is stored within the shopping cart columns that are defined in the eCart Object specific to your shopping cart implementation. See [Columns](#) in the section [Understanding the eCart Object](#) for more information on configuring shopping cart columns.

The Add to Cart Button is flexible in terms of defining information that populates the columns of the shopping cart. You can specify column data:

- statically during the configuration of the button
- dynamically from a database using a recordset, or other server-side code reference
- using a form field associated to the button, allowing the user to control column data (e.g. Quantity)

Configuration of the Add to Cart Button is broken into three tabs within the Add to Cart Button dialog, each of which is discussed in detail in the following sections:

- [General configuration](#): configures whether the button uses static information, or information retrieved from a recordset to populate shopping cart columns when an item is added to the shopping cart. Also specifies the default redirect location for the button when clicked.
- [Bindings configuration](#): configures the information that populates each shopping cart column for the item. Information can be specified statically within the interface, or can be retrieved from a recordset or server-side code. As well, the redirect action and display configuration for the item within the shopping cart display can be further customized.
- [Button configuration](#): specifies the image or form button used for the display of the button, as well as the ALT text if an image is used.

Note: Applying and configuring the Add to Cart Button is discussed in detail in the tutorials provided with this extension. Please refer to the Getting started guide to learn more about configuring the Add to Cart Button specific to your needs.

Feature access

The Add to Cart Button is available for insert from the following locations:

- eCart Insert panel



- Insert > WebAssist > eCart > Add to Cart Button
- Server Behaviors > WA eCart > Add to Cart

To insert the Add to Cart Button:

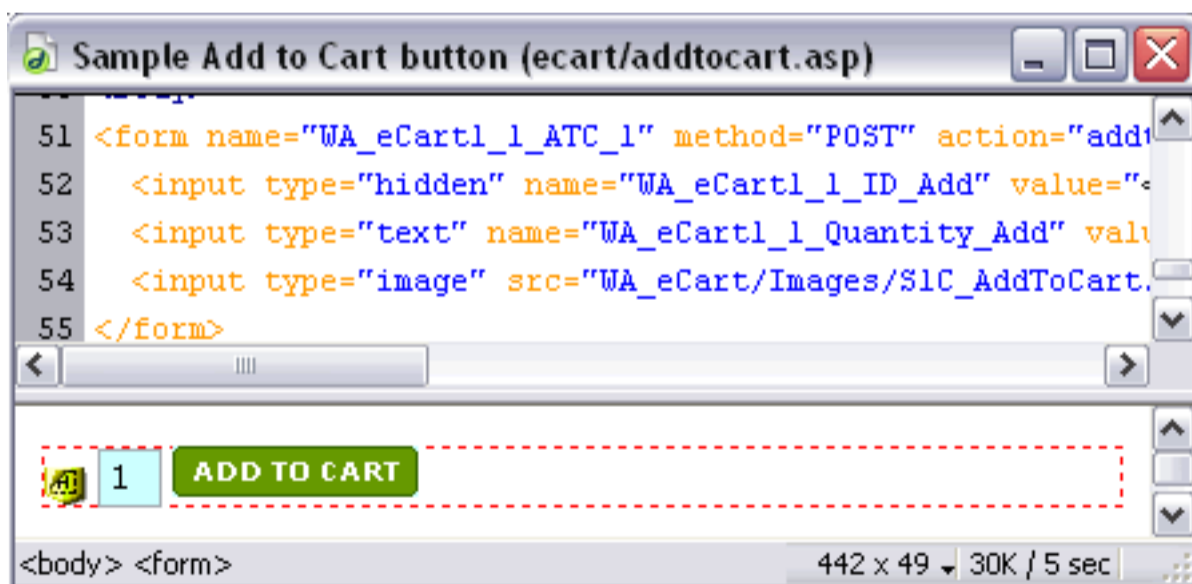
1. In Design View, place your cursor in the location where the Add to Cart button is to be placed.
2. In the eCart Insert panel, click the eCart Add to Cart Button icon (see Feature access above).

The configuration dialog box will open.

3. Configure the button specific to your needs, as outlined in the sections detailed above.
4. Click OK to apply the button to the page.

A form is placed on the page that contains:

- a hidden field containing the ID of the item
- all fields that allow the user to specify column data (e.g. Quantity)
- a form button or image button, depending on your configuration



In addition, server-side code is placed at the top of the page that contains all the configuration information specific to the item.

To modify the Add to Cart button:

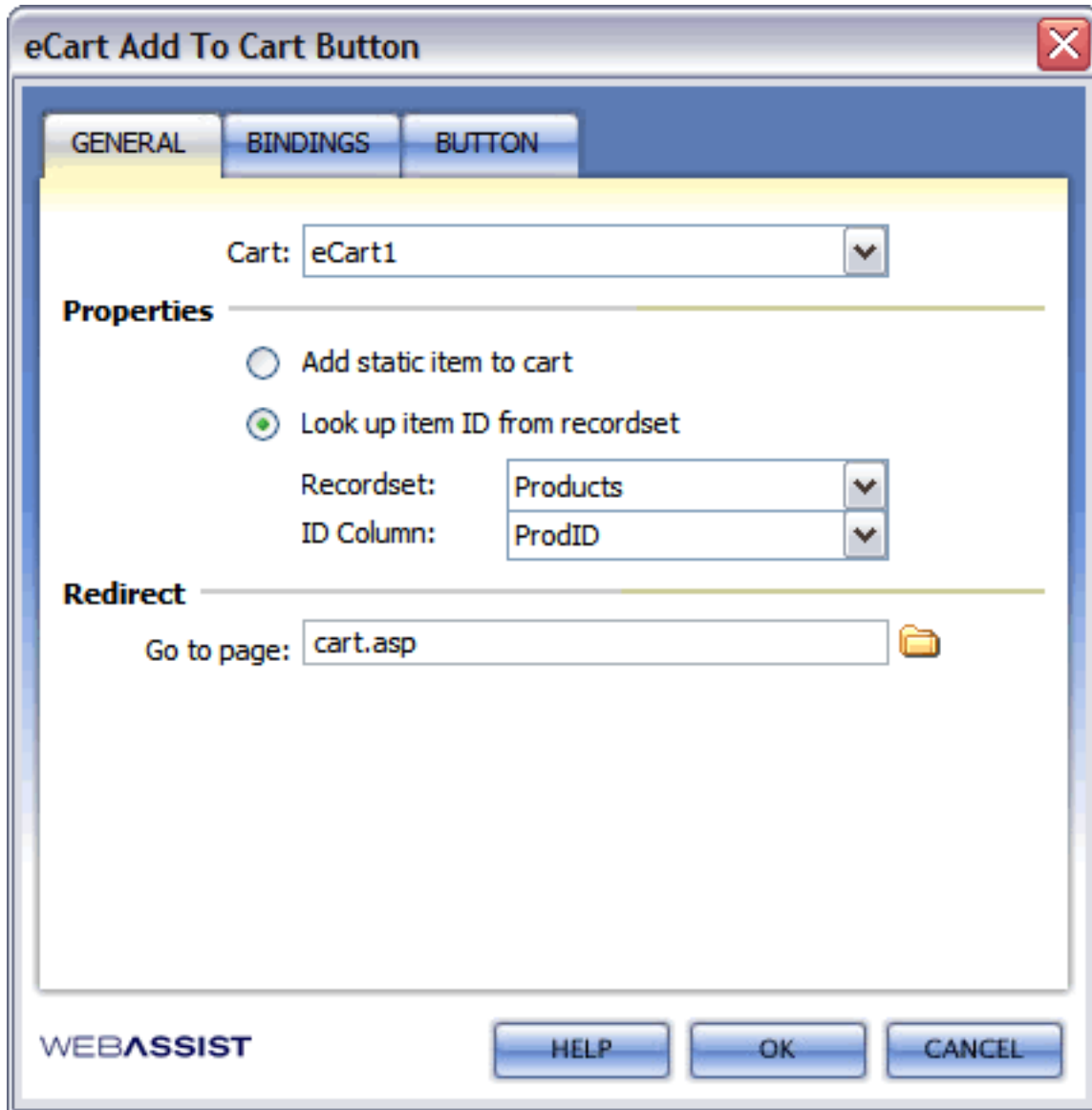
1. In the Server Behaviors panel of the Application panel group, click the eCart Add to Cart listing which corresponds to the button to be modified.

The configuration dialog box will open.

2. Make any necessary configuration changes.
 3. Click OK to apply the changes to the page.
-

General configuration

The general tab of the Add to Cart Button dialog configures the eCart Object associated to the button, whether the properties of the item are retrieved from a recordset or specified directly, and where the button is re-directed to when clicked.



The screenshot shows the 'eCart Add To Cart Button' dialog box with the 'GENERAL' tab selected. The 'Cart' dropdown is set to 'eCart1'. Under the 'Properties' section, the 'Look up item ID from recordset' radio button is selected. The 'Recordset' dropdown is set to 'Products' and the 'ID Column' dropdown is set to 'ProdID'. Under the 'Redirect' section, the 'Go to page' text box contains 'cart.asp' and a folder icon is visible to its right. At the bottom, there are 'HELP', 'OK', and 'CANCEL' buttons, and the 'WEBASSIST' logo is in the bottom left corner.

To complete General configuration for the Add to Cart button:

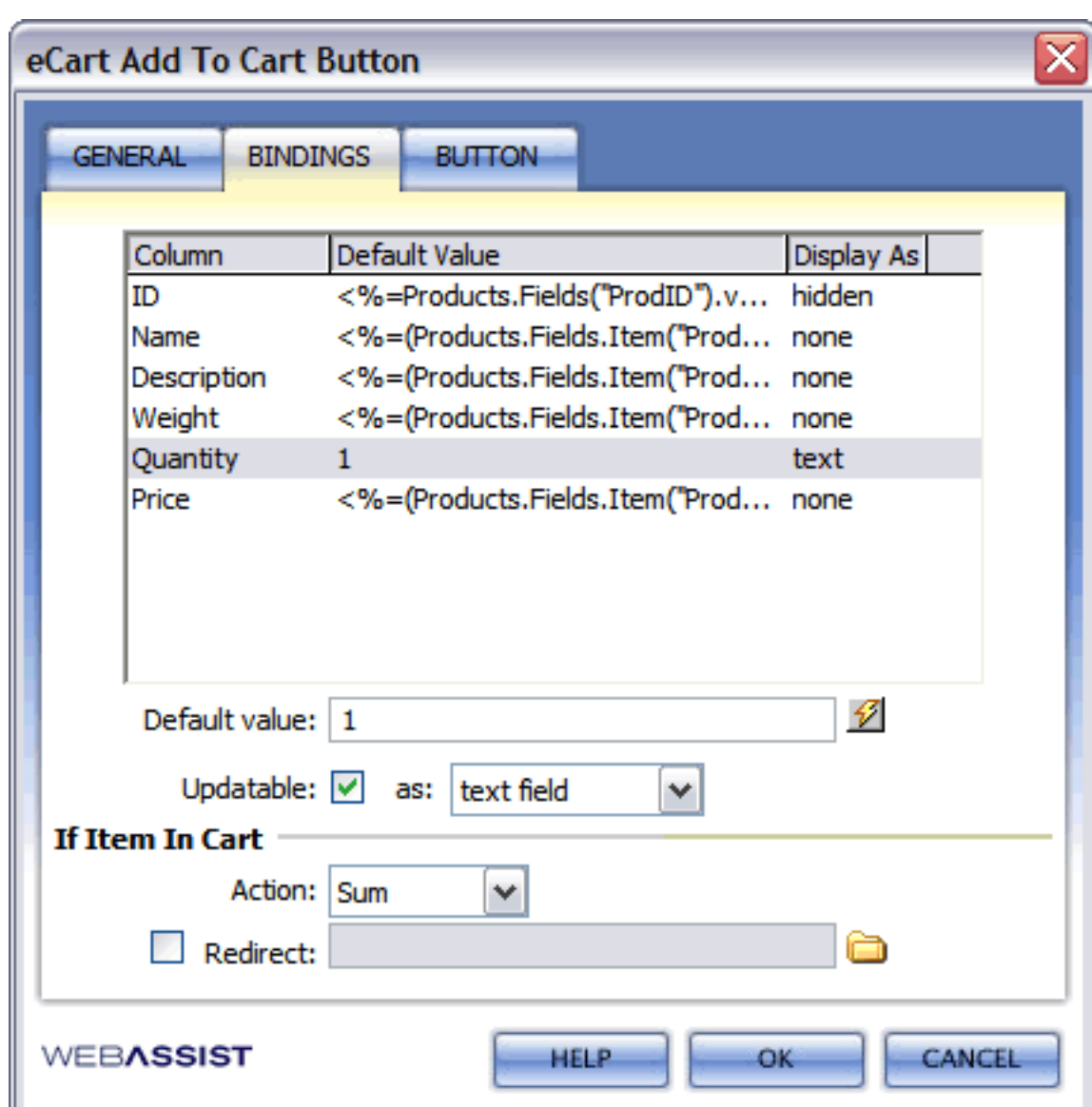
1. From the Cart name list, select the eCart object available on the page which corresponds to the shopping cart to be populated with this item.
2. Select whether the item to be added to the cart is to be defined statically or retrieved from a recordset. The specified item ID is placed in a Hidden form element associated with the Add to Cart button when applied.
 - o **Static definition:** Specify a unique Item ID which will identify the product within the shopping cart.
 - o **Recordset definition:** Select a recordset on the page which provides the item's definition information. Specify the column within the recordset which contains the unique id for the item.
3. Specify the destination page which is redirected to upon adding an item to the shopping cart.

Use the Browse button to select a file within your site or specify an http:// location.

Leave this field blank to redirect to the current page.

Bindings configuration

The Bindings tab of the Add to Cart button dialog configures the data source for the available cart columns defined in the eCart Object. As well, it determines how the item is added to the cart, and if a redirect takes place based on the item's addition to the cart.



To complete Bindings configuration for the Add to Cart button:

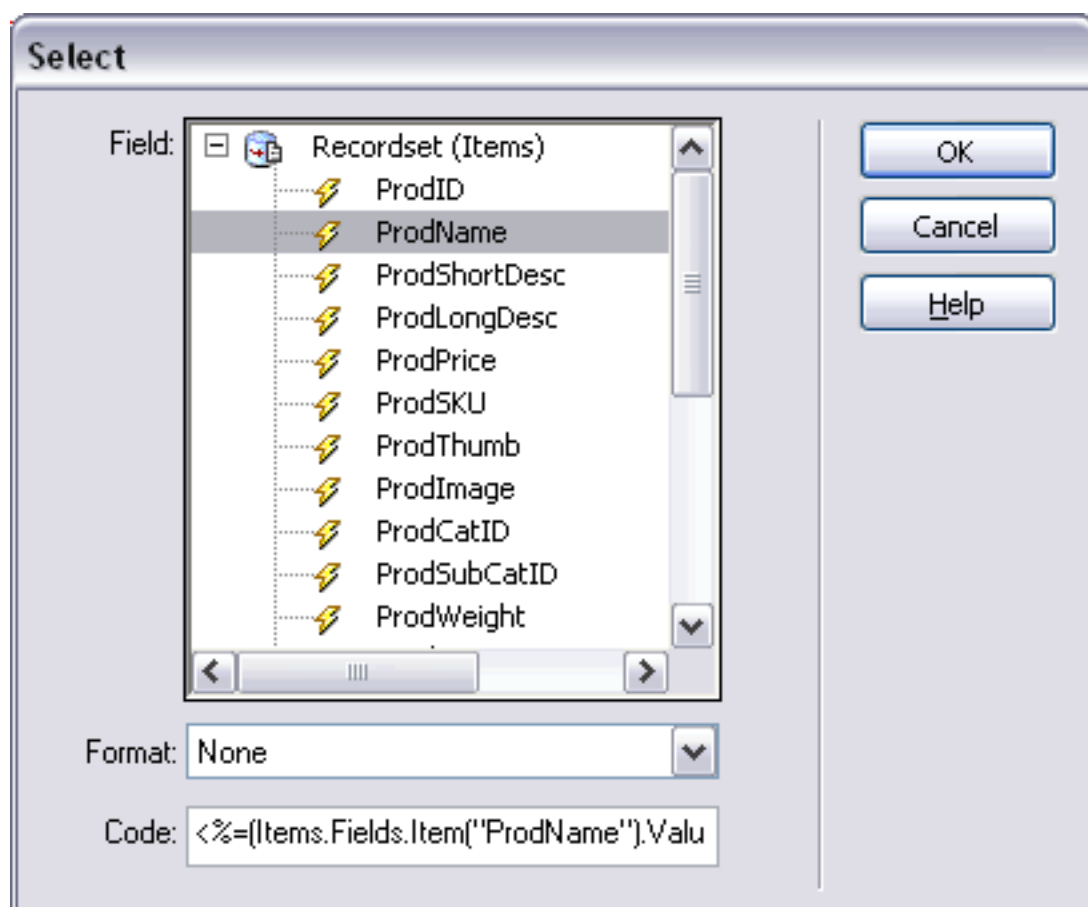
- Click on the Bindings tab to continue with configuration. Bindings information correlates item properties with the columns that store the data within the cart, as well as the method used to populate the item into the cart contents.
- The configuration window lists all shopping cart columns available for the specified eCart Object (see [Columns](#) for more information) and their corresponding values for the given item. The following default columns are always available for any eCart object definition:
 - ID
 - Name
 - Description
 - Weight
 - Quantity
 - Price

Note: If you have previously configured an Add to Cart button and then add a custom column to the corresponding eCart object, you must return to the Add to Cart and manually bind the new column or you will get an error. The new column will not be automatically detected from the eCart Object for the Add to Cart button configuration.

Values for each column are specified by selecting a column from the display pane and specifying the data for that column in the *Default value:* field.

Default columns require a value of zero if no data is bound to them.

- To configure static items, enter static values to bind to the ID, Name, Description, Weight, Quantity and Price of the item.
- To configure items retrieved from a recordset, select recordset columns to specify the ID, Name, Description, Weight, and Price of the item through the available data bindings:



- You have the option of making a column value editable and including an updateable form field to the user to specify the information for that column.

You may configure any column to be updateable. If a column is updateable, you may then select the type of form field that will retrieve the associated column's data for this item.

Note: Using a select list to update a cart column requires that you configure the list's options after insertion of the Add to Cart Button. This can be done directly through the Property Inspector for the list.

By default, the Quantity of the item added to the cart is automatically configured to be set using a text field placed on the page with the Add to Cart button. If the quantity of the item added to the cart is not determined by the user:

- Change the binding to reflect how the quantity is to be set (static or server-side code).
- Uncheck the Updateable: checkbox.

- In the If Item in Cart section, select one of the following options from the *Action* list to determine how the item is populated into the shopping cart if the same item is already present:
 - Sum:** increases the quantity of the item in the cart.
 - Ignore:** does not allow the item to be added.
 - Add New:** creates a new instance of that item in the cart separate from the existing instance.
 - Replace:** adds the item to the shopping cart and replaces the existing item

If you wish the user to redirect to a page other than the current page, click the Redirect checkbox and enter a relative URL or a full http:// address where the user is to be directed to upon adding the item to the shopping cart. This redirect will override the redirect specified on the General tab for this situation when a location is specified.

Button configuration

The Button tab in the Add to Cart Button dialog configures the type of button displayed. Users have the option of selecting from available images in eleven color schemes, a form button, or may specify a custom image location.

Provided images correspond to the available display skins provided for the shopping cart display.

As well, the Alt text displayed on rollover of the button can be specified.



To complete Button configuration for the Add to Cart button:

1. Click on the Button tab to configure the visual properties of your Add to Cart button.
2. Specify the button to be placed on the page.

You have the option of selecting from a form button or several image buttons provided with the extension, as well as specifying an image button that you provide.

The Button URL field will accept an `http://` location, or a relative image file within your site.

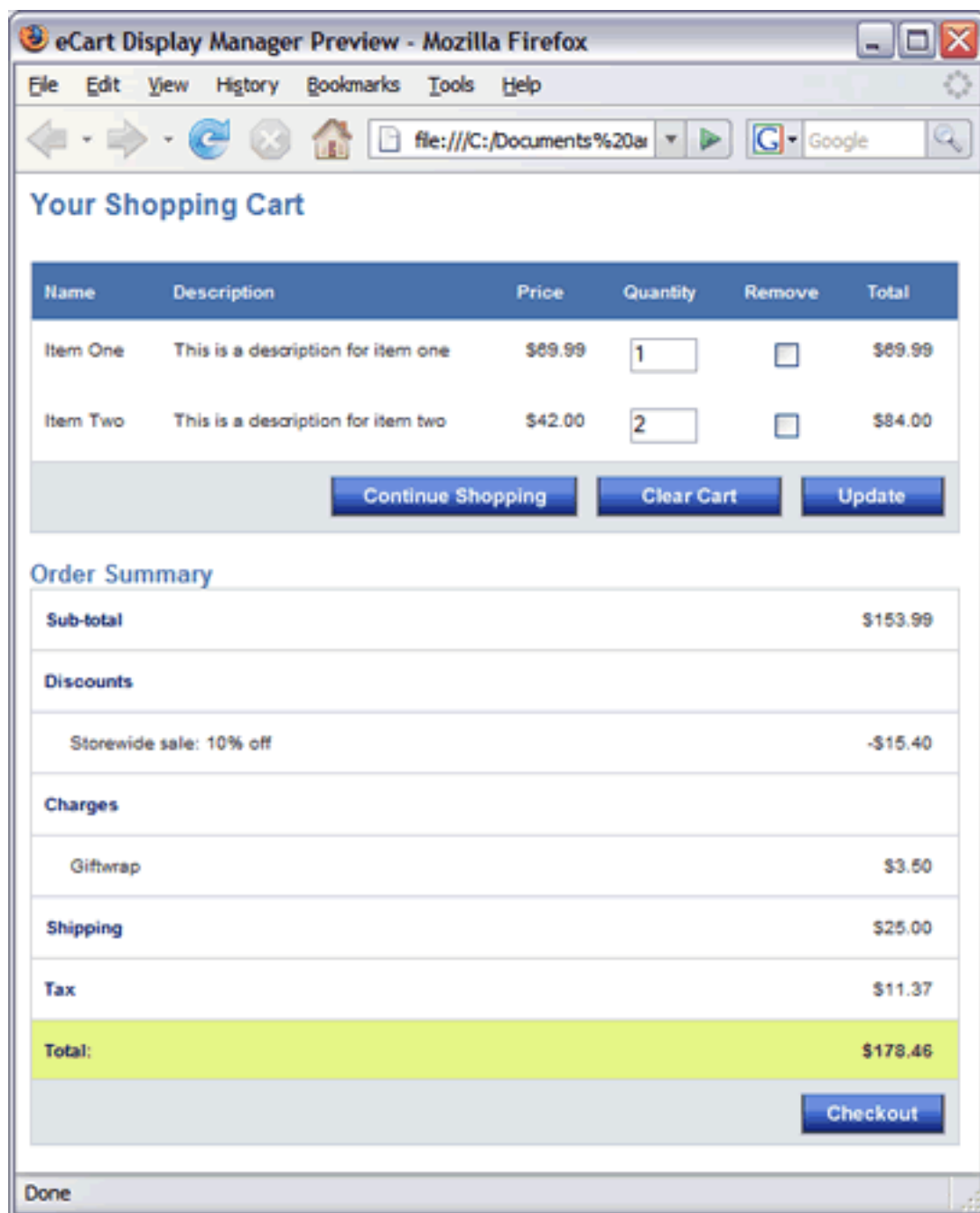
3. Enter text or server-side code to specify the alt text that will be displayed on the rollover event for the button. Leaving this blank removes the alt text attribute for the button.
-

Creating a shopping cart display

One of the primary components of a shopping cart is the shopping cart display: the portion of your shopping cart that lists all items added to the cart, their detail information, and any additional discounts and charges applied to the cart total, including tax and shipping calculations.

Items added to the shopping cart can be configured to allow updates to item information contained within the cart, the most common example being the quantity of an item. Buttons included in the display allow the user to:

- update items
- redirect to a specified page to continue the shopping experience
- remove selected items
- initiate the checkout process



Features

eCart ships with the eCart Display Manager feature: a simple step-by-step wizard that configures your shopping cart display. 3 professionally designed display skins are available in 11 different color schemes for you to use to create your display design. Users have the option of using provided button images or form buttons for each skin, offering a total of 66 different skin configuration possibilities. Each skin uses CSS to control the design attributes that renders the shopping cart.

In addition, there are four types of shopping cart displays available for insertion in the page:

- Updateable Cart - fully functional shopping cart display
- Read-Only cart - display only, users will not be able to apply changes to the shopping cart from within the display
- Email-friendly cart - display only, this cart design is designed for sending order details to user via email, providing a display type that is compatible with most html email-rendering engines.
- Quick Cart - minimal cart display providing the number of items, the cart total, and the ability to re-redirect to the Updateable Cart, clear the contents of the cart, or checkout.

The eCart Display Manager inserts all necessary code for the shopping cart display to function on the page. Once inserted, you may use the WYSIWYG features of Dreamweaver to customize the shopping cart display layout to your needs.

As well, each component of the shopping cart display is available as individual server behaviors and data bindings within eCart. Instead of using the eCart Display Manager, advanced users who are comfortable with the architecture of the shopping cart display may develop using each of these components in conjunction with their own html and CSS display layout.

Sections

This portion of the help contains the following sections to assist in the creation of a shopping cart display using the features shipped with eCart. Each links to the beginning of that section, and contains a brief summary of the content covered by each. We suggest you review all sections in order to familiarize yourself with the features available within eCart, as well as the shopping cart display development process, prior to beginning development:

- [Using the eCart Display Manager](#)
- [Shopping cart display components](#)
- [Customizing your shopping cart display](#)

Using the eCart Display Manager

The eCart Display Manager is the easiest, and most direct way to create a shopping cart display for your website. This feature steps you through a wizard to configure the components of your shopping cart display. Once completed, all the necessary run-time code is placed on your page to render the display, as well as control the options and actions the end-user will be able to use in the shopping cart display.

The wizard consists of three configuration steps. Each of the steps configures a different aspect of the shopping cart display configuration, and are covered in detail in the pages that follow.

The following is a breakdown of each of these steps, with links to the help page specific to each. You may also use the navigation arrows above to step through each section as you would step through the wizard. Please review the information provided below these section breakouts prior to advancing:

- [Step 1 of 3: Configure the design of your cart](#)
This section details the configuration options for selecting from available shopping cart display skins. It includes selecting the layout design, color scheme, buttons, and type of shopping cart display.
- [Step 2 of 3: Configure the columns of your cart](#)
This section details the configuration of the shopping cart columns to be included in the display.
- [Step 3 of 3: Set Discounts and Charges display](#)
This section details the configuration of how charges and discounts are rendered within the shopping cart display.
- [Completing display configuration](#) This section details the insertion of the shopping cart display components on the page, as well as additional configuration considerations for the newly inserted shopping cart display.

Before you begin...

The following considerations should be taken into account prior to using the eCart Display Manager to create a shopping cart display:

1. The eCart Object must be applied to the page prior to using the eCart Display Manager. If you are unfamiliar with eCart's shopping cart architecture, please review the following sections to familiarize yourself with this component:
 - Getting started guide
 - [Understanding the eCart Object](#)
2. Prior to inserting a shopping cart display using the eCart Display Manager, it is necessary to configure any custom columns within the eCart Object to be shown in the shopping cart display. Otherwise, these columns will not be available for display configuration within the wizard.

Further, changes made to columns in the eCart Object after creating a shopping cart display require that you create the display again for those changes to be reflected correctly within the shopping cart display. Save yourself the trouble and configure your columns in advance.

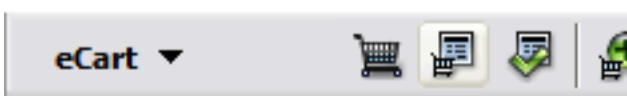
3. Prior to inserting a shopping cart display using the eCart Display Manager, it is necessary to configure any merchandising rules that applies charges and discounts to the shopping cart if the display will list each charge and discount rule as an individual line item. Otherwise, these line items will not be added to the shopping cart display when it is inserted into the page.

For more information on displaying individual charge and discount line items in the shopping cart display, see [Step 3 of 3: Set Discounts and Charges display](#) in this section.

For more information on configuring merchandising rules for use in the shopping cart, see [Discounts, Charges, Taxes and Shipping](#) in the Understanding the eCart Object section.

To access the eCart Display Manager:

1. In Design View in Dreamweaver, place your cursor in the location on your page where the shopping cart display is to be inserted.
2. Access the eCart Display Manager through one of the following methods:
 - eCart Insert panel



- Insert > WebAssist > eCart > Display Manager

The [next page](#) details the first step in configuring your display using the eCart Display Manager.

Step 1 of 3: Configure the design of your cart

The first step in configuring your shopping cart display using the eCart Display Manager is to establish the design of the display. Selections made in this step update the Preview pane below the configuration options to accurately reflect how the display will appear when inserted into the page and browsed.

The following is a breakdown of each of the configuration settings available and how they affect the shopping cart display. When you have made your selections, click the Next button to proceed to the next step:

- **Design:** Selects the design for the shopping cart layout. This configures the HTML structure and CSS used in the shopping cart display.
- **Display type:** Selects the type of shopping cart display to be inserted into the page. The following three options are available:
 - Updateable Cart: The most commonly-used display type. It inserts a complete shopping cart display containing all the items stored in the cart, charges, discounts, cart totals, and button controls available for use with the display.
 - Read-Only Cart: Displays all the same information as the Updateable Cart, but does not insert updateable fields for shopping cart line items.
 - Email-friendly cart: Places all CSS references in-line and does not include any images, to allow you to readily add the display to an email e.g. for order receipts
 - Quick Cart: Creates a scaled down display that only provides minimal cart details and options. It is intended for insertion in pages across the site as a quick reference to the current state of the cart and provide high level shopping cart functionality.

Only the cart total and the number of items in the cart is displayed. It also includes a View Cart button to redirect to a complete shopping cart display page, a Clear Cart button to empty the cart, and a Checkout button.

Note: If creating a Quick Cart display, this is the only configuration step necessary to add the display to the page.

- **Color scheme:** Selects the color scheme applied to the shopping cart display from 11 available options.
- **Button type:** Selects the type of buttons to be used in the shopping cart display. Options are:
 - Image: uses image buttons that correspond to the selected Style.
 - Form: uses form buttons

Note: The PNG source files for all buttons are available in your Download Center. These files are available to modify the buttons provided in the shopping cart display using the Edit with Fireworks feature in Dreamweaver. See the Buttons section under Shopping cart display components for more information.

- **Font:** Selects the default display font family for html text in your shopping cart.

eCart Display Manager

Step 1 of 3: Configure the design of your cart

Use these settings to configure the design of your cart. The preview reflects your design choices, but not your specific cart columns, discount or charges.

Settings

Design: Simple Display type: Updateable cart
Color scheme: Pacifica Button type: Image
Font: Arial, Helvetica, sans-serif

Preview

Your Shopping Cart

Name	Description	Price	Quantity	Remove	Total
Item One	This is a description for item one	\$69.99	1	<input type="checkbox"/>	\$69.99
Item Two	This is a description for item two	\$42.00	2	<input type="checkbox"/>	\$84.00

[Continue Shopping](#) [Clear Cart](#) [Update](#)

Order Summary

Sub-total	\$153.99
Discounts	-\$15.40
Charges	\$3.00
Shipping	\$26.00
Tax	\$11.37
Total:	\$178.46

[Checkout](#)

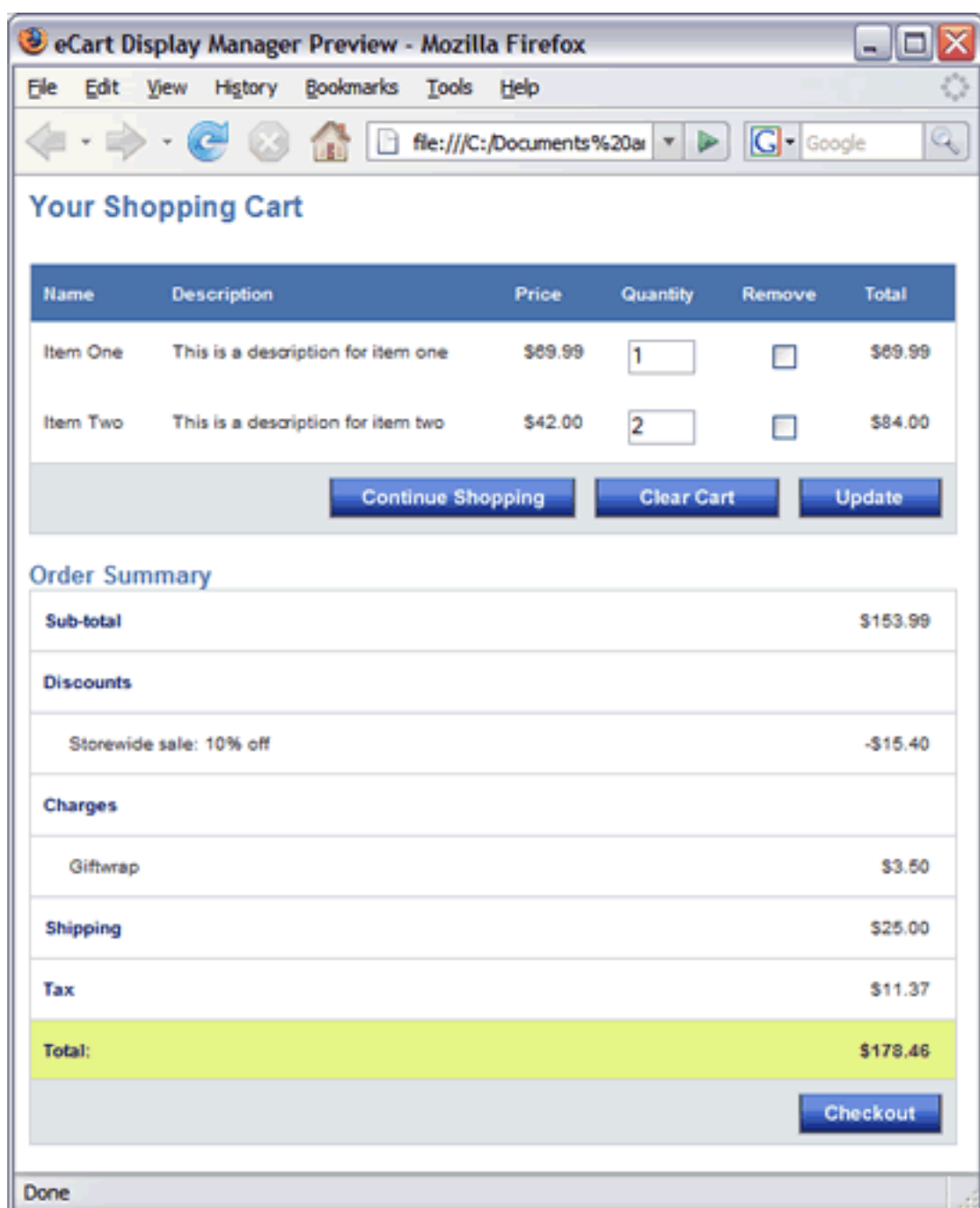
WEBASSIST [HELP](#) [← BACK](#) [NEXT ►](#) [FINISH](#) [CANCEL](#)

Step 2 of 3: Configure the columns of your cart

The second step in configuring your shopping cart display using the eCart Display Manager is to select the shopping cart object and corresponding cart columns to be displayed within your shopping cart.

Shopping cart columns are the individual details associated to an item when it is added to, or updated in, the shopping cart. For example, the below shopping cart display shows the following columns:

Name, Description, Price, Quantity Edit, Remove Checkbox, and Total



Common cart columns are the ID, Name, and Price of an item in the shopping cart. These populate the shopping cart when an item is added with information that is unchanging for the purposes of the shopping cart.

Columns can also contain calculated information based on other data within the cart (e.g. The Price of an item multiplied by the Quantity of an item = TotalPrice of an item).

Columns can also contain fields that allow you to update the state of an item that has been added to the shopping cart. These are used in conjunction with the Update Cart button to make changes to the state of the cart.

Common examples of updateable fields are the Quantity Edit field and the Delete checkbox. By default, these updateable fields are available for configuration directly within the eCart Display Manager.

Custom updateable fields can be added as well once a shopping cart display has been created (for example, replacing the display text for the Size column with an Updateable select list to select a Size option). See [Updateable form fields](#) for more information.

Available cart columns are configured within the eCart Object specific to your shopping cart implementation. For more information regarding the configuration of shopping cart columns, please see [Columns](#) in the Understanding the eCart Object section.

Configuring display columns

If more than one ecart Object is applied to the page, select the eCart Object that corresponds to the shopping cart display you are configuring.

To select the columns you wish to be included in the shopping cart display, select a column using the Add (+) button to add it to the display pane.

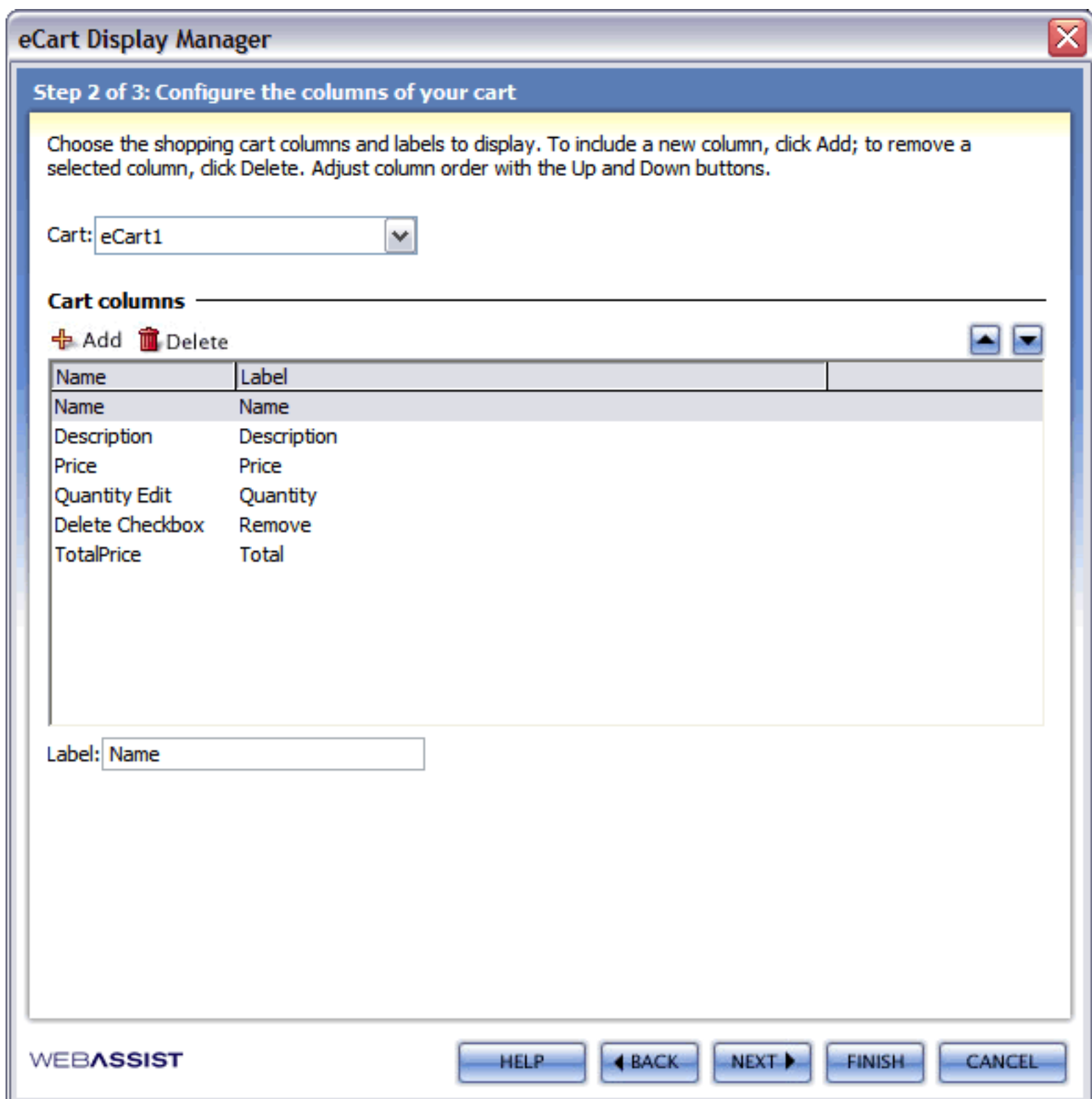
To remove a column from the columns to be displayed, use the Delete button to remove a selected column from the display pane.

The order that the columns are listed in the *Display columns* list represents the order that they will be created left-to-right within the shopping cart display.

The Up and Down arrow buttons will move a column selected in the *Display columns* list up or down one position in the hierarchy.

When you have completed configuration, click the Next button to proceed to the next step.

The text displayed to identify a column can be configured by changing the text in the Label field for a selected column.



Step 3 of 3: Set Discounts and Charges to display

The third step in configuring your shopping cart display using the eCart Display Manager is to configure how charges and discounts applied to the shopping cart are displayed.

Discounts and charges are additional line items applied to the shopping cart. Different than items added to the cart, they are calculated based on merchandising rules configured within the eCart Object specific to your shopping cart implementation.

Merchandising rules calculate a charge or a discount based on a condition satisfied within the state of the shopping cart. Common rules apply shipping and sales tax charges, or apply discounts for bulk purchases of a given item.

Shipping and tax, additional specific discount types, are available to display as distinct sections and line items within the shopping cart display.

The name given to a rule is used as the display text for the given charge or discount applied in the cart that the rule is associated to.

Only rules that have an active status in the eCart Object configuration during configuration of the shopping cart display are available to be included in the shopping cart display as line items.

For more information on creating discounts and charges within the shopping cart, see [Discounts, Charges, Taxes and Shipping](#) in the Understanding the eCart Object section.

Configuring the display of charges, discounts, shipping, and tax

Each rule is available for display as a line item in the shopping cart display.

Four classes of discounts and charges are available for display: Discounts, Charges, Shipping, and Tax

For each of these sections, you have the option of displaying a summary of all charges for the section, individual line items within each section, or disabling that section in the display.

Selecting the summary option includes the section header and a corresponding total for the rules applicable. Individual rules are not included as line items in the display for this option.

Selecting the individual option displays a line item for each rule triggered in the shopping cart, as well as the sum for the section.

Note: Even if an active charge or discount is applied to the cart, if display is not enabled, that section will not be displayed. The cart grand total will still be affected by the rule, which may cause the displayed section totals to not appear to sum to the displayed grand total number.

When you have completed configuration, click the Next button to proceed to the Review selections portion of the wizard and application of your configured display to the page.

eCart Display Manager

Step 3 of 3: Set Discounts and Charges display

Determine how your discounts and charges should be displayed, if at all. The preview reflects your design choices, but not your specific cart columns, discount or charges.

Display options

Discounts: Summary Shipping: Summary
Charges: Summary Tax: Summary

Preview

Your Shopping Cart

Name	Description	Price	Quantity	Remove	Total
Item One	This is a description for item one	\$69.99	1	<input type="checkbox"/>	\$69.99
Item Two	This is a description for item two	\$42.00	2	<input type="checkbox"/>	\$84.00

Continue Shopping Clear Cart Update

Order Summary

Sub-total	\$153.99
Discounts	-\$15.40
Charges	\$3.50
Shipping	\$25.00
Tax	\$11.37
Total:	\$178.46

Checkout

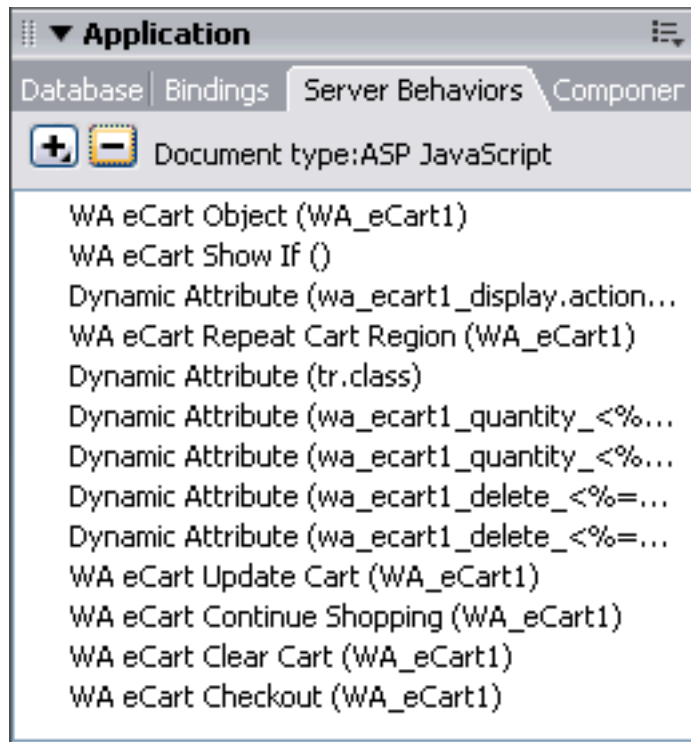
WEBASSIST HELP BACK NEXT FINISH CANCEL

Completing display configuration

Upon completing the insert of your shopping cart display using the wizard, you will need to modify the shopping cart buttons included in the display to redirect to the desired location when clicked.

Each of these buttons have corresponding server behaviors that contain this configuration information. Each was applied to the page when the shopping cart display was inserted by the eCart Display Manager.

Updates can be made to each button configuration by accessing and modifying the corresponding server behaviors through the server behaviors panel (see the last four entries in the following figure).



When the buttons are inserted, their redirect location is left blank, which simply redirects the page to itself when the button is clicked.

For the Update Cart button, redirecting to the same page is the most common scenario as the user will want to see the changes reflected in the cart that the Update button is submitting.

For the Continue Shopping button, you may wish to specify a product catalog landing page or enter server-side code that takes them to a location determined dynamically.

For the Clear Cart button, it is common to redirect to the same page so the user can see that the cart has been cleared of its contents.

For the Checkout button, it is common to redirect to a page that has the checkout form to begin payment processing.

Note: the Checkout button can also be removed and replaced with a checkout form configured using eCart Insert Checkout Form. See [Configuring checkout within the shopping cart display](#) for more information on redirecting directly to a merchant provider's checkout form.

You should note that all the images included in the shopping cart display have been inserted into an images folder in the eCart folder at the root of your site.

As well, a CSS file containing the guidelines for the design of your shopping cart display has been included in a CSS folder within the eCart folder. You can use the CSS panel for the styles included in this file to incorporate similar design elements throughout your website.

Note: Email-friendly displays will have the css included in-line, and no images are included in the shopping cart display.

The next section, [Shopping cart display components](#), details the individual components that comprise a shopping cart display. This information proves useful if creating a shopping cart display from the ground up, or modifying the features of an existing shopping cart display.

Shopping cart display components

Shopping cart displays are comprised of several components that work together to form a fully functional shopping cart.

The [eCart Display Manager](#) makes the process of configuring and creating your shopping cart display a simple process by applying all the necessary components for the display in a single insert, including the HTML structure and any necessary server-side code.

Once the display has been created in this manner, users have the option of further customizing how the shopping cart display works by modifying the HTML directly in Dreamweaver's WYSIWYG design view, as well as modifying the server-side code through the server behaviors and bindings provided.

Advanced users can also create their own HTML structure and build a shopping cart display from the ground up, specific to their own design needs, using the provided components.

The following sections detail all the shopping cart display components provided in this extension to assist in their configuration and application. Information specific to how they are used in the context of the display, as well as specifics on their configuration, are provided in detail:

- [Buttons](#)
 - [Data bindings](#)
 - [Updateable form fields](#)
 - [Show If server behavior](#)
 - [Redirect If Empty server behavior](#)
 - [Repeat Cart Region server behavior](#)
 - [Rules Repeater](#)
-

Buttons

There are 5 eCart buttons available for inclusion in a shopping cart display. Each has a specific function within the shopping cart, but they are almost identical in their method of configuration and application:

- Update Cart Button:** Applies changes to the shopping cart that have been made directly within the shopping cart display using updateable form fields (e.g. editing the Quantity of an item). It typically redirects to itself in order to display the changes made to the shopping cart to the user.
Note: the shopping cart display is capable of advanced update configuration using the provided Updateable form fields. See [Updateable form fields](#) in this section for more information.
- Continue Shopping Button:** Redirects to a specified location to continue the shopping experience. Typically this is configured to redirect to a product catalog page. Can be configured directly to return to the page that initially directed the user to the shopping cart display page.
- Clear Cart Button:** Removes all items currently in the shopping cart. The redirect location is usually configured to return to the shopping cart display page to display the fact that the shopping cart is empty to the end-user of the site.
- Checkout Button:** Redirects to a specified location to initiate the checkout process. This is typically configured to redirect to a location that contains a checkout form.

Note: The eCart Insert Checkout Form feature configures a checkout form to submit transaction and payment data to the merchant gateway provider to complete the transaction.

All merchant providers supported directly within eCart have a hosted form that gathers this information on their website.

If you don't require collecting any customer information on your website prior to checkout, it is possible to replace this checkout button with a checkout form using this feature. The form will only pass the necessary account information to complete the transaction to the merchant gateway using hidden form elements.

This means its possible to checkout directly from the shopping cart display.

See Configuring checkout within the shopping cart display in the Configuring payment gateway support section of this documentation for more information.

- View Cart Button:** Redirects to a specified location to view the shopping cart display. This is a simple redirect button insert, and is the only button provided that has no server behavior component. It is only inserted by the eCart Display Manager for Quick Cart shopping cart displays.

Feature access

Each of these buttons can be inserted from the following locations:

- eCart Insert panel:



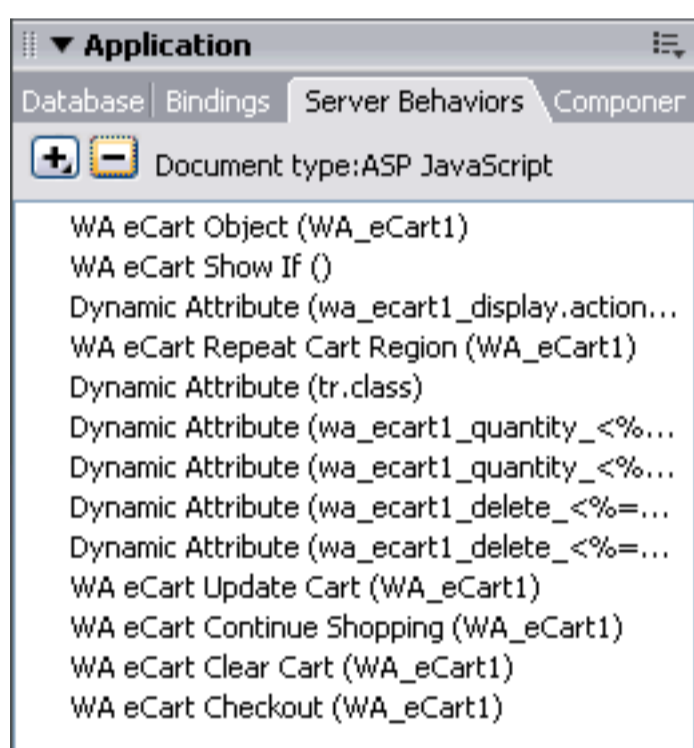
- Insert > WebAssist > eCart > [Name] Button
- Server Behaviors > eCart > Cart Display > [Name] Button
Server Behaviors > eCart > Checkout > Checkout Redirect Button

Note: the View Cart Button does not have a corresponding server behavior and, therefore, is not available for insert or configuration through the server behaviors panel.

Editing button configurations

With the exception of the View Cart Button, each of these buttons has a corresponding server behavior that is applied to the page when the button is inserted.

When modifying a button configuration, access the configuration user interface by editing the applied server behavior through the server behaviors panel (Note the last four entries in the below figure):

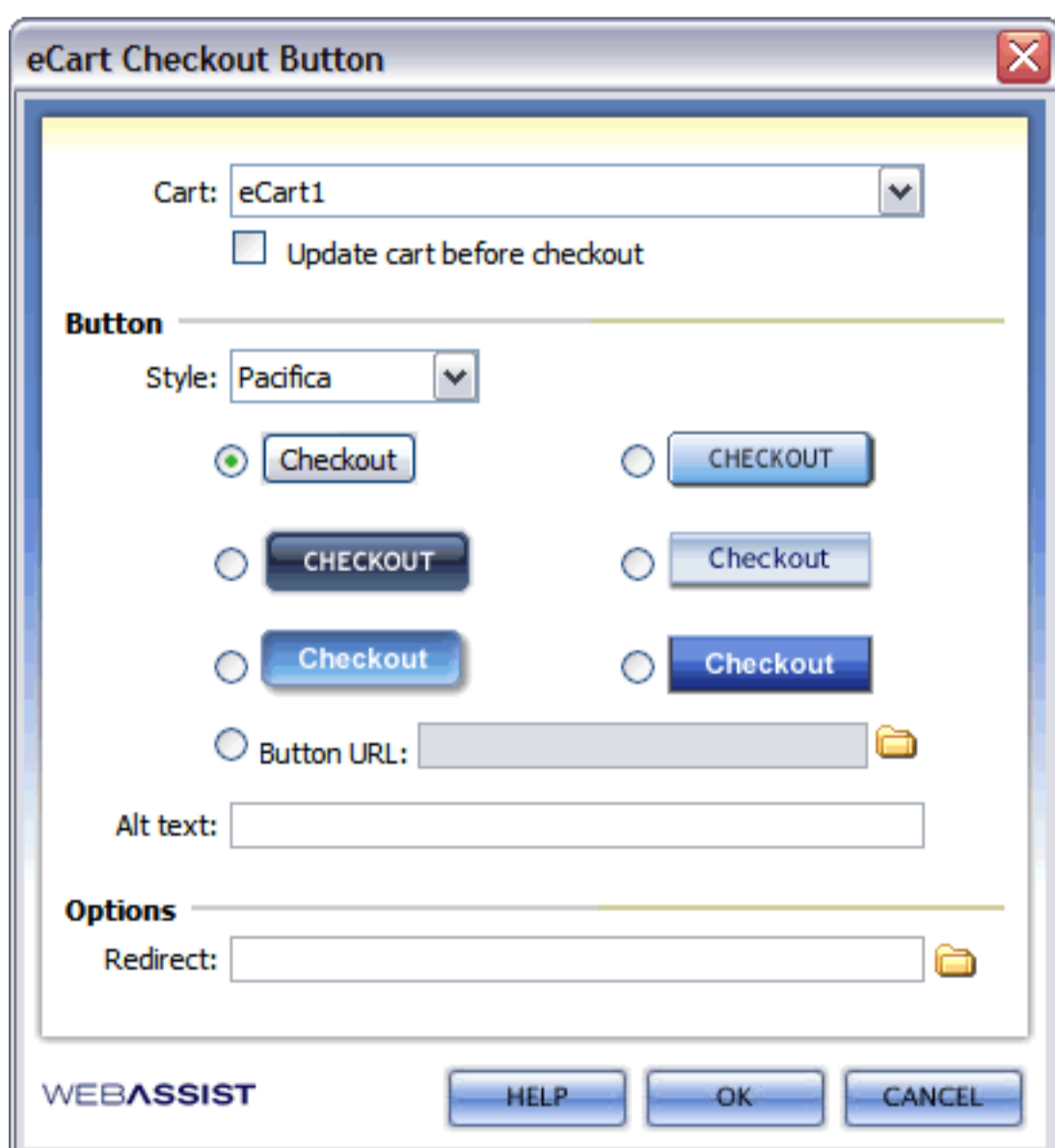


In the case of the View Cart Button, you may modify its properties after application by selecting it in Design View on the page, and using the corresponding image property inspector to make any necessary modifications.

To configure a button for your display:

- In Design View, place your cursor in the location where the button is to be inserted.
- In the eCart Insert panel, click the icon that corresponds to the button to be inserted (see Feature access above).

The button configuration dialog will open specific to the button type. The figure below shows the Checkout Button configuration dialog:



- From the Cart list, select the name of the eCart object that corresponds to your shopping cart implementation.
- Update cart before checkout* : Select whether the any changes made to the shopping cart should be updated prior to initiating the Checkout process.
- Select the Style of the button. This selects the color scheme for the image buttons available, and updates the images shown in the dialog to that color scheme to assist in your selection.
- Specify the button to be placed on the page. You have the option of selecting from a form button or several image buttons provided with the extension, as well as specifying an image button that you provide. The Button URL field will accept an http location or a relative location of an image file within your site.
- Specify the destination page which is redirected to upon click of the button. Use the Browse button to select a file within your site or specify an http:// location. Server-side code can also be entered in this field to dynamically retrieve a location for the redirect.
- Enter text or server-side code to specify the alt text which is displayed on the rollover event for the button. Leaving this blank removes the alt text attribute for the button.
- Click OK to apply the button configuration to the page.

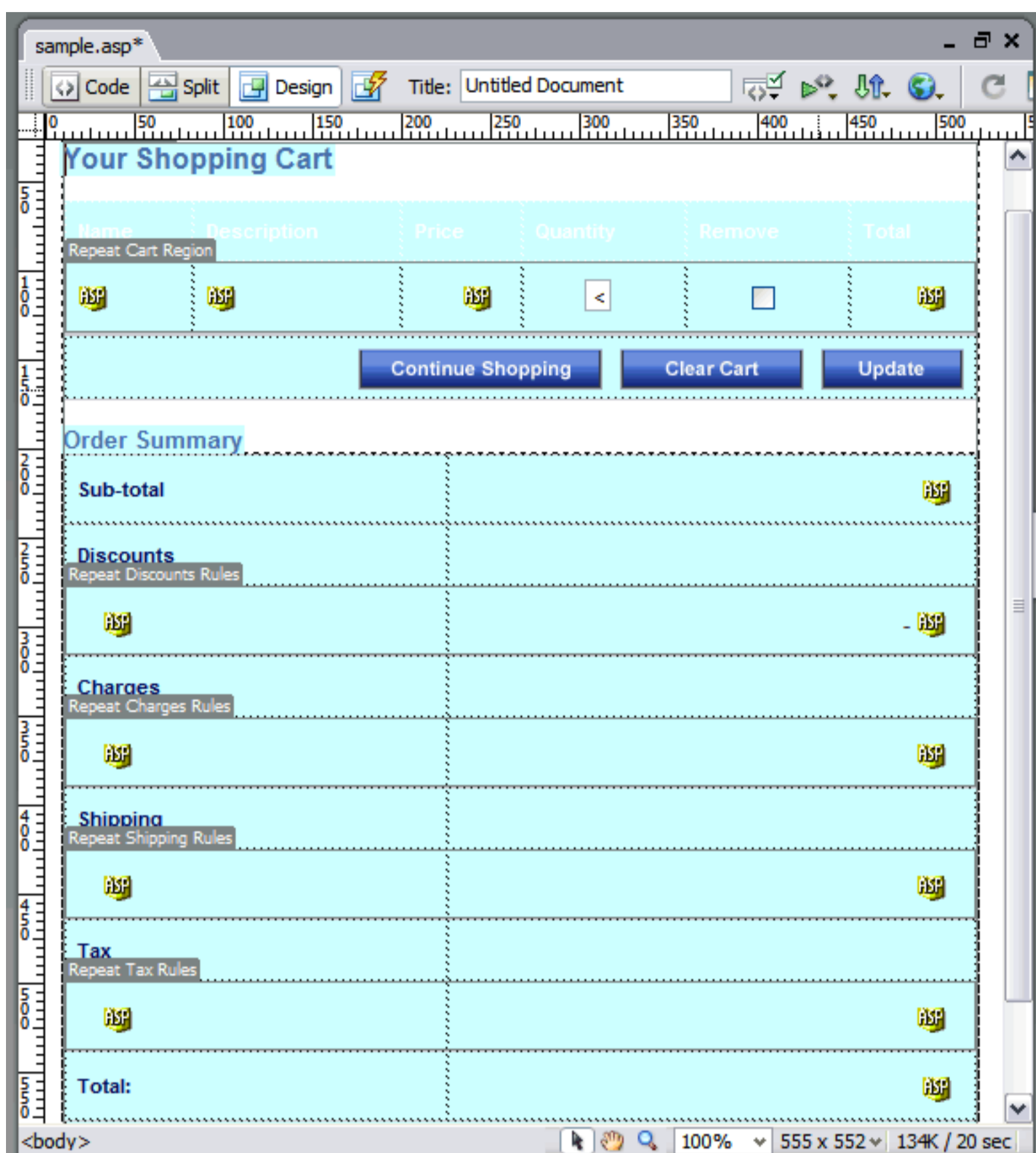
Data bindings

Server-side code is used to retrieve information used by the shopping cart. This information is available in the Bindings panel to either display that information, or use it to define a value for a form element within the shopping cart.

If you created a shopping cart display using the eCart Display Manager, this server-side code was automatically placed on the page.

If you create a shopping cart display without using the eCart Display Manager, or need modify an existing shopping cart display that has been created, the dynamic elements that specify information within the shopping cart need to be applied using the Bindings panel.

The figure below illustrates the information used in the shopping cart that is specified in the display using server-side code, specifically ASP code in this example.

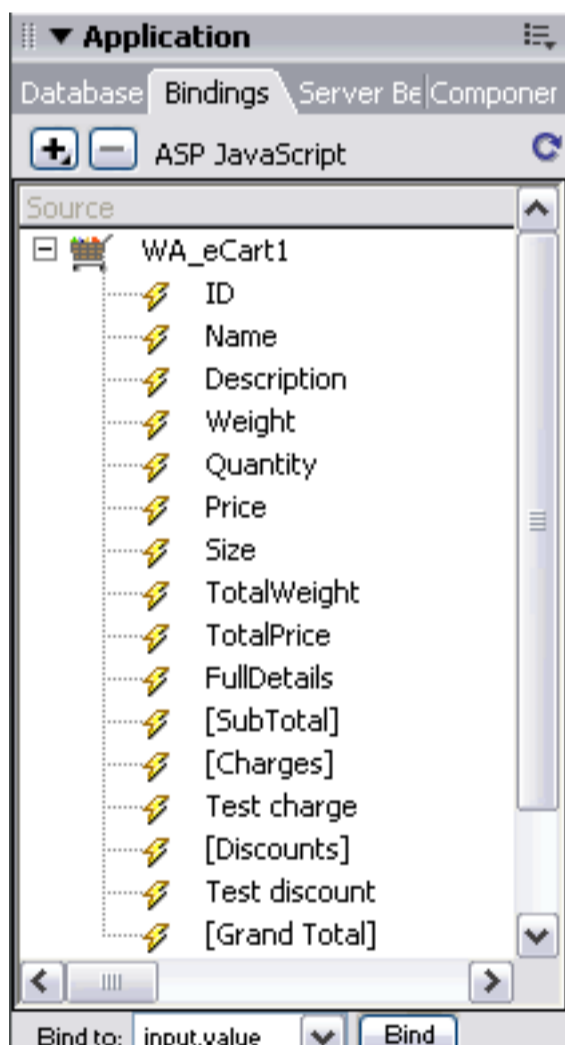


Within the table row that displays contents of the shopping cart, you'll notice that each piece of information displayed within each cart column uses a server-side code reference.

Updateable fields, such as the Quantity and Delete checkbox may not visibly show the server-side code, but their values are defined by the server-side code specific to their cart column.

In addition, you'll also notice that the subtotal, charges, discounts, and total of the shopping cart are also server-side code references.

The below figure illustrates the Bindings panel for the above shopping cart display. All of the server-side code referenced above is available as individual bindings for the eCart Object associated with the above display.



As you can see, all of the shopping cart columns are available as bindings (ID through Size), as well as the calculations (TotalWeight, TotalPrice, FullDetails) and individual merchandising rules (Test charge, Test discount) configured for the given eCart Object.

In addition, bindings specific to the shopping cart display are available and are referenced in brackets ([]). These include, SubTotal, Charges, Discounts, and GrandTotal.

Having familiarized yourself with the concept of data bindings, you can now use the Bindings panel to insert this information for display on the page, or bind that information to attributes of page elements, such as the default value of a given form element.

To insert a data binding into the page for display:

1. In Design View, place your cursor in the location where you want the information for a data binding displayed.
2. In the Bindings panel, select the specific data binding to be inserted.
3. At the bottom of the Bindings panel, click the Insert button.
4. In Design View, the server-side code image will appear in the insert location.

To bind data to an attribute of a page element:

1. In Design View, select an element of the page that has an attribute you wish to bind shopping cart data to.
e.g. You wish to set the default value of a textfield to the Quantity of an item in the shopping cart. Select the given textfield.
2. In the Bindings panel, select the specific data binding you wish to set the given element's attribute to.
e.g. Select the Quantity data binding.
3. At the bottom of the Bindings panel, select the attribute for the given page element from the list of available attributes in the Bind to: list.
e.g. Select the *input.value* option from the list.
4. Click the Bind button.
5. In Design View, select the page element and look at the property inspector for the element. The attribute selected in Step 3 should reflect the server-side code for the given data binding.
e.g. The default value in the property inspector for the textfield should reflect the server-side code for the Quantity binding.

In this manner, you can update an existing shopping cart display with data available in the shopping cart, as well as build a shopping cart display from the ground up without using the eCart Display Manager.

Updateable form fields

The shopping cart display not only displays the data for an item in the shopping cart, but it provides the ability to update information about items in the shopping cart as well.

The Update Cart button in the shopping cart display has a server behavior component that checks the state of form fields in the shopping cart display, and applies any changes to the contents of the shopping cart.

By default, the eCart Display Manager allows you to include two common updateable fields as columns in the shopping cart display. These are the Quantity Edit field, and the Delete checkbox.

The Quantity Edit field allows the user to update the number of a specific item within the shopping cart.

The Delete checkbox removes an item from the shopping cart when checked.

It is possible to configure your shopping cart display to use additional updateable form fields as your needs dictate. For example, you may wish a customer to be able to change an option such as size for a given item in the shopping cart.

eCart provides three types of updateable form fields for use within the shopping cart display. They are:

- [Updateable Checkbox](#)
- [Updateable Text Field](#)
- [Updateable List](#)

The Delete Checkbox and the Quantity Edit field are common use cases of the Updateable Checkbox and Updateable Text Field, respectively.

You may use these updateable form fields to accept input from the customer, and then apply their changes to the shopping cart using the Update button.

Feature access

Updateable form fields can be inserted from the following locations:

- The eCart Insert panel:



- Insert > WebAssist > eCart > Updateable [form field]

Configuring the shopping cart display to use an updateable form field

For an updateable form field to be included in the shopping cart display, the display must first have a column configured specific to the information for the given item. This means the column must be configured within the eCart Object prior to creating the shopping cart display.

As stated above, the Quantity Edit and Delete Checkbox are available directly for configuration within the eCart Display Manager for use in creating a shopping cart display. These two special cases do not require that a column be defined within the eCart Object.

When creating the shopping cart display, the column associated to the updateable field needs to be included in the shopping cart display.

If using the eCart Display Manager to create the shopping cart display, you will not have the option of specifying that a given column is to contain an updateable form field. Therefore, when the shopping cart display is inserted, the value for that column will be display text.

Once the display has been created, you may remove the display text for the given column, and replace it with the appropriate updateable form field specific to your needs.

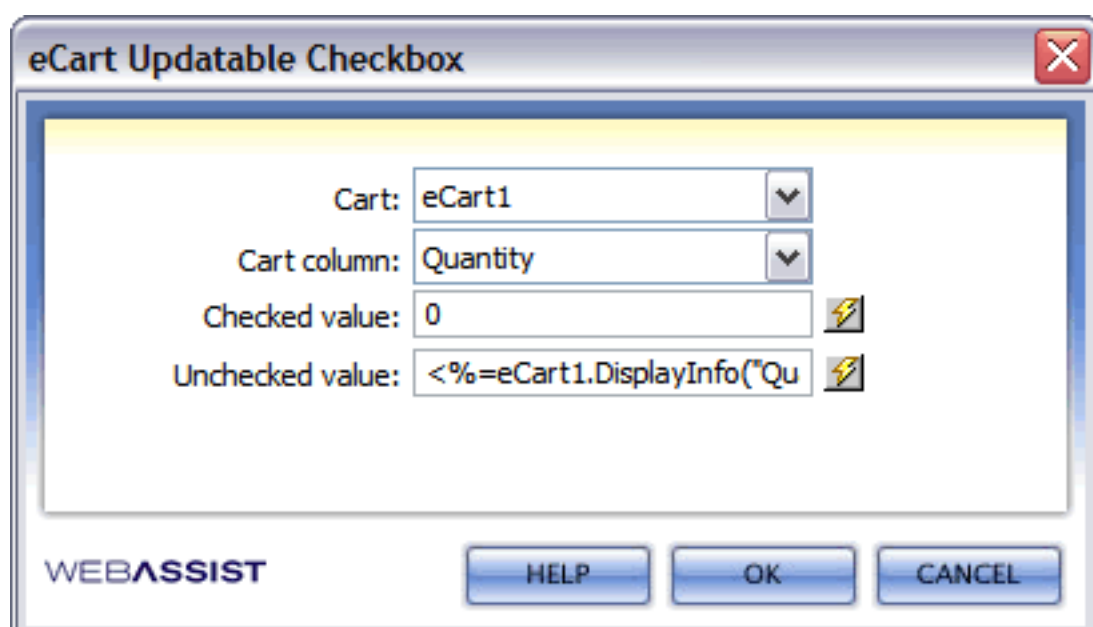
The following sections detail the steps for inserting each type of updateable form field into a shopping cart display and configuring it to update the appropriate shopping cart column.

Note: The configuration interfaces for each updateable form field can only be used to insert the given field and cannot be used to modify configuration after insertion. Unless you are comfortable with modifying the code directly, please delete and apply the given field to the page again to ensure that configuration changes are made correctly.

To apply an Updateable Checkbox to your shopping cart display:

1. In Design View, place your cursor in the shopping cart display where you wish to insert the updateable checkbox.
2. In the eCart Insert panel, click the eCart Updateable Checkbox icon.

The eCart Updateable checkbox dialog opens:



By default, the dialog is configured to insert a Delete Checkbox.

Note that the Cart column to be updated is the Quantity column, and that a checked value updates this to the default value of 0, effectively removing the item from the shopping cart.

As well, note that the Unchecked value is a server-side code reference. This is a data binding that sets the value of the Quantity column to the current value of that column, effectively leaving the item unchanged in the cart.

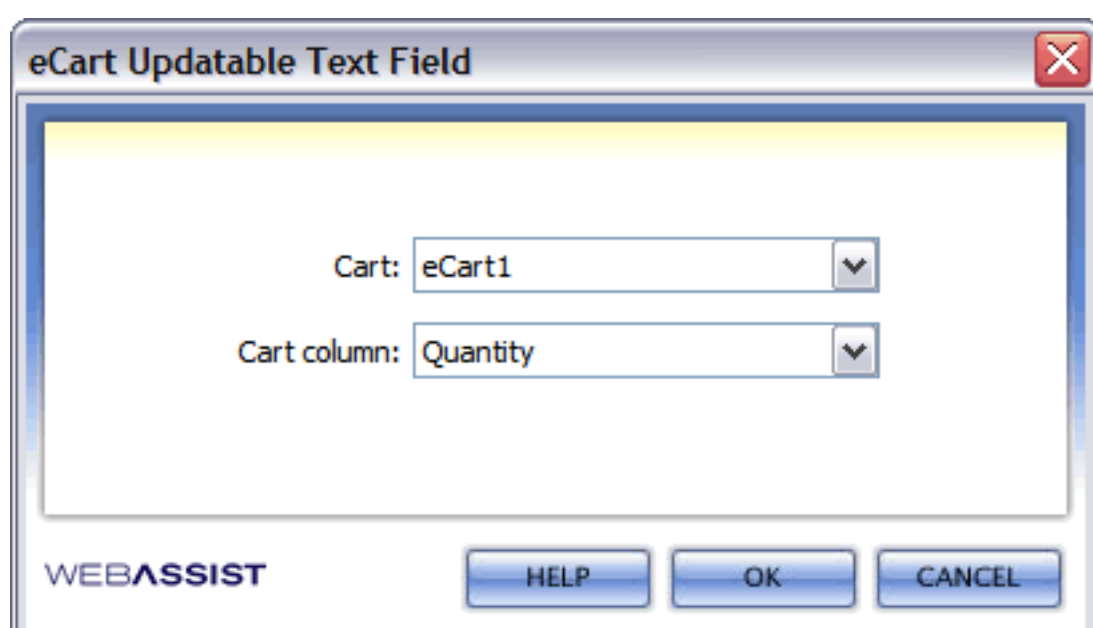
In this manner, you can set checked and unchecked values to either change or maintain the current state of any cart column.

3. Select the cart name specific to the eCart Object associated to the shopping cart display.
4. For your configuration, select the cart column whose value will be updated based on the state of the checkbox.
5. Specify a static value, or click the lightning bolt to specify a data binding, that will set the value of the selected column based on the state of the checkbox (checked or unchecked).
6. Click OK to insert the updateable checkbox into the shopping cart display.

To apply an Updateable Text Field to your shopping cart display:

1. In Design View, place your cursor in the shopping cart display where you wish to insert the updateable text field.
2. In the eCart Insert panel, click the eCart Updateable Text Field icon.

The eCart Updateable Text Field dialog opens:

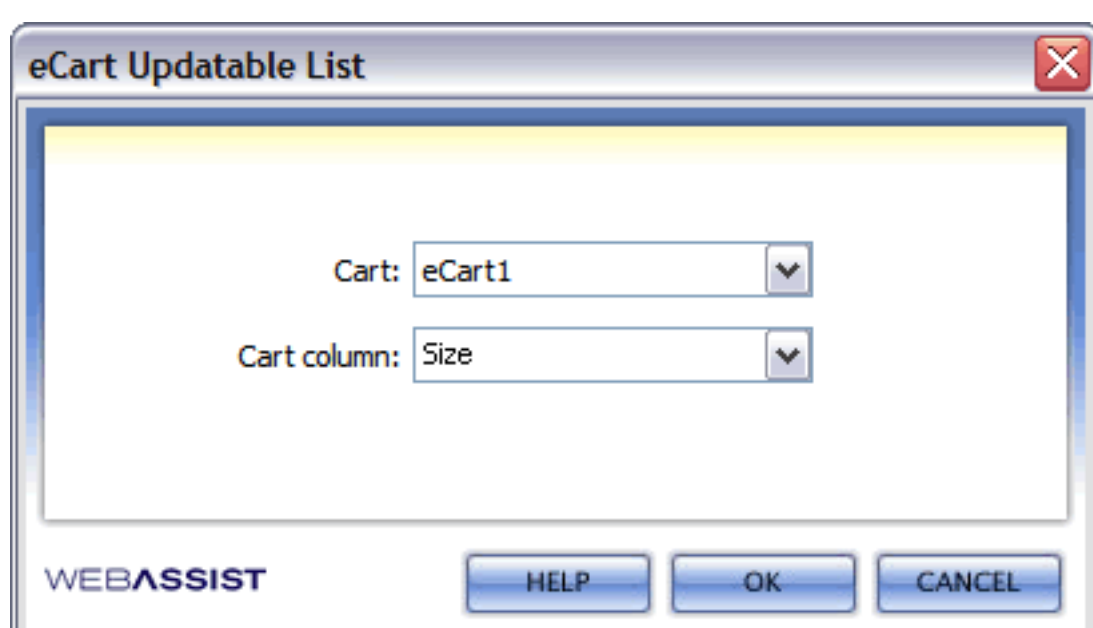


3. Select the Cart name for the eCart Object associated to the shopping cart display.
4. Select the shopping cart column to be updated for the given item.
5. Click OK to insert the updateable text field in the shopping cart display.

To apply an Updateable List to your shopping cart display:

1. In Design View, place your cursor in the shopping cart display where you wish to insert the updateable list.
2. In the eCart Insert panel, click the eCart Updateable List icon.

The eCart Updateable List dialog opens:



3. Select the Cart name for the eCart Object associated to the shopping cart display.
4. Select the shopping cart column to be updated for the given item.
5. Click OK to insert the updateable list in the shopping cart display.
6. In Design View, select the list on the page.
7. Using the property inspector, configure the options available for the list.

Show If server behavior

The Show If server behavior displays content on a page dependent on whether the cart is or is not empty.

It is applied to the shopping cart display automatically when it is created using the eCart Display Manager. The shopping cart is configured to display when the shopping cart is not empty.

As well, default HTML is provided, specific to the cart design style, that states "The cart is empty." This HTML is displayed when there are no items in the cart.

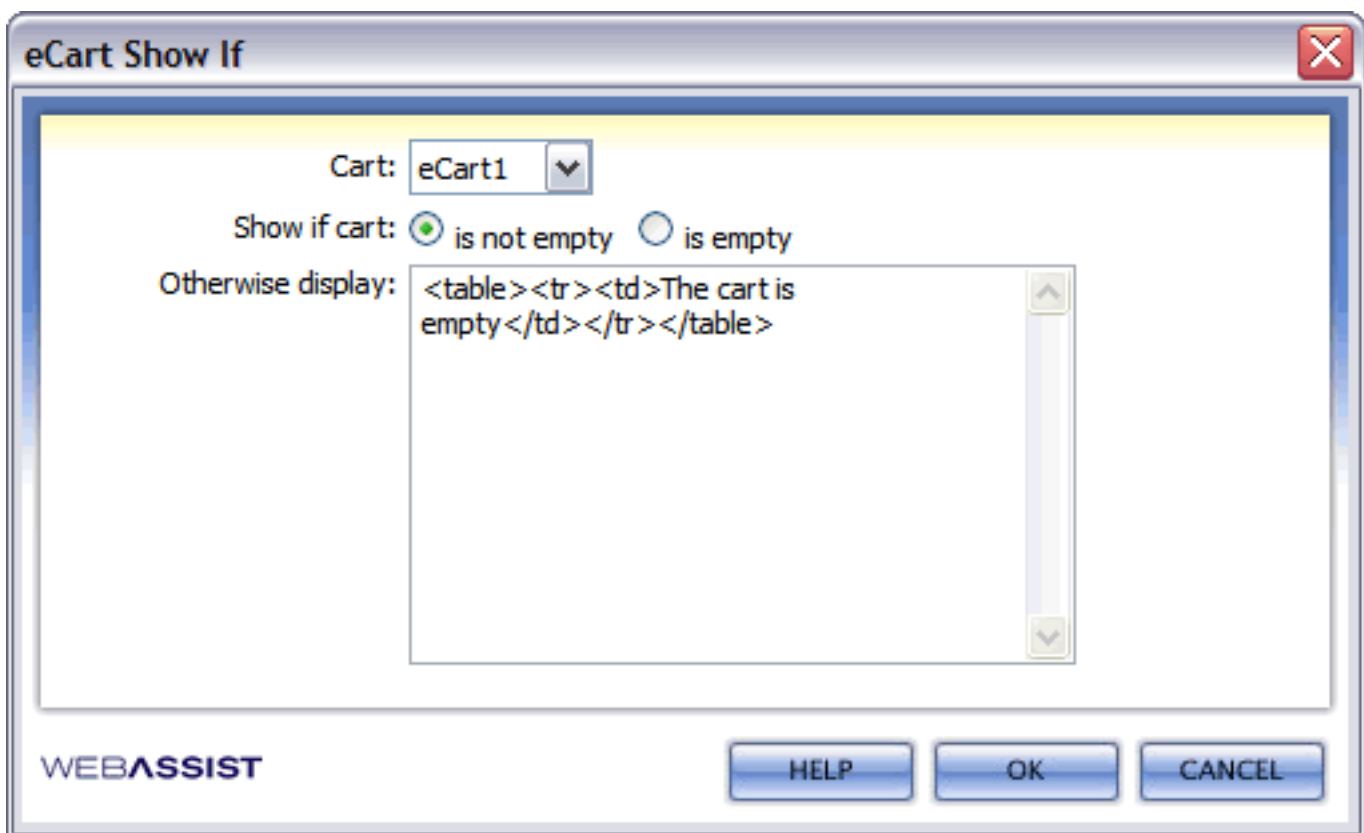
Accessing this server behavior for a shopping cart display created using the eCart Display Manager allows you to update this HTML displayed specific to your own needs.

This server behavior can also be applied to any specified HTML region, as well as server-side code. It proves useful for displaying a region conditional to the state of the shopping cart, such as:

- Displaying buttons - Display [Quick Cart display](#) buttons (e.g. checkout, view cart) only when there are items in the shopping cart
- Displaying text - display information to the user applicable to the whether the cart does or does not contain items (e.g. "Your current cart total is X." or "Your cart is empty.").

To apply the eCart Show If server behavior:

1. On your page, select the display area that will be controlled by the Show If server behavior.
2. In the Server Behaviors panel group, click the Plus button and select WA eCart > Cart Display > Show Region If. The following dialog box will open:



3. Select the Cart name for the eCart Object specific to your shopping cart implementation. This can only be changed if multiple eCart Objects for different shopping cart implementations are applied to the page. .
4. Select whether you wish the selected display area to be shown if the cart is empty, or is not empty.
5. Specify the HTML or server-side code to be displayed/run if the state of the cart does not correspond to your selection in step 3.

If you leave this field blank, the selected display region will not be replaced.

Note: Ensure that the display area to which the server behavior is applied coincides with the format used in this field. For example, if you selected a `<td>` tag as the display area, be sure the replacement information contains a `<td>` tag to ensure that the HTML is rendered correctly.

6. Click OK to apply the server behavior to the page.
-

Redirect If Empty server behavior

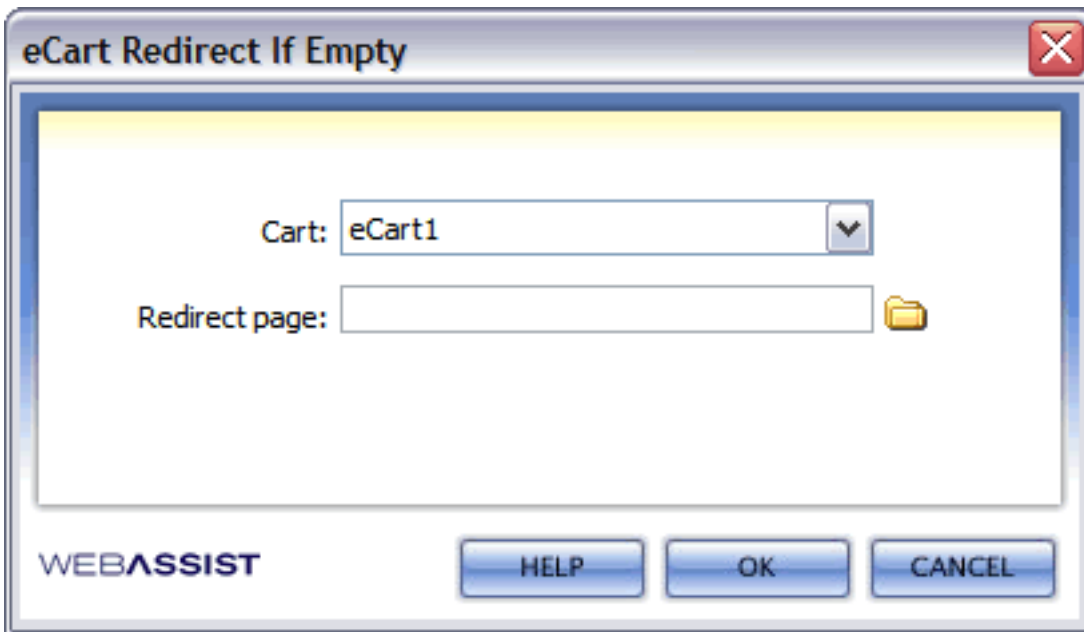
The eCart Redirect if Empty server behavior sets a default location to redirect to if the shopping cart is empty.

It is most commonly used on a shopping cart display page in place of the [Show If server behavior](#), which displays specified HTML in place of the shopping cart when it is empty.

To apply the eCart Redirect if Empty server behavior:

1. In the Server Behaviors panel group, click the plus button and select WA eCart > General > Redirect if Empty.

The eCart Redirect if Empty dialog box opens.

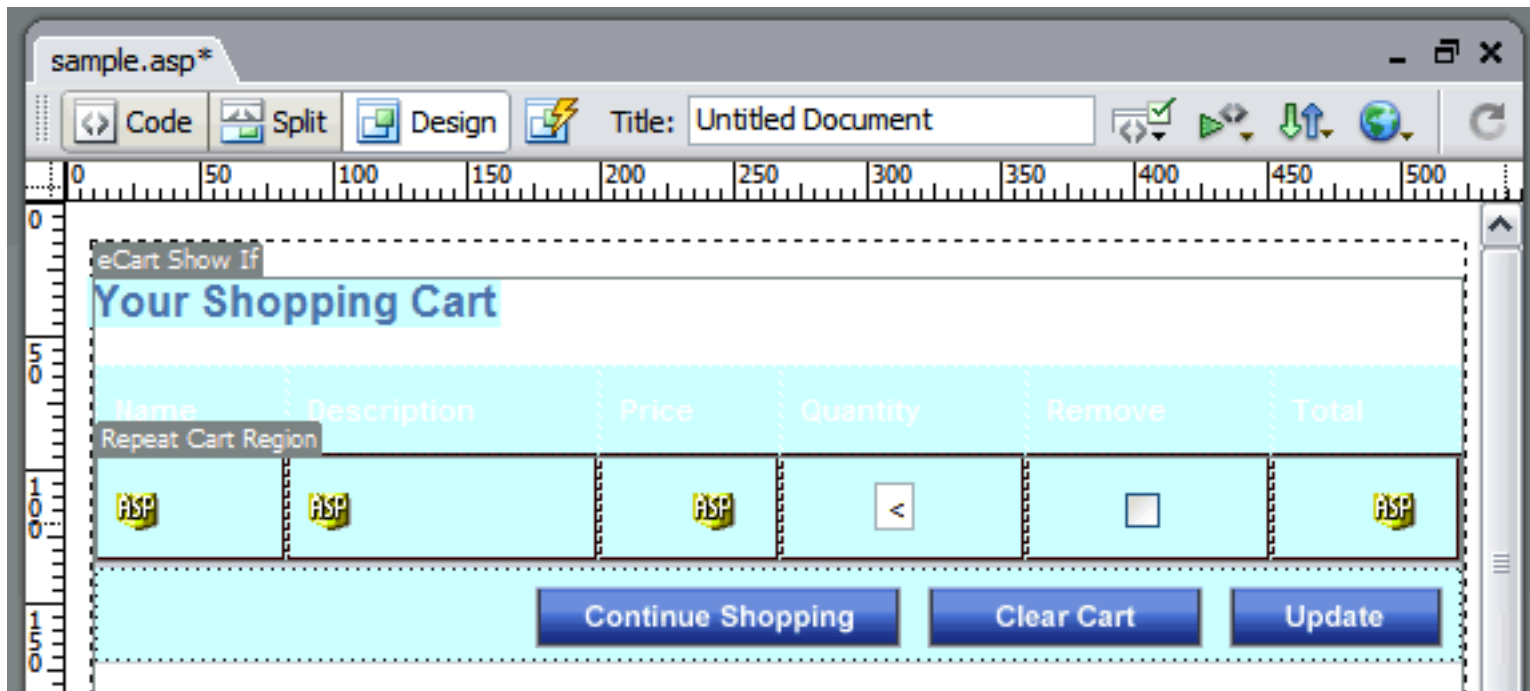


2. Select the eCart object applied to the page specific to the shopping cart implementation that will trigger this server behavior.
(i.e. the shopping cart implementation that will be empty).
 3. Specify the redirect location either by browsing to the page within the context of your site, or by specifying an http:// address.
 4. Click OK to apply the server behavior to the page.
-

Repeat Cart Region server behavior

The Repeat Cart Region server behavior is applied to the table row (<tr>) that displays the individual items contained in the shopping cart.

The figure below shows the Repeat Cart Region server behavior as it is applied to the line item table row:



The server behavior parses the data containing the shopping cart contents and displays the data for each shopping cart column displayed specific to that item. As you can see, each column contains server-side code that retrieves the specific column data for a given item.

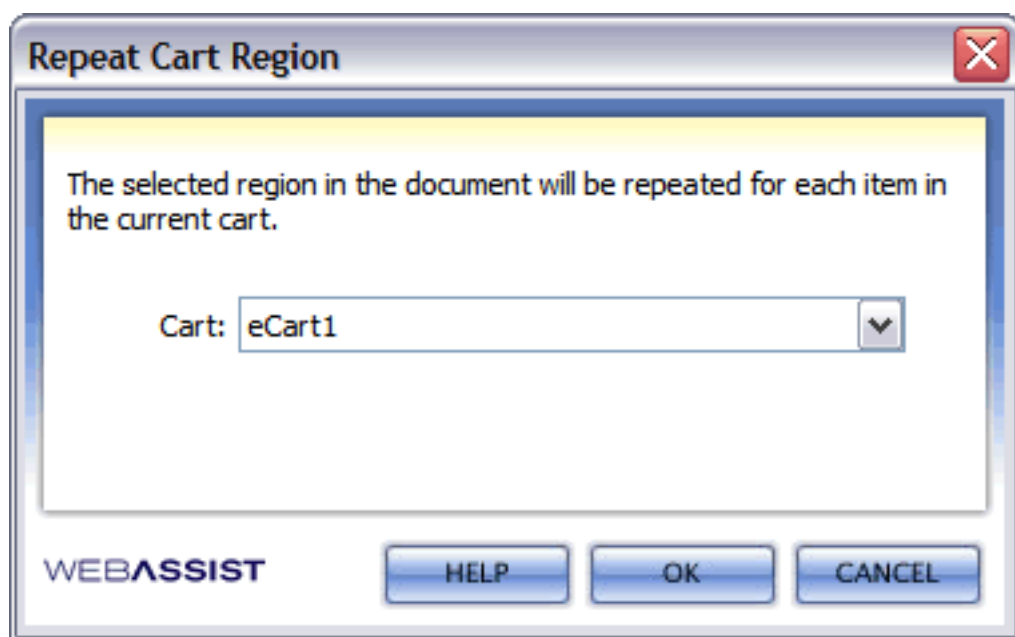
Note: The server-side code for each column is available for insert as display text, or to be bound to the default value of a given form element using the Bindings panel (see [Bindings](#) in this section for more information).

The first item in the cart is displayed, and it repeats the code contained in the table row for each additional item currently in the shopping cart.

To apply the Repeat Cart Region server behavior:

1. In Code view, select the entire table row (<tr>content</tr>) within your display specific to the shopping cart contents line items.
2. In the Server Behaviors panel group, click the plus button and select WA eCart > Cart Display > Repeat Cart Region.

The Repeat Cart Region dialog box opens.

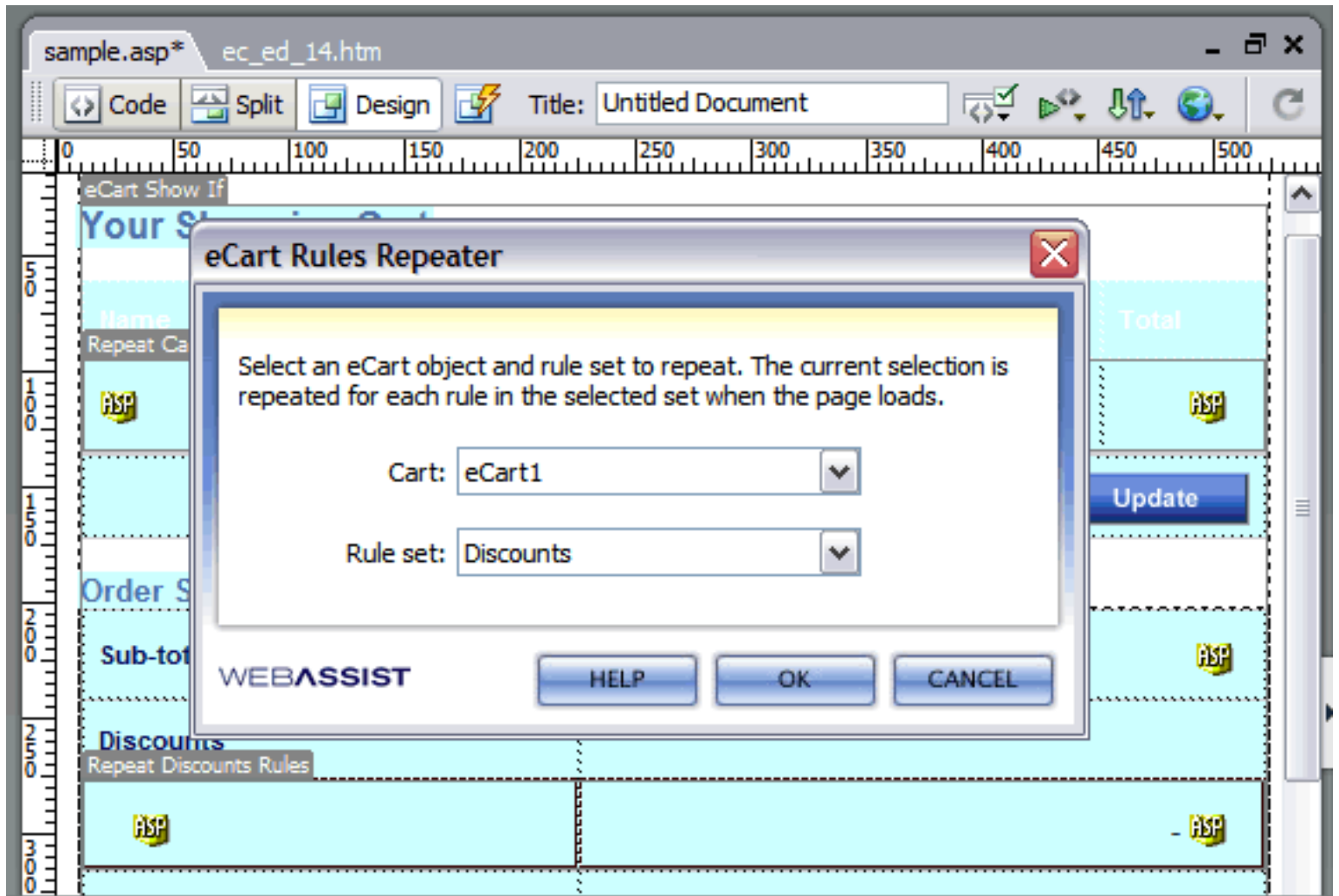


3. Select the cart name that corresponds to the eCart Object specific to the shopping cart implementation for the given display.
4. Click OK to apply the server behavior to the page.

Rules Repeater server behavior

The Rules Repeater server behavior is applied to the table row (<tr>) that displays the one of the four available rules display sections: Discounts, Charges, Shipping and Tax.

The figure below shows the Rules Repeater server behavior as it is applied to the discount rules table row:



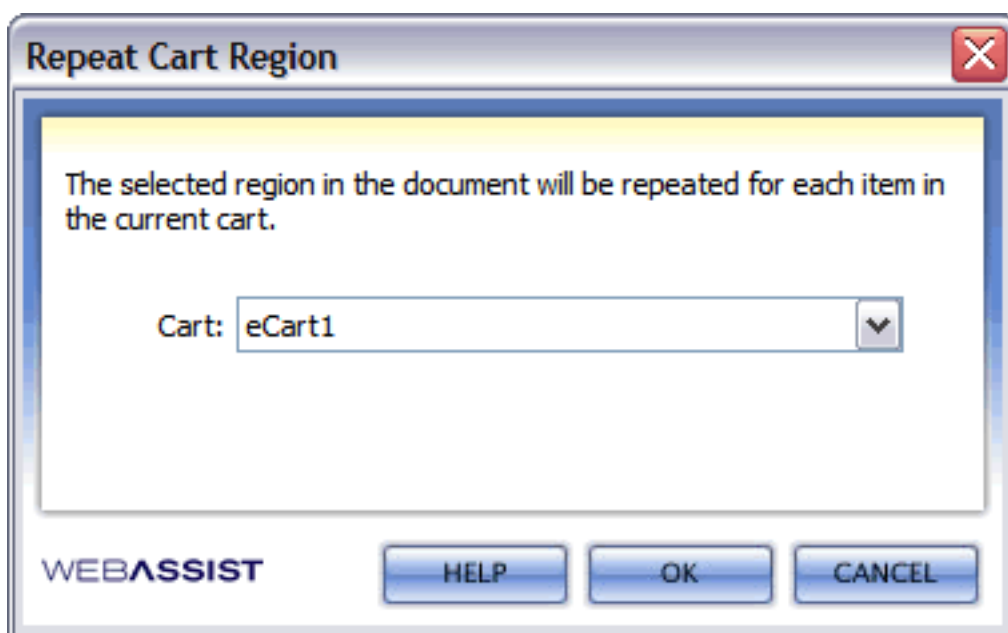
The server behavior parses the data containing the shopping cart contents and displays the data for each rule specific to that category. As you can see, each column contains server-side code that retrieves the rule name and corresponding line item total.

Note: The server-side code for the rule name and rule charge/discount value is available for insert as display text, or to be bound to the default value of a given form element using the Bindings panel (see [Bindings](#) in this section for more information).

To apply the Rules Repeater server behavior:

1. In Code view, select the entire table row (<tr>content</tr>) within your display specific to a given rules category.
2. In the Server Behaviors panel group, click the plus button and select eCart > Cart Display > Rules Repeater.

The Repeat Cart Region dialog box opens.



3. Select the cart name that corresponds to the eCart Object specific to the shopping cart implementation for the given display.
4. Click OK to apply the server behavior to the page.

Customizing your shopping cart display

The eCart Display Manager provides several professional shopping cart displays to help meet your needs, but you may wish to create a shopping cart display that more appropriately fits with the design elements of your site.

If you do need to customize your shopping cart display to fit your needs, all the components of a shopping cart display are available to directly create a shopping cart display from scratch.

The previous section, [Shopping cart display components](#), details all of the individual components of the shopping cart display and how they fit into the entire shopping cart display configuration.

We suggest that if you intend to proceed in this direction, you should use the eCart Display Manager to first create a functional shopping cart display. This will enable you to see the HTML and CSS structure, as well as how the individual components work in the larger display scheme.

Using the display you've created, you can make a determination if it is easier to modify the existing HTML, CSS, and display components specific to your needs, or if it is easier to take a "ground up" approach and start from scratch.

If you decide to start from scratch, you can create your own HTML and CSS structure specific to your needs using Dreamweaver's WYSIWYG. You can then apply the dynamic components of the shopping cart display using the available features within the extension.

Configuring checkout support

eCart provides several features to define the checkout workflow for your site, from processing shopping cart payments, to designing the look and feel of the checkout forms to storing order information and sending email receipts. The two primary features for implementing checkout functionality are:

- **Checkout Wizard**
The easiest, and most direct way to implement shipping rate requests, payment processing, and order management during the checkout process. This feature steps you through a wizard to configure the components of completing customer transactions. Once completed, all the necessary pages are generated and the necessary run-time code is created to complete payment through your merchant provider.
- **Insert Checkout Form**
Directly configures the look and feel of your checkout form, and allows you to directly configure the properties of your form specific to your payment gateway's requirements or your workflow requirements.

Also covered in this section are the two methods supported for transferring transaction information using eCart. These are:

- **Local Checkout**
Local checkout allows you to process transactions locally without redirecting away from your site to a merchant's hosted form to complete the transaction. This allows you to create a contiguous shopping experience where your customers never have to leave your site, and transaction data is returned directly to the same page posting the transaction. Data bindings are available to easily display information returned by your gateway.
- **Remote Checkout**
Remote checkout completes transactions by posting data to a form page hosted by your merchant provider. The Insert Checkout Form feature assists you in configuring a checkout form specific to your merchant provider to post the transaction. The Get Contents from Form server behavior retrieves the transaction data from your provider by applying it to a page that your merchant gateway has been configured to POST the processed transaction data to. You can then use that information within the context of your website to complete the transaction.

The following sections detail the features and methods available to incorporate merchant gateway support into your website using one of the above methods:

- [Using the eCart Checkout Wizard](#)
 - [Insert Checkout Form](#)
 - [Creating a local checkout form](#)
 - [Creating a remote checkout form](#)
 - [Gateway-specific configuration information](#)
-

Using the eCart Checkout Wizard

The eCart Checkout Wizard is the easiest, and most direct way to implement shipping rate requests, payment processing, and order management during the checkout process. This feature steps you through a wizard to configure the components of completing customer transactions. Once completed, all the necessary pages are generated and the necessary run-time code is created to complete payment through your merchant provider.

The wizard consists of an introduction and 4 configuration sections. Each of the steps configures a different aspect of the checkout process, and are covered in detail in the pages that follow.

The following is a breakdown of each of these sections, with links to the help section specific to each. You may also use the navigation arrows above to step through each section as you would step through the wizard. Please review the information provided below these section breakouts prior to advancing:

- [Introduction: Welcome to eCart Checkout Wizard](#)

This section selects your payment gateway, determines if you are configuring shipping for checkout in the wizard, and selects the eCart Object, if multiple are configured, specific to the cart for this checkout experience.

- [Shipping](#)

This section configures shipping rate requests from available providers.

- [Taxes](#)

This section configures tax amounts and VAT charges based on your configuration options.

- [Payments](#)

This section directly configures support for several payment gateway options.

- [Orders](#)

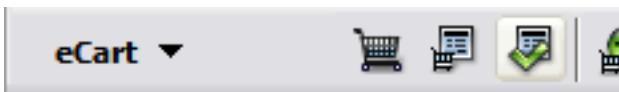
This section directly configures the management of orders, by both storing data in a database, as well as configuring email receipts sent to the end-user.

- [Designs](#)

This section configures the look and feel of the checkout process by selecting a style for the checkout form, as well as configuring all the necessary pages for the checkout workflow.

To access the eCart Checkout Wizard:

1. In Design View in Dreamweaver, open a page that contains either your shopping cart display or a checkout redirect button.
2. Access the eCart Checkout Wizard through one of the following methods:
 - eCart Insert panel



- Insert > WebAssist > eCart > Checkout Wizard

The [next page](#) details the introduction step to begin configuring your checkout process using the eCart Checkout Wizard.

Upon completing the introduction questionnaire, click the 'Get Started' button to begin the wizard.

Use the Back and Next buttons to navigate the steps of the wizard, and click Finish at the end to complete the process and generate the configured checkout pages.

Introduction: Welcome to eCart Checkout Wizard

The introduction is a simple questionnaire that prepares the checkout wizard to offer configuration options specific to your needs. It asks who your merchant gateway provider is, what checkout methodology supported by eCart you wish to use for that provider, whether your shopping cart needs to address shipping charges, and identifies the eCart Object associated to the shopping cart that is going through this checkout process. The questions listed are as follows:

- **Select your payment gateway:** Selects the payment gateway that you have selected to process your payments. Supported gateways include:
 - PayPal
 - WebAssist
 - 2Checkout
 - Authorize.Net
 - eWay
 - Linkpoint
- **Where will payments be processed?:** Based on the gateway selected, this region will populate with all available services provided by that gateway for completing transactions that are supported in eCart. This includes options that process the transaction without leaving your website (local checkout), as well as methods that re-direct to the provider's website to complete the payment transaction (remote checkout). Details specific to each of these options is available in the [Gateway-specific configuration](#) section of this Help documentation.
- **Will you offer shipping services?:** This option determines if shipping configuration should be included in the wizard. This configures the rate request to an available shipping provider supported in eCart, and creates the merchandising rule in the eCart Object that applies that amount to the cart total.
- **Which cart will this checkout be for?:** If more than one eCart Object is configured for your website, this selects the appropriate eCart Object that the checkout process being configured will attempt to process a transaction against.

eCart Checkout Wizard

Welcome to eCart Checkout Wizard

The checkout process gathers information from a variety of sources. This includes information about your site, payment processor, and optional shipping quote provider.

Select your payment gateway:

PayPal

Where will payments be processed?

Locally — Process payments on your site

- Website Payments Pro - US
- Website Payments Pro - UK
- Payflow Pro

Remotely — Process payments at provider's site

- Website Payments Standard
- Payflow Link

Will you offer shipping services?

Yes No

Which cart will this checkout be for:

eCart1

Need help?

[What's a payment gateway?](#)

SIGN UP FOR A GATEWAY

Want to see the entire process? See the Checkout Feature Tour.

LAUNCH FEATURE TOUR

WEBASSIST

HELP **GET STARTED** **CANCEL**

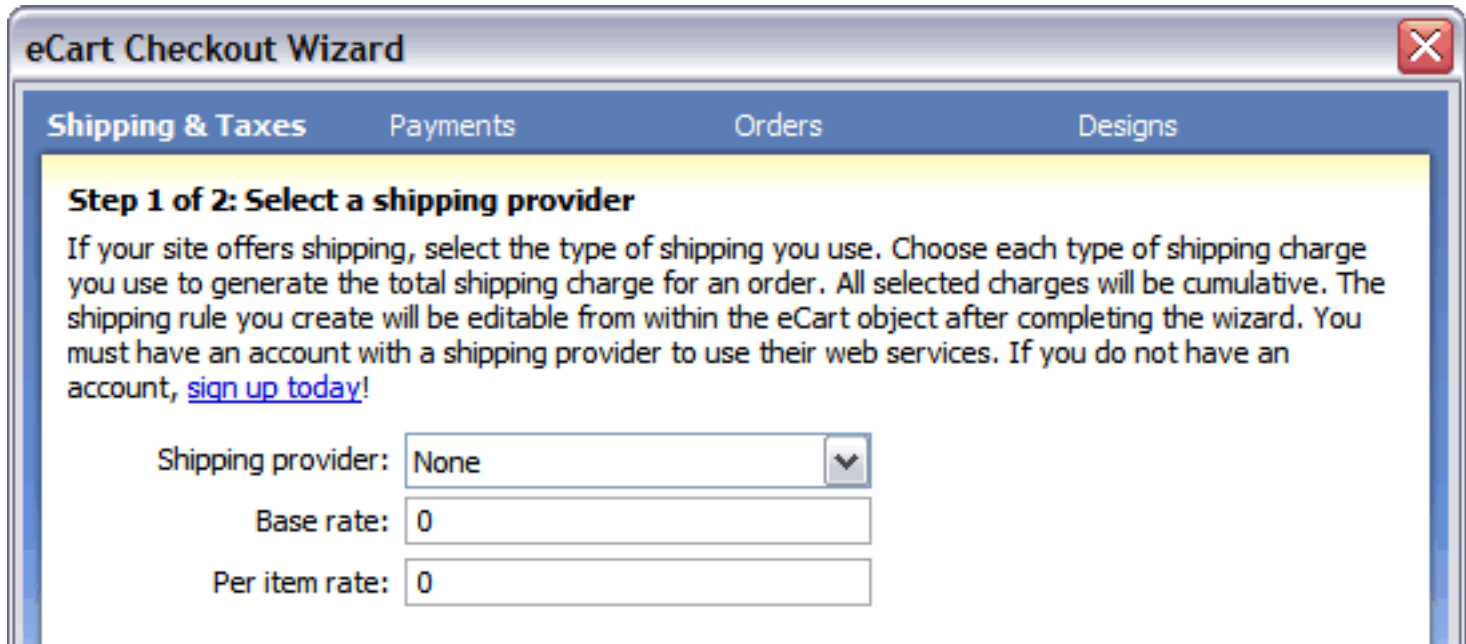
Shipping

If you are dealing in physical goods, the Shipping portion of the Checkout Wizard configures shipping rate requests from available providers.

Merchandising rules are added to the Shipping tab of the eCart Object (see [Discounts, Charges, Taxes and Shipping](#) for more information) that apply the requested shipping rate, a base shipping and handling rate, and an additional per item charge for shipping and handling to the shopping cart subtotal based on the configuration options specified.

These rules can be updated in the eCart Object after completing the Checkout Wizard if you need to customize your configuration or apply more a more advanced calculation scenario. The options available for immediate configuration are as follows:

- **Shipping provider:** Selects a provider to retrieve a shipping rate request from during the checkout process.
- **Base rate:** Applies a flat shipping and handling rate to the shopping cart.
- **Per item rate:** Applies an additional flat rate per item in the shopping cart.



eCart Checkout Wizard

Shipping & Taxes Payments Orders Designs

Step 1 of 2: Select a shipping provider

If your site offers shipping, select the type of shipping you use. Choose each type of shipping charge you use to generate the total shipping charge for an order. All selected charges will be cumulative. The shipping rule you create will be editable from within the eCart object after completing the wizard. You must have an account with a shipping provider to use their web services. If you do not have an account, [sign up today!](#)

Shipping provider:

Base rate:

Per item rate:

Selecting a shipping provider adds subsequent steps to this section of the wizard to configure service for the provider. The following sections detail the configuration options in the Checkout Wizard for each of the available shipping providers:

- [Australia Post](#)
 - [DHL](#)
 - [FedEx](#)
 - [UPS](#)
 - [US Postal Service](#)
-

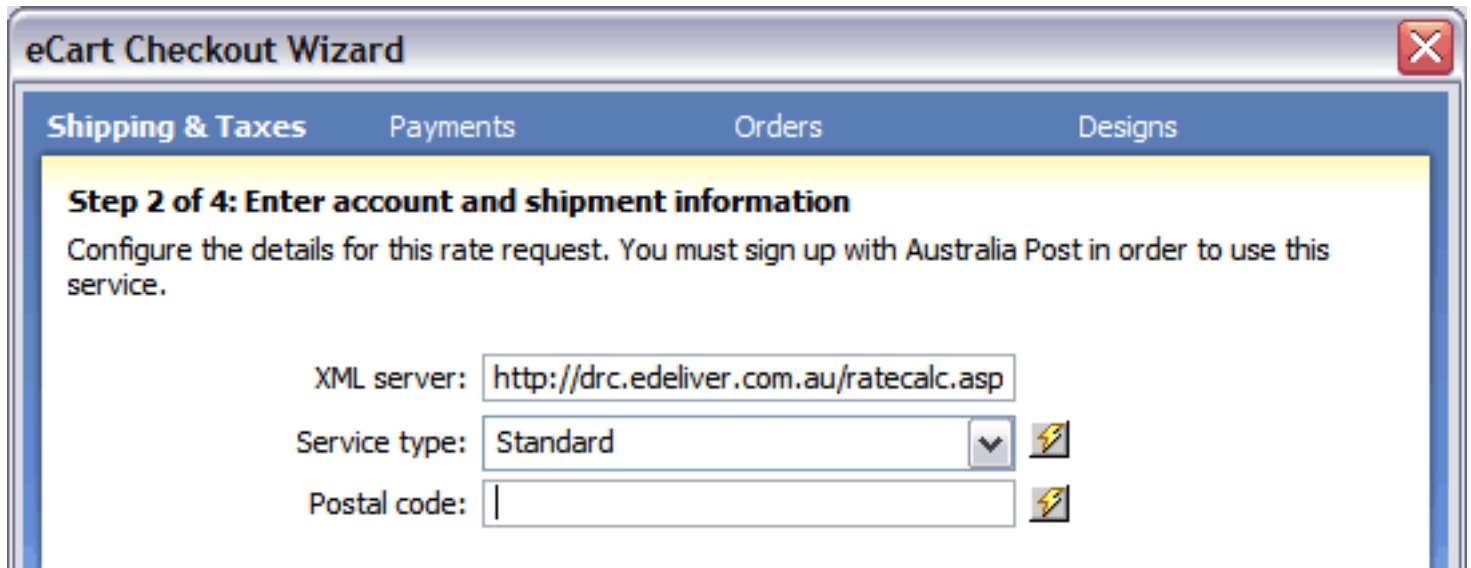
Configuring support for Australia Post

The Australia Post Shipping Rate Request offered in eCart returns shipping quotes in real-time from the Australia Post eDeliver Delivery Rate Calculator.

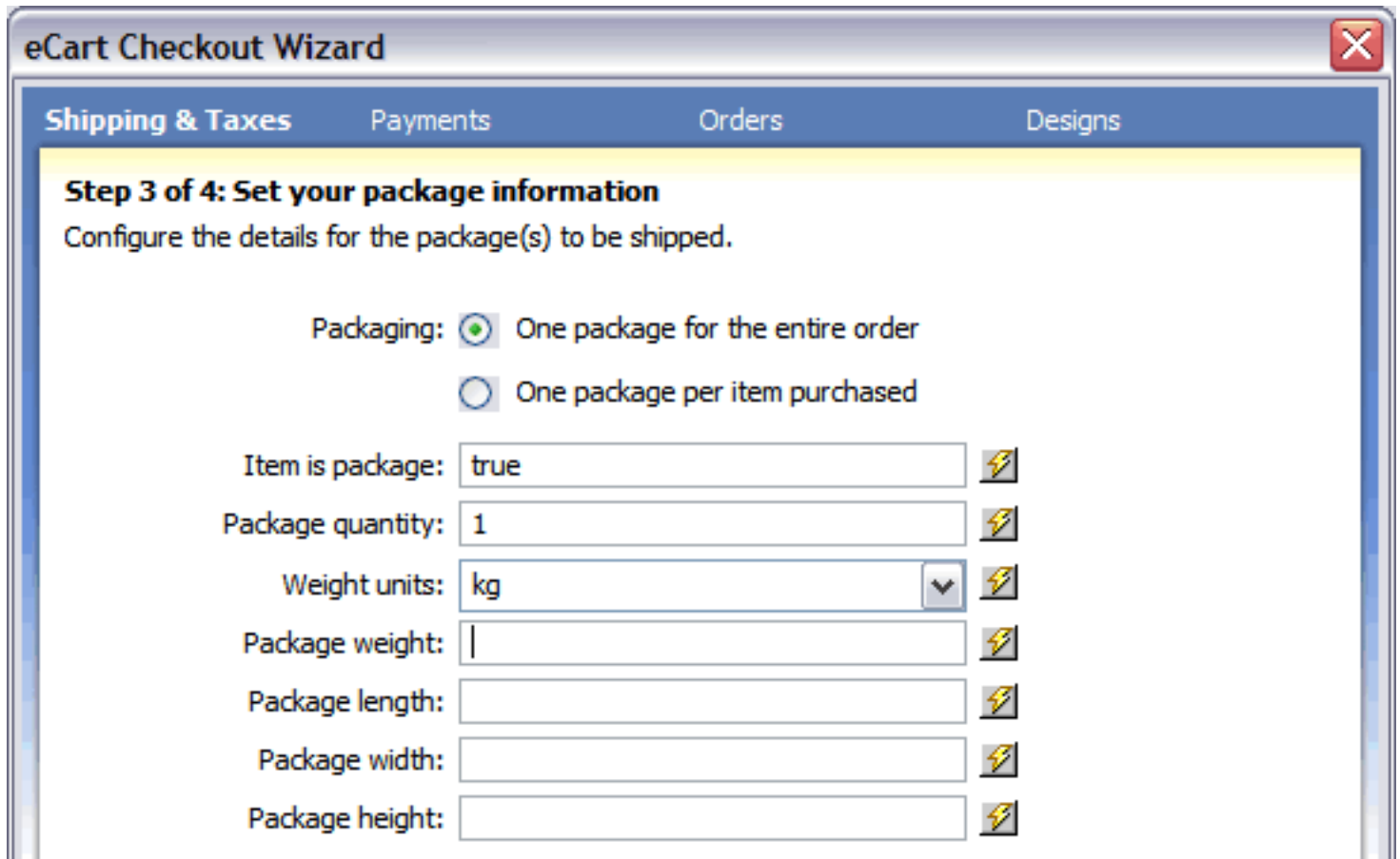
More information regarding signing up for service from this provider can be found at:

http://www.webassist.com/go/australia_post

The following details the configuration requirements to support a rate request from Australia Post:



- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).
- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the service option, insert the Australia Post Service Types snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
- **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.



- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is package:** Determines whether the item in the cart is to be shipped via Australia Post or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Package quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment.
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Advanced configuration

The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

For more detailed information on shipping rate request configuration through Australia Post, please see [Configuring the eCart Australia Post Shipping Rate Request](#).

Configuring support for DHL

The DHL Shipping Rate Request offered in eCart returns shipping quotes in real-time from DHL's XML developer tools for ShipIT and International.

More information regarding signing up for this service from this provider can be found at:

<http://www.webassist.com/go/dhl>

The following details the configuration requirements to support a rate request from DHL:

eCart Checkout Wizard

Shipping & Taxes Payments Orders Designs

Step 2 of 6: Enter DHL account information
You must sign up with DHL in order to use this service.

User ID:

Password:

Domestic shipping key:

International shipping key:

Account number:

XML server:

- **User ID:** Enter the User ID associated with your DHL shipping account.
- **Password:** Enter the password associated to your account's User ID.
- **Domestic shipping key:** Enter the domestic shipping key for your XML ShipIT account. If you do not have this information, contact a DHL account representative at XMLRequests@dhl.com
- **International shipping key:** Enter the domestic shipping key for your XML International account. If you do not have this information, contact a DHL account representative at XMLRequests@dhl.com
- **Account number:** Enter your DHL account number.
- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).

eCart Checkout Wizard

Shipping & Taxes Payments Orders Designs

Step 3 of 6: Enter your shipment information
Configure the details for this rate request. Choose your service type first, then enter some information on how this shipment will be billed. Some options will effect the shipping cost.

Carrier type: ⚡

Billing party: ⚡

Billing account: ⚡

COD payment: ⚡

Trace: ⚡

Shipping date:

- **Carrier type:** Selects the specific kind of service desired. To allow the customer to choose the service option, insert the DHL Service Types snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
- **Billing party:** Selects the appropriate party to be billed. This can be set statically or retrieved from a dynamic data binding.
- **Billing account:** Selects the DHL account to be billed. This can be set statically or retrieved from a dynamic data binding.
- **COD payment:** Sets the options for available Cash on Delivery payments. This can be set statically or retrieved from a dynamic data binding.
- **Trace:** Sets tracking information specific to your rate request. This is not processed by DHL when the rate request is submitted, but is returned in the rate request response for your tracking purposes. This can be retrieved from a dynamic data binding.
- **Shipping date:** Sets the date the package is to be picked up by DHL. This can be set statically or retrieved from a dynamic data binding.

eCart Checkout Wizard

Shipping & Taxes Payments Orders Designs

Step 4 of 6: Select your international options
Select the country of origin for this shipment. Packages shipped from or to locations outside of the United States will use International Shipping.

Duty paying party: ⚡

Customs value: ⚡

Duty account number: ⚡

Country of origin: ⚡

Override: ⚡

- **Duty paying party:** Sets the party responsible for paying any duty associated with the international order, affecting the rate total. This can be set statically or retrieved from a dynamic data binding.
- **Customs value:** Sets the value of the shipment to determine the customs charges. This can be set statically or retrieved from a dynamic data binding, typically the subtotal of the items included in the shipment.
- **Duty account number:** Sets the account number for duty collection. This can be set statically or retrieved from a dynamic data binding.
- **Country of origin:** Sets the country the shipment originates from, determining if the order is international or not. This can be set statically or retrieved from a dynamic data binding.
- **Override:** Selects the appropriate override scenario if the postal code entered is not available in the system, or if a delivery option is selected that doesn't meet the rate request requirements. This allows a rate to still be generated, which may be incorrect, but does cause the request to fail. This can be set statically or retrieved from a dynamic data binding.

eCart Checkout Wizard

Shipping & Taxes Payments Orders Designs

Step 5 of 6: Set your package information
Configure the details for the package(s) to be shipped. All size measurements are in inches.

Packaging: One package for the entire order
 One package per item purchased

Item is package: ⚡

Quantity: ⚡

Weight units: ⚡

Package weight: ⚡

Package type: ⚡

Package length: ⚡

Package width: ⚡

Package height: ⚡

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is package:** Determines whether the item in the cart is to be shipped via Australia Post or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment.
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Determines the type of packaging used to make the shipment.
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Advanced configuration

The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

For more detailed information on shipping rate request configuration through DHL, please see [Configuring the eCart DHL Shipping Rate Request](#).

Configuring support for FedEx

The FedEx Shipping Rate Request offered in eCart returns shipping quotes in real-time from the FedEx Web service.

More information regarding signing up for this service from this provider can be found at:

<http://www.webassist.com/go/fedex>

The following details the configuration requirements to support a rate request from FedEx:

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 2 of 6: Enter FedEx account information'. Below the title, it says 'You must sign up with FedEx in order to use this service.' There are three input fields: 'Account number:' (empty), 'Meter number:' (empty), and 'XML server:' (containing 'https://gatewaybeta.fedex.com:443/G/').

- **Account number:** Enter the FedEx account number assigned to your account.
- **Meter number:** Enter the meter number you want to use for your account.
- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 3 of 6: Enter your shipment information'. Below the title, it says 'Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment.' There are four dropdown menus: 'Carrier type:' (FedEx Ground), 'Service type:' (FedEx Ground), 'Dropoff type:' (Regular pick up), and 'Return type:' (None). Each dropdown has a lightning bolt icon to its right.

- **Carrier type:** Chooses the primary carrier method, either FedEx Ground or FedEx Express. To allow the customer to choose the option, insert the FedEx Carrier Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the option, insert the FedEx Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
- **Dropoff type:** Determines the type of drop-off desired. Options are Regular pick up, Request courier, Drop box, Business service center, and Station; typically these are set to one type by the shipper.
- **Return type:** Sets the type of return allowed. Options are Non-return, Non-return or online label, Return manager, Print return label, Email label, and Express/Call Tag. Again, these are typically set to one value by the shipper.

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 4 of 6: Select your shipping origin'. Below the title, it says 'Enter the location from which the order will be shipped. Postal code is the only required field for shipments within the United States.' There are three dropdown menus: 'State:' (Not Applicable), 'Postal code:' (empty), and 'Country:' (United States). Each dropdown has a lightning bolt icon to its right.

- **State:** Enter or choose the state or province where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
- **Postal code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
- **Country:** Enter or choose the country where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 5 of 6: Set your package information'. Below the title, it says 'Some of these settings may not apply depending on your chosen carrier type.' There are two radio buttons: 'Packaging:' (One package for the entire order, selected) and 'One package per item purchased'. Below are several input fields: 'Item is package:' (true), 'Quantity:' (1), 'Declared value (USD):' (empty), 'Weight units:' (LBS), 'Package weight:' (empty), 'Package type:' (Your Packaging), 'Size units:' (IN), 'Package length:' (empty), 'Package width:' (empty), and 'Package height:' (empty). Each input field has a lightning bolt icon to its right.

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is shipped:** Determines whether the item in the cart is to be shipped via FedEx or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
- **Declared value (USD):** Sets the declared value of each object in the cart. If the items have separate values and they are shipped separately, use the Dynamic Data dialog box to assign this field to the proper column; to combine values, use the Dynamic Data dialog box to select a calculated shopping cart column like TotalPrice.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used. Choose the Your Packaging option if you use non-FedEx packaging; to use FedEx packaging choose one of the following: FedEx Envelope, FedEx Pak, FedEx Express Box, FedEx Tube, FedEx 10Kg Box, or FedEx 25Kg Box. If you previously selected FedEx Ground as your service, only the Your Packaging option is available.
- **Size units:** Declares the type of size units to be used in calculating the total size of the shipment. Standard values are either IN (inches) or CM (centimeters).
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Advanced configuration

The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

For more detailed information on shipping rate request configuration through FedEx, please see [Configuring the eCart FedEx Shipping Rate Request](#).

Configuring support for UPS

The UPS Shipping Rate Request offered in eCart returns shipping quotes in real-time from the UPS Web service.

More information regarding signing up for this service from this provider can be found at:

<http://www.webassist.com/go/ups>

The following details the configuration requirements to support a rate request from UPS:

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 2 of 6: Enter UPS account information'. Below the title, it says 'You must sign up with UPS in order to use this service.' There are three input fields: 'User ID:', 'Password:', and 'XML access key:'.

- **User ID:** Enter the UPS user ID assigned to your account.
- **Password:** Enter the password for your account.
- **XML access key:** Enter the XML access key for your account.

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 3 of 6: Enter your shipment information'. Below the title, it says 'Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment.' There are four fields: 'Service type:' (dropdown menu with 'Ground' selected), 'Pickup type:' (dropdown menu with 'Daily Pickup' selected), 'Customer class:' (dropdown menu with 'None' selected), and 'UPS customer number:' (text input field).

- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the option, insert the UPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
- **Pickup type:** Determines the type of pickup desired. Options are Daily Pickup, Customer Counter, One Time Pickup, On Call Air Pickup, Suggested Retail (UPS Store), Letter Center and Air Service Center; typically these are set to one type by the shipper.
- **Customer class:** Sets the customer class. Options are None, Wholesale, Occasional Pickup and Retail. Again, these are typically set to one value by the shipper.
- **UPS customer number:** Sets the customer number of the shipper; typically this is a single value.

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 4 of 6: Select your shipping origin'. Below the title, it says 'Enter the location from which the order will be shipped. Postal code is the only required field for shipments within the United States. Countries marked with an asterisk require a postal code.' There are four fields: 'City:' (text input field), 'State:' (dropdown menu with 'Not Applicable' selected), 'Postal code:' (text input field), and 'Country:' (dropdown menu with 'United States*' selected).

- **City:** Enter or choose the city where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
- **State:** Enter or choose the state or province where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
- **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
- **Country:** Enter or choose the country where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.

The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The title is 'Step 5 of 6: Set your package information'. Below the title, it says 'Configure the details for the package(s) to be shipped.' There are several fields: 'Packaging:' (radio buttons for 'One package for the entire order' and 'One package per item purchased'), 'Item is package:' (text input field with 'true'), 'Quantity:' (text input field with '1'), 'Weight units:' (dropdown menu with 'LBS'), 'Package weight:' (text input field), 'Package type:' (dropdown menu with 'Your Packaging'), 'Size units:' (dropdown menu with 'IN'), 'Package length:' (text input field), 'Package width:' (text input field), and 'Package height:' (text input field).

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is package:** Determines whether the item in the cart is to be shipped via UPS or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used. Choose the Your Packaging option if you use non-UPS packaging; to use UPS packaging choose one of the following: UPS Express Envelope, UPS Express Box, UPS Pak, UPS Tube, UPS 10Kg Box, or UPS 25Kg Box. If you previously selected UPS Ground as your service, only the Your Packaging, UPS 10Kg Box and UPS 25Kg Box options are available.
- **Size units:** Declares the type of size units to be used in calculating the total size of the shipment. Standard values are either IN (inches) or CM (centimeters).
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Advanced configuration

The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

For more detailed information on shipping rate request configuration through UPS, please see [Configuring the eCart UPS Shipping Rate Request](#).

Configuring support for USPS

The USPS Shipping Rate Request offered in eCart returns shipping quotes in real-time from the USPS Web service.

More information regarding signing up for this service from this provider can be found at:

<http://www.webassist.com/go/usps>

The following details the configuration requirements to support a rate request from USPS:

The screenshot shows the 'eCart Checkout Wizard' window with tabs for 'Shipping & Taxes', 'Payments', 'Orders', and 'Designs'. The current step is 'Step 2 of 6: Enter USPS account information'. Below the title, it says 'You must sign up with USPS in order to use this service.' There are two input fields: 'User ID:' with an empty text box, and 'XML server:' with a text box containing 'http://testing.shippingapis.com/Shipping'.

- **User ID:** Enter the USPS user ID assigned to your account.
- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).

The screenshot shows the 'eCart Checkout Wizard' window with tabs for 'Shipping & Taxes', 'Payments', 'Orders', and 'Designs'. The current step is 'Step 3 of 6: Enter your shipment information'. Below the title, it says 'Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment.' There are two dropdown menus: 'Service type:' with 'First-Class' selected and a lightning bolt icon, and 'Default international:' with 'None' selected and a lightning bolt icon.

- **Service type:** Selects the specific kind of domestic service desired. To allow the customer to choose the option, insert the USPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
- **Default international:** Selects the specific kind of international service desired. To allow the customer to choose the option, insert the USPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.

The screenshot shows the 'eCart Checkout Wizard' window with tabs for 'Shipping & Taxes', 'Payments', 'Orders', and 'Designs'. The current step is 'Step 4 of 6: Select your shipping origin'. Below the title, it says 'Enter the location from which the order will be shipped. Postal code is the only required field.' There is one input field: 'Postal code:' with an empty text box and a lightning bolt icon.

- **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.

The screenshot shows the 'eCart Checkout Wizard' window with tabs for 'Shipping & Taxes', 'Payments', 'Orders', and 'Designs'. The current step is 'Step 5 of 6: Set your package information'. Below the title, it says 'Configure the details for the package(s) to be shipped.' There are several options and input fields: 'Packaging:' with radio buttons for 'One package for the entire order' (selected) and 'One package per item purchased'; 'Item is package:' with a text box containing 'true' and a lightning bolt icon; 'Quantity:' with a text box containing '1' and a lightning bolt icon; 'Weight units:' with a dropdown menu containing 'lb' and a lightning bolt icon; 'Package weight:' with an empty text box and a lightning bolt icon; 'Package type:' with a dropdown menu containing 'Your Packaging' and a lightning bolt icon; 'Package size:' with a dropdown menu containing 'Regular' and a lightning bolt icon; 'Package machinable:' with a dropdown menu containing 'True' and a lightning bolt icon; and 'Intl. package type:' with a dropdown menu containing 'Package' and a lightning bolt icon.

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is shipped:** Determines whether the item in the cart is to be shipped via UPS or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used.
- **Package size:** Sets the size of the package for the rate request. Options are:
 - Regular: package length plus girth is 84 inches or less
 - Large: package length plus girth that measures more than 84 inches (but less than 108 inches) and weighs less than 15 pounds.
 - Oversized: package length plus girth is more than 108 but not more than 130 inches
- **Package machinable:** Applies only to Parcel Post. The size, content, and weight of a Parcel Post package determine whether it is machinable or nonmachinable. For Parcel Post, the value entered must be either "True" or "False." The "True" or "False" value specifically applies to the size and content of the package. A package is deemed nonmachinable if it weighs less than 6 ounces or over 35 pounds.
- **International package type:** If the shipment is international, sets the valid packaging type.

Advanced configuration

The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

For more detailed information on shipping rate request configuration through USPS, please see [Configuring the eCart USPS Shipping Rate Request](#).

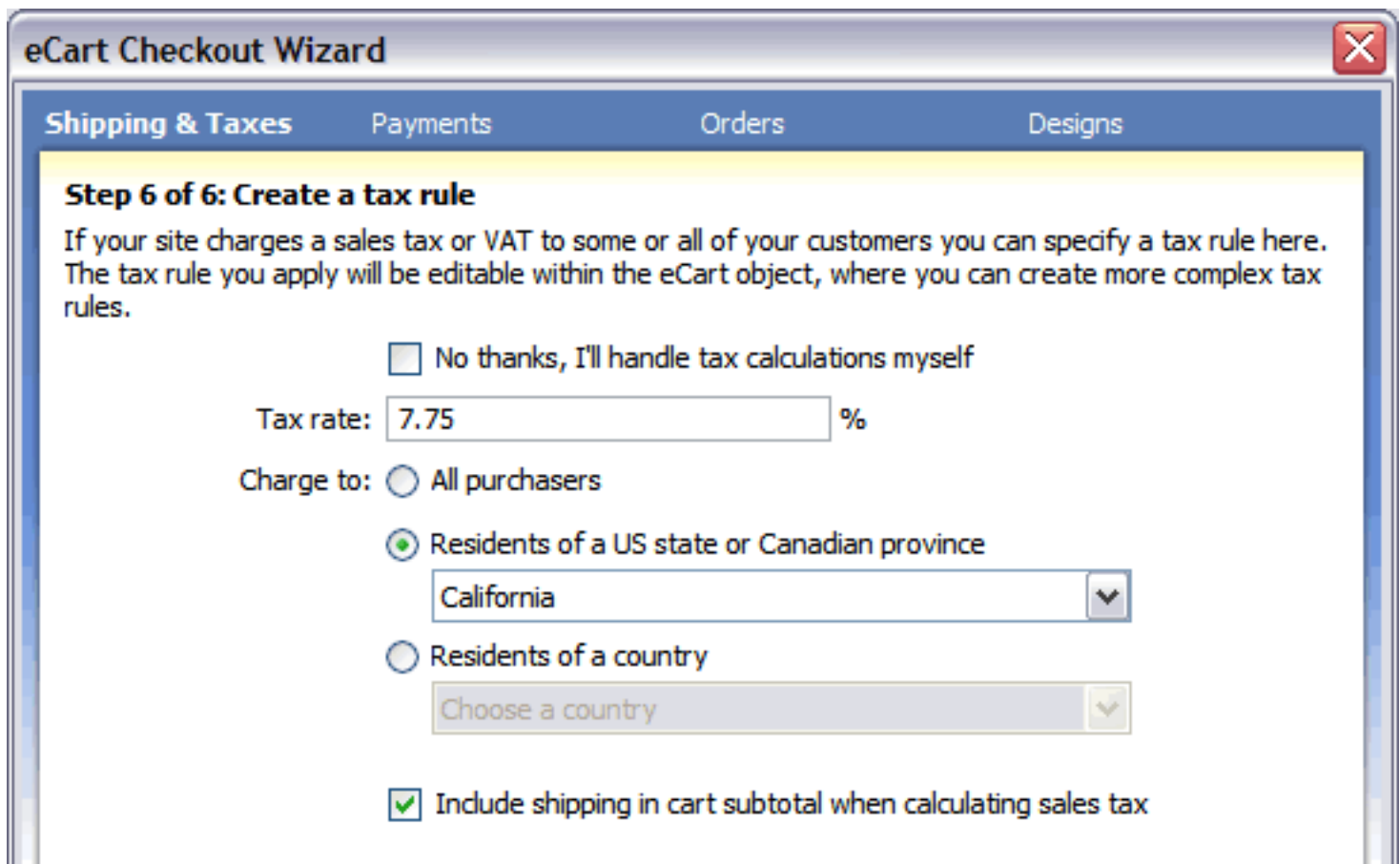
Taxes

If you need to apply a tax or VAT charge to the cart, the Taxes portion of the Checkout Wizard configures tax amounts based on your configuration options.

A merchandising rule is added to the Taxes tab of the eCart Object (see [Discounts, Charges, Taxes and Shipping](#) for more information) that applies a tax charge based on the configuration options specified.

These rules can be updated in the eCart Object after completing the Checkout Wizard if you need to customize your configuration, apply multiple tax rules, or create a more advanced tax calculation scenario. The options available for immediate configuration are as follows:

- **No thanks, I'll handle tax calculations myself:** Skips configuration of tax rules in the Checkout Wizard. Rules can always be added after the fact directly to the eCart Object.
- **Tax rate:** A numeric value that sets the percentage of tax calculated based on the cart subtotal.
- **Charge to:** Sets criteria for applying the tax rate based on geography information gathered from the checkout form.
- **Include shipping in cart subtotal when calculating sales tax:** Determines if the shipping charges applied to the cart should be considered taxable for this rule.



The screenshot shows the 'eCart Checkout Wizard' window with the 'Shipping & Taxes' tab selected. The window title is 'eCart Checkout Wizard' and it has a close button in the top right corner. The main content area is titled 'Step 6 of 6: Create a tax rule' and contains the following text: 'If your site charges a sales tax or VAT to some or all of your customers you can specify a tax rule here. The tax rule you apply will be editable within the eCart object, where you can create more complex tax rules.'

The form includes the following options:

- No thanks, I'll handle tax calculations myself
- Tax rate: %
- Charge to:
 - All purchasers
 - Residents of a US state or Canadian province
 -
 - Residents of a country
 -
- Include shipping in cart subtotal when calculating sales tax

Payments

The Checkout Wizard directly configures support for several payment gateway options. The gateway configured in this step is based on your provider and service selection in the [Introduction: Welcome to eCart Checkout Wizard](#) section.

For each of the available providers, a selection of services are available that are supported directly in the Checkout Wizard. Services fall into two categories: remote and local. Local checkout support means that the service allows you to complete the payment transaction without leaving your website. Remote service means that to complete your checkout transaction, you must redirect to a different website.

If you are using PayPal Website Payments Pro, PayPal Express Checkout service is automatically added as an optional process in your checkout, per PayPal's service requirements. In addition, in step 2 of the Payments section, all other local checkout services supported in the Checkout Wizard have the option of including PayPal Express Checkout as an alternative payment method supported in your website. Support for Express Checkout is covered in the PayPal section below.

More information specific to configuration of all supported services through the Checkout Wizard is available in the following provider sections:

- [PayPal](#)
 - [WebAssist](#)
 - [2Checkout](#)
 - [Authorize.Net](#)
 - [eWay](#)
 - [Linkpoint](#)
-

PayPal

The following details the multiple service offerings available from PayPal for completing payment transactions through their services.

More information regarding available services and support from this provider can also be found on our website at:

U.S.
http://www.webassist.com/go/paypal_gateway

U.K.
http://www.webassist.com/go/paypal_UK_gateway

Each of these sections links to the configuration details within the Checkout Wizard specific to each service:

- [Website Payments Pro \(US\)](#)
- [Website Payments Pro \(UK\)](#)
- [Payflow Pro](#)
- [Website Payments Standard](#)
- [Payflow Link](#)
- [Express Checkout](#)

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the server behavior applied to the checkout page specific to your gateway.

For more detailed information specific to server behavior configuration and PayPal services, please refer to the PayPal support section.

Website Payments Pro (US)

- **Username:** Email address associated to your PayPal account to login to PayPal's site and access your account services.
- **Password:** Password associated to your username.
- **Signature:** The encrypted string value to identify yourself.

Note: Follow these steps to acquire your username, password, and signature for the PayPal API. If you are testing with a Sandbox account, repeat these steps for both your Sandbox and Live accounts. We recommend opening a separate web browser to perform these steps.

1. Log in to your PayPal Premier or Business account.
2. Click the **Profile** subtab located in the top navigation area.
3. Click the **API Access** link under the Account Information header.
4. Click the **Request API Credentials**.
5. Click the **API signature** radio button under the Credential Type heading.
6. Complete the API Credential Request form by clicking the agreement checkbox and clicking **Submit**.
7. Save the values for API Username and Password.
8. Select the value listed next to Signature Hash, copy it, and save it to a file or some other location. This is your API Signature.

Website Payments Pro (UK)

- **USER:** The profile ID for a user in your account authorized to process transactions.
- **VENDOR:** Your merchant ID specified during the creation of your account.
- **PWD:** The password specified at the time you registered for the account.

Payflow Pro

- **USER:** The profile ID for a user in your account authorized to process transactions.
- **VENDOR:** Your merchant ID specified during the creation of your account.
- **PWD:** The password specified at the time you registered for the account.

Website Payments Standard

- **business:** the user id associated to your PayPal account to login to PayPal's site and access your account services.

Payflow Link

- **LOGIN:** the user id associated to your PayPal account to login to PayPal's site and access your account services.

Express Checkout

- **Do not include PayPal Express Checkout at this time:** For local checkout services other than PayPal Payments Pro (US), you have the option of including support for PayPal Express Checkout. To disable adding this feature, this box must be checked.
- **Username:** Email address associated to your PayPal account to login to PayPal's site and access your account services.
- **Password:** Password associated to your username.
- **Signature:** The encrypted string value to identify yourself.

Note: Follow these steps to acquire your username, password, and signature for the PayPal API. If you are testing with a Sandbox account, repeat these steps for both your Sandbox and Live accounts. We recommend opening a separate web browser to perform these steps. The sandbox server testing environment will not work for ASP configurations if you do not have your SSL certificate installed on your website. Please refer to PayPal's developer center for more information.

1. Log in to your PayPal Premier or Business account.
2. Click the **Profile** subtab located in the top navigation area.
3. Click the **API Access** link under the Account Information header.
4. Click the **Request API Credentials**.
5. Click the **API signature** radio button under the Credential Type heading.
6. Complete the API Credential Request form by clicking the agreement checkbox and clicking **Submit**.
7. Save the values for API Username and Password.
8. Select the value listed next to Signature Hash, copy it, and save it to a file or some other location. This is your API Signature.

WebAssist

The following details the WebAssist Payment Gateway service for completing payment transactions.

Support for adding PayPal's Express Checkout service as an alternate payment method to your customers is available in the [PayPal](#) page of the Checkout Wizard.

More information regarding available services and support from this provider can also be found on our website at:

http://www.webassist.com/go/webassist_gateway

The screenshot shows a window titled "eCart Checkout Wizard" with a red close button in the top right corner. The window has a blue header bar with four tabs: "Shipping & Taxes" (with a checkmark), "Payments" (selected), "Orders", and "Designs". Below the tabs, the main content area has a yellow background and contains the following text:

Step 1 of 2: Specify payment provider account details

The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page.

publisher_name: Merchant ID issued by us to you at time of sign-up.

publisher_email: Address where order confirmation is sent. This is also the return address for email sent to customer.

- **publisher_name:** the merchant account ID associated to your account
- **publisher_email:** the email address where notification is sent for completed orders.

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the server behavior applied to the checkout page specific to your gateway.

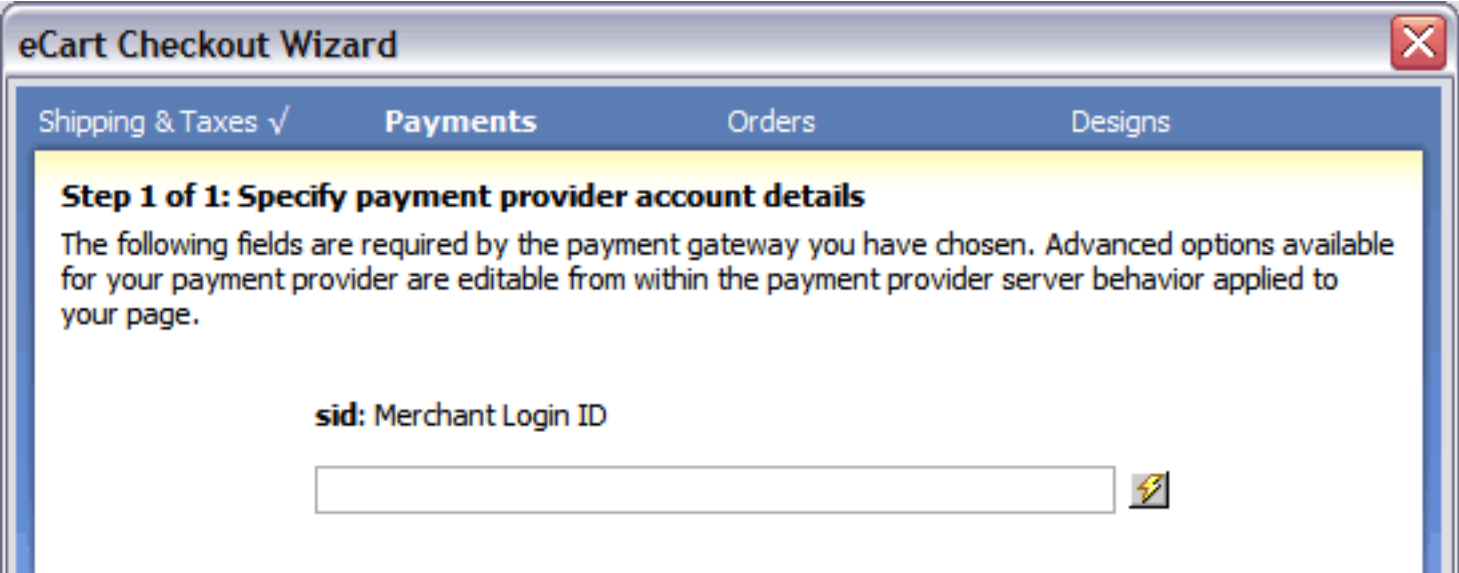
For more detailed information specific to server behavior configuration and WebAssist services, please refer to the WebAssist support section.

2Checkout

The following details the 2Checkout service for completing payment transactions.

More information regarding available services and support from this provider can also be found on our website at:

http://www.webassist.com/go/2checkout_gateway



- **sid:** the merchant account ID associated to your account

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the checkout form specific to your gateway.

For more detailed information specific to form configuration and 2Checkout services, please refer to the 2Checkout support section.

Authorize.Net

The following details the multiple service offerings available from Authorize.Net for completing payment transactions through their services.

Support for adding PayPal's Express Checkout service as an alternate payment method to your customers is available in the [PayPal](#) page of the Checkout Wizard.

More information regarding available services and support from this provider can also be found on our website at:

http://www.webassist.com/go/authorize_net_gateway

Each of these sections links to the configuration details within the Checkout Wizard specific to each service:

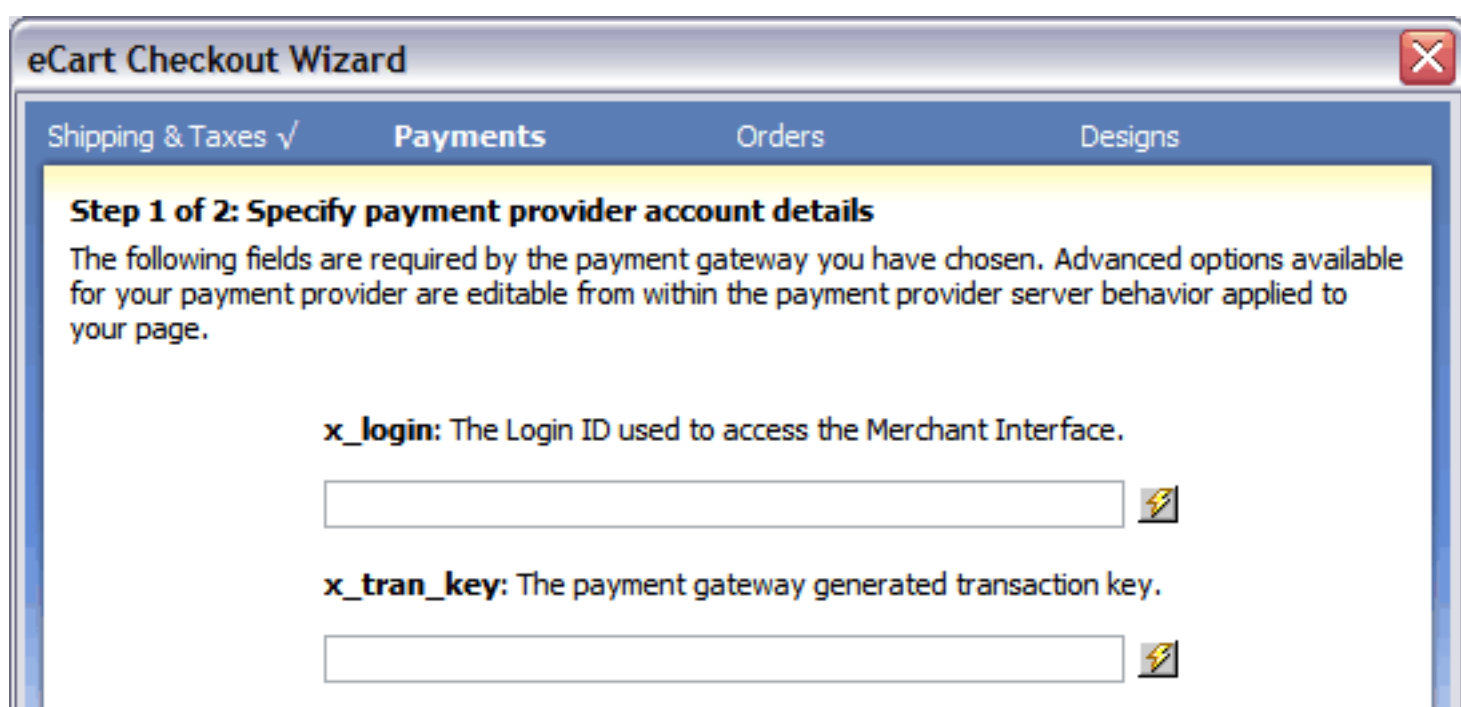
- [AIM and SecureSource Credit Card](#)
- [SIM](#)

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the server behavior applied to the checkout page specific to your gateway.

For more detailed information specific to server behavior configuration and Authorize.NET services, please refer to the Authorize.Net support section.

AIM and SecureSource




eCart Checkout Wizard

Shipping & Taxes ✓ Payments Orders Designs


Step 1 of 2: Specify payment provider account details

The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page.

x_login: The Login ID used to access the Merchant Interface.

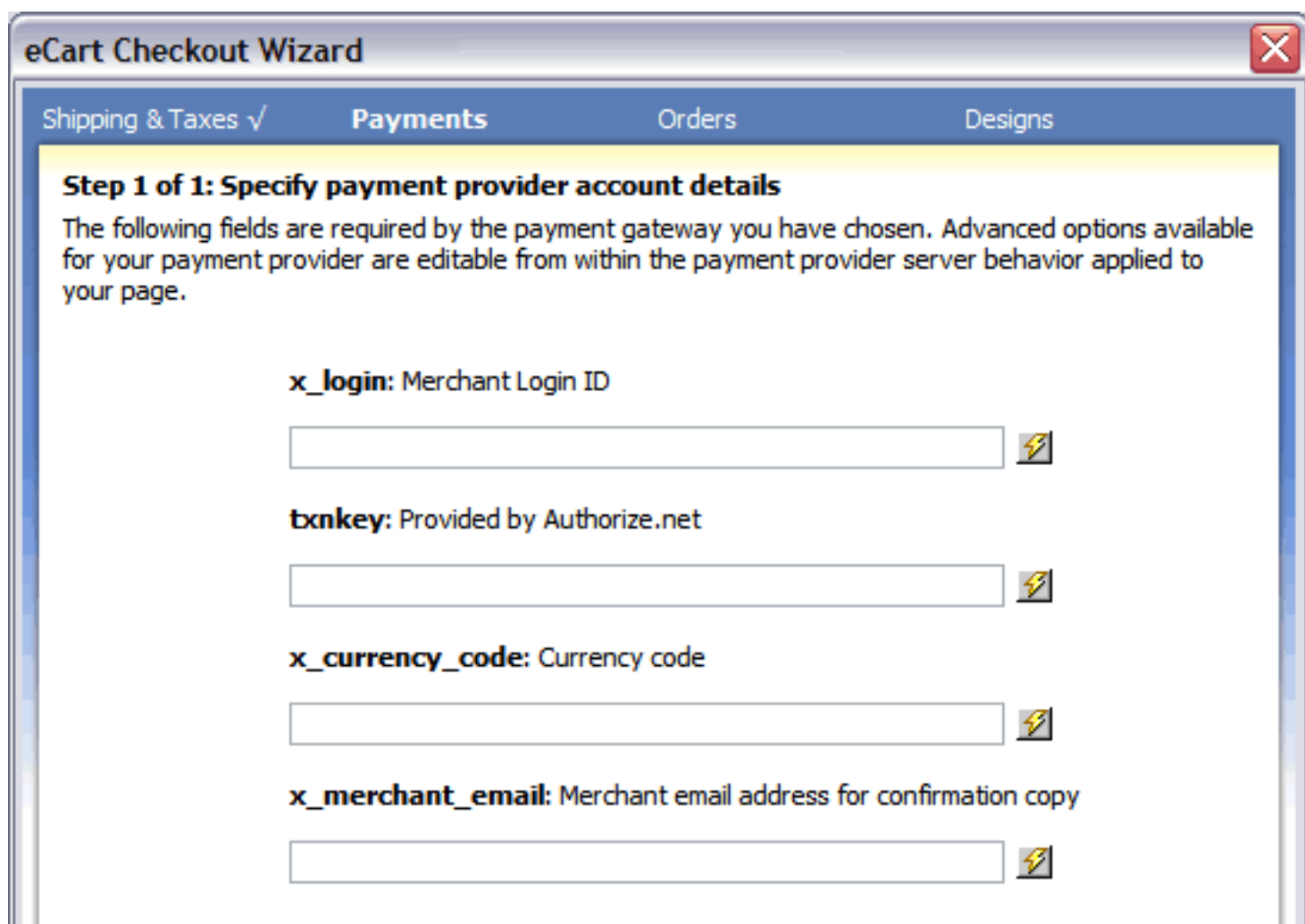


x_tran_key: The payment gateway generated transaction key.



- **x_login:** The primary login ID for your Authorize.Net account.
- **x_tran_key:** A string generated by Authorize.Net to securely identify you as eligible to process transactions through the account. To get the key:
 1. Log in to your Authorize.Net account.
 2. Click the **Account** subtab located in the top navigation area.
 3. Under Security Settings, click the **API Login ID and Transaction Key** link.
 4. Follow the provided instructions to reset your transaction key. **IMPORTANT:** If you already have a gateway up and running using this provider, this process will disable the old key and your prior checkout configuration. Do not reset this key if you have an active existing gateway. Speak to the resource responsible for the current system's configuration to get this information.

SIM




eCart Checkout Wizard

Shipping & Taxes ✓ Payments Orders Designs


Step 1 of 1: Specify payment provider account details

The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page.


x_login: Merchant Login ID




txnkey: Provided by Authorize.net



x_currency_code: Currency code



x_merchant_email: Merchant email address for confirmation copy



- **x_login:** The primary login ID for your Authorize.Net account.
- **txnkey:** A string generated by Authorize.Net to securely identify you as eligible to process transactions through the account. To get the key:
 1. Log in to your Authorize.Net account.
 2. Click the **Account** subtab located in the top navigation area.
 3. Under Security Settings, click the **API Login ID and Transaction Key** link.
 4. Follow the provided instructions to reset your transaction key. **IMPORTANT:** If you already have a gateway up and running using this provider, this process will disable the old key and your prior checkout configuration. Do not reset this key if you have an active existing gateway. Speak to the resource responsible for the current system's configuration to get this information.
- **x_currency_code:** The currency the transaction is to be processed in. US Dollars is USD and British Pounds sterling is GBP. Please refer to Authorize.Net's SIM Integration guide for a comprehensive list of all available currency codes.
- **x_merchant_email:** The email address transaction confirmation emails are sent to.

eWay

The following details the necessary configuration details in the Checkout Wizard to use eWay for completing payment transactions, using either the XML or Shared methods.

Support for adding PayPal's Express Checkout service as an alternate payment method to your customers is available in the [PayPal](#) page of the Checkout Wizard.

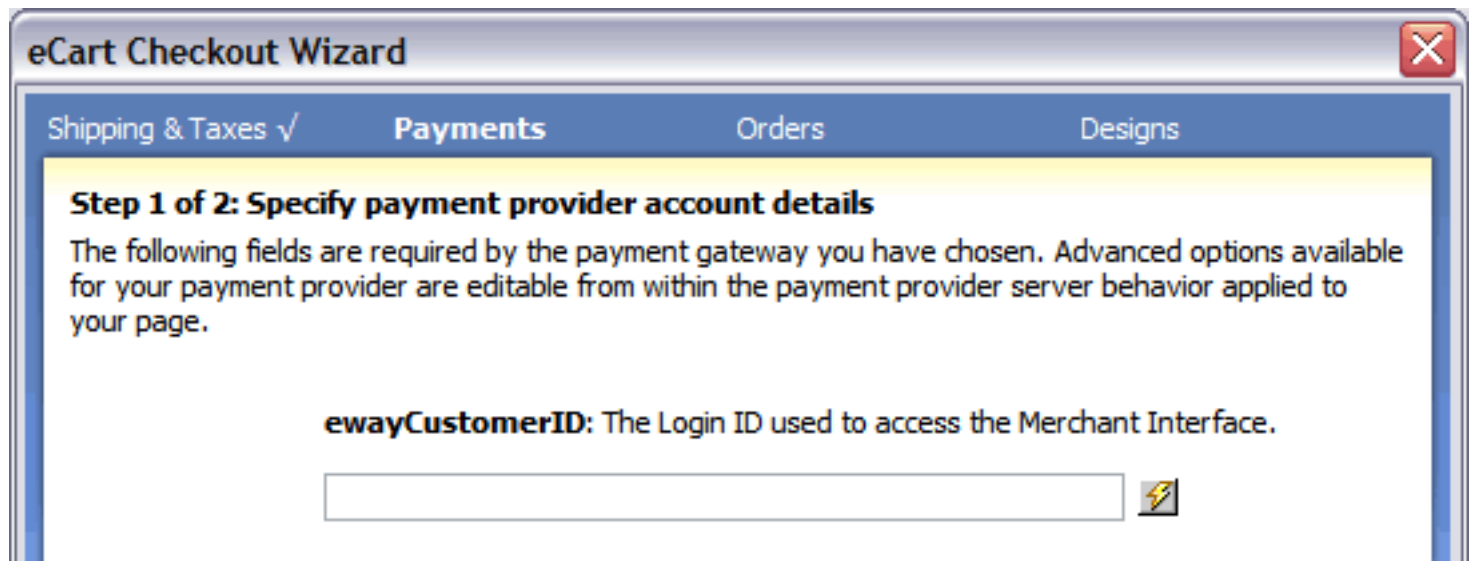
More information regarding available services and support from this provider can also be found on our website at:

http://www.webassist.com/go/eway_gateway

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the server behavior applied to the checkout page specific to your gateway.

For more detailed information specific to server behavior configuration and eWay services, please refer to the eWay support section.



The screenshot shows a window titled "eCart Checkout Wizard" with a close button in the top right corner. The window has a blue header bar with four tabs: "Shipping & Taxes ✓", "Payments", "Orders", and "Designs". The "Payments" tab is selected. Below the tabs, there is a yellow highlighted area with the text: "Step 1 of 2: Specify payment provider account details". Below this, there is a paragraph: "The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page." Below the paragraph, there is a label: "ewayCustomerID: The Login ID used to access the Merchant Interface." Below the label, there is a text input field and a lightning bolt icon.

- **ewayCustomerID:** The primary login ID to access your eWay account.
-

Linkpoint

The following details the multiple service offerings available from Linkpoint for completing payment transactions through their services.

Support for adding PayPal's Express Checkout service as an alternate payment method to Linkpoint API for your customers is available in the [PayPal](#) page of the Checkout Wizard.

More information regarding available services and support from this provider can also be found on our website at:

http://www.webassist.com/go/linkpoint_gateway

Each of these sections links to the configuration details within the Checkout Wizard specific to each service:

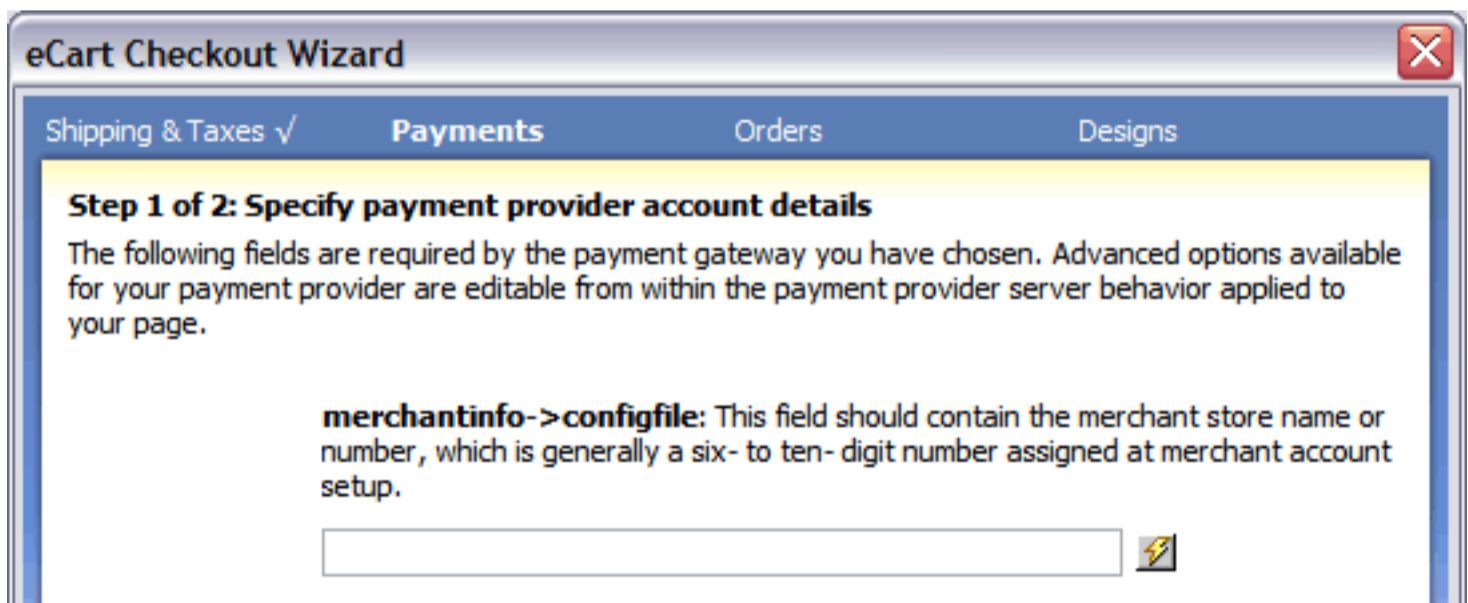
- [API](#)
- [Connect](#)

Advanced configuration

The Checkout Wizard is the most direct method for configuring payment gateway support for your shopping cart. However, advanced payment configuration may be necessary to properly complete your transactions. This can be accomplished by modifying the server behavior applied to the checkout page specific to your gateway.

For more detailed information specific to server behavior configuration and Linkpoint services, please refer to the Linkpoint support section.

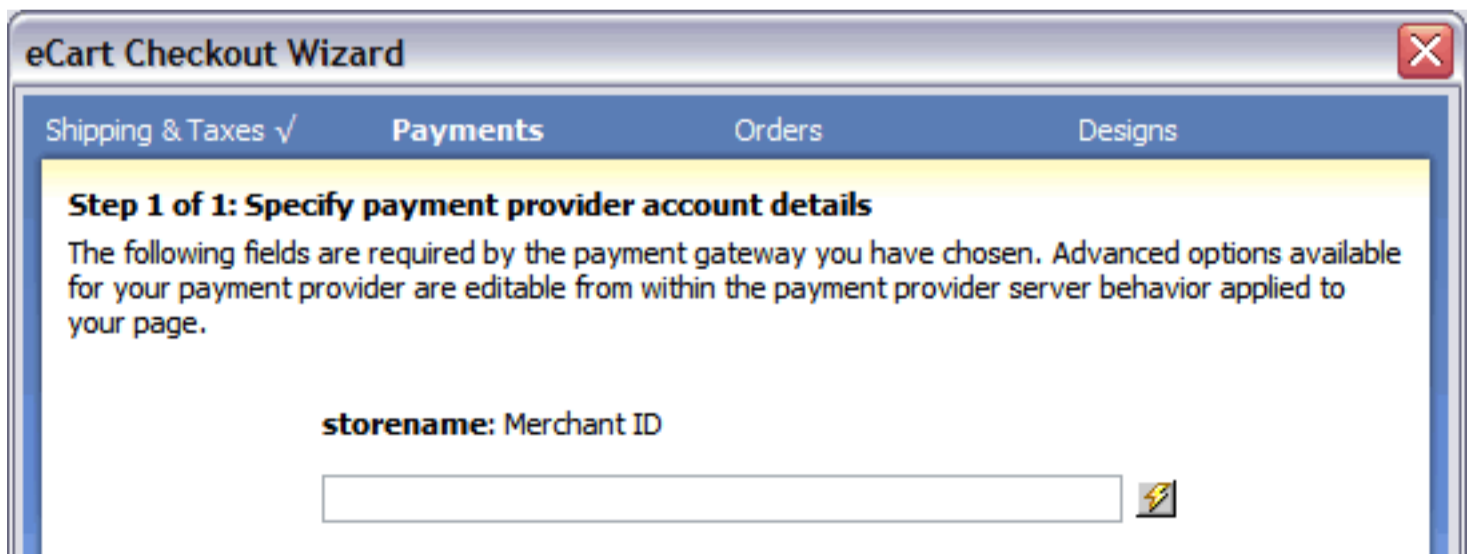
API



The screenshot shows the 'eCart Checkout Wizard' window with the 'Payments' tab selected. The title bar reads 'eCart Checkout Wizard' and the window has a close button. The navigation bar includes 'Shipping & Taxes ✓', 'Payments', 'Orders', and 'Designs'. The main content area is titled 'Step 1 of 2: Specify payment provider account details'. Below the title, it says: 'The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page.' A specific field is highlighted: 'merchantinfo->configfile: This field should contain the merchant store name or number, which is generally a six- to ten- digit number assigned at merchant account setup.' Below this text is a text input field with a lightning bolt icon to its right.

- **merchantinfo ->configfile:** The store number for your Linpoint account.

Connect



The screenshot shows the 'eCart Checkout Wizard' window with the 'Payments' tab selected. The title bar reads 'eCart Checkout Wizard' and the window has a close button. The navigation bar includes 'Shipping & Taxes ✓', 'Payments', 'Orders', and 'Designs'. The main content area is titled 'Step 1 of 1: Specify payment provider account details'. Below the title, it says: 'The following fields are required by the payment gateway you have chosen. Advanced options available for your payment provider are editable from within the payment provider server behavior applied to your page.' A specific field is highlighted: 'storename: Merchant ID'. Below this text is a text input field with a lightning bolt icon to its right.

- **storename:** The store number for your Linpoint account.
-

Orders

The Checkout Wizard directly configures the management of orders, by both storing data in a database, as well as configuring email receipts sent to the end-user (requires the installation of Universal Email, also available from WebAssist. This featured is disabled if UE is not installed).

The following sections details the steps necessary to insert orders into a database, as well as send email receipts to the end user:

- [Store order summary in database](#)
 - [Store order details in database](#)
 - [Send email receipt](#)
-

Store order summary in database

The Checkout Wizard configures the Store Cart Summary in Database server behavior to insert a new record in a database table with basic information about the order.

Databases typically use two tables to handle orders, a general Orders table and an OrderDetails table. The two tables are linked by key field, usually OrderID; each record in the OrderDetails table includes a column called something like DetailsOrderID that corresponds to the OrderID column in the Orders table.

This makes it possible to retrieve a summary of all the orders placed by a particular customer by querying just the Orders table or to get a listing of all the products on a particular order by querying the OrderDetails page.

You can also employ a database view (also called a query of a query) that combines information from the two tables.

The Store Cart Summary in Database server behavior adds records to the more general Orders table. This server behavior is used together with the Store Cart Details in Database server behavior configured in the next step to save all the cart information at the same time.

If you wish to configure order insertion in the database through the Checkout Wizard, you will need to set up your database connection and corresponding order tables in the database prior to accessing the Checkout Wizard.

To configure storing the order summary:

1. Leave the "No thanks..." field unchecked. To bypass this step and order management configuration, check this box and click the Next button. Otherwise proceed with the configuration process.
2. Select the desired connection from the Connection list.
3. Choose a table to insert the shopping cart information into from the Table list.

The table chosen should have, at a minimum, columns for an order ID (the unique key field) and order total.

Other useful columns include ones for an order reference ID, a user ID, the shipping amount and the tax on the order.

4. Select the key field for the table from the Unique ID list.

The screenshot shows the 'eCart Checkout Wizard' dialog box, specifically 'Step 1 of 2: Store order summary in database'. The dialog has tabs for 'Shipping & Taxes', 'Payments', 'Orders' (selected), and 'Designs'. The main area contains instructions: 'To store information about an order select the connection, table and the unique column for that table. Next set the value for columns in the table. Click the lightning bolt to set values from your cart.' Below this is an unchecked checkbox: 'No thanks, I'll handle order management myself'. Under the 'Order' section, there are three dropdown menus: 'Connection' (set to 'eCart'), 'Table' (set to 'Orders'), and 'Unique ID' (set to 'OrderID'). A 'Columns' table is shown with columns 'Column' and 'Value'. The 'Value' column contains dynamic expressions for 'OrderReferenceID', 'OrderUserID', 'OrderShipping', 'OrderTax', and 'OrderTotal'. Below the table is a 'Value:' field containing '<%=eCart1.GrandTotal()%' and a lightning bolt icon. At the bottom, there are buttons for 'HELP', 'BACK', 'NEXT', and 'CANCEL', and the 'WEBASSIST' logo.

Column	Value
OrderID	
OrderReferenceID	<%= Session("SessionID") %>
OrderUserID	<%= Session("UserID") %>
OrderShipping	<%=eCart1.GetShipping() %>
OrderTax	<%=eCart1.GetTax() %>
OrderTotal	<%=eCart1.GrandTotal() %>
OrderDate	
OrderStatus	

5. Bind values to the table columns listed in the columns area. Either static or dynamic values can be used.

To bind a static value, enter it in the Value field for a selected column.

To bind a dynamic value, choose the lightning bolt to open the Dynamic Data dialog box and select an item from a shopping cart entry; click OK to close the Dynamic Data dialog.

Dynamic values can also be entered by hand in the Value field.

In the example figure, the following values are assigned:

- OrderID - Left blank and not updated. OrderID is an autonumber field.
- OrderReferenceID - Choose [Cart ID] from the shopping cart dynamic data entry. The code appropriate to your server model is written to include the SessionID, a unique value.
- OrderUserID - In this example, the user has logged in and his unique user ID has been assigned to a session variable, UserID. The code for writing this session variable appropriate to the server model has been entered manually into the Value field and the binding updated.
- OrderShipping - Shipping is retrieved from the eCart object data binding for the shipping value in the cart.
- OrderTax - Tax is retrieved from the eCart object data binding for the tax value in the cart.
- OrderTotal - From the shopping cart entry of the Dynamic Data dialog, the [GrandTotal] item is selected and the appropriate code is written.

6. Click Next when you're done to proceed to configuration of insertion of the order details in the database.

You can modify this configuration after the Checkout Wizard has been completed by accessing the Store Cart Summary in Database server behavior through the Server Behaviors panel. As well, you can also apply this server behavior outside of the Checkout Wizard specific to your own needs, or to apply more advanced configuration options not immediately available in the Checkout Wizard. See [Store Cart Summary in Database server behavior](#) for more information.

Store order details in database

The Checkout Wizard configures the Store Cart Details in Database server behavior to insert a new record in a database table for each line item detail specific to the order.

Databases typically use two tables to handle orders, a general Orders table and an OrderDetails table. The two tables are linked by key field, usually OrderID; each record in the OrderDetails table includes a column called something like DetailsOrderID that corresponds to the OrderID column in the Orders table.

This makes it possible to retrieve a summary of all the orders placed by a particular customer by querying just the Orders table or to get a listing of all the products on a particular order by querying the OrderDetails page.

You can also employ a database view (also called a query of a query) that combines information from the two tables.

The Store Cart Details in Database server behavior adds records to the OrdersDetails table. This server behavior is used together with the Store Cart Summary in Database server behavior configured in the previous step to save all the cart information at the same time.

To configure storing the order details:

1. Choose a table to insert the shopping cart information into from the Table list.

The table chosen should have, at a minimum, columns for a details order ID (the unique key field), an order ID, an item ID, and quantity. Other useful columns include ones for a product name, price and selected options.

2. Select the field that contains the unique key for the order summary record in the previous step. This is how the line items can all be related back to the main details of the order.

The screenshot shows the 'eCart Checkout Wizard' dialog box, specifically 'Step 2 of 2: Store order details in database'. The dialog has tabs for 'Shipping & Taxes', 'Payments', 'Orders', and 'Designs'. The 'Orders' tab is active. The main area contains the following configuration:

Order details

Table: OrderDetails

Order ID: DetailOrderID

Column	Value
DetailsID	
DetailOrderID	<%=Session("eCart1_OrderID")%>
DetailProductID	<%=eCart1.DisplayInfo("ID")%>
DetailProductName	<%=eCart1.DisplayInfo("Name")%>
DetailProductDesc	<%=eCart1.DisplayInfo("Description")%>
DetailQuantity	<%=eCart1.DisplayInfo("Quantity")%>
DetailPrice	<%=eCart1.DisplayInfo("Price")%>
DetailOption1	
DetailOption2	

Value: <%=eCart1.DisplayInfo("Price")%>

At the bottom, there are buttons for 'HELP', 'BACK', 'NEXT', and 'CANCEL'. The 'WEBASSIST' logo is visible in the bottom left corner.

3. Bind values to the table columns listed in the columns area. Either static or dynamic values can be used.

To bind a static value, enter it in the Value field for a selected column.

To bind a dynamic value, choose the lightning bolt to open the Dynamic Data dialog box and select an item from a shopping cart entry; click OK to close the Dynamic Data dialog.

Dynamic values can also be entered by hand in the Value field.

In the example figure, the following values are assigned:

- DetailsID - Left blank and not updated. DetailsID is an autonumber field.
- DetailOrderID - Expand the Session entry in the Dynamic Data dialog and select OrderNumber. The OrderNumber session variable is defined in the Save Cart Summary in Database server behavior.
- DetailProductID - The ID column from the shopping cart dynamic data entry is selected.
- DetailProductName - The Name column from the shopping cart dynamic data entry is selected.
- DetailQuantity - The Quantity column from the shopping cart dynamic data entry is selected.
- DetailPrice - The Price column from the shopping cart dynamic data entry is selected.
- DetailOption1, DetailOption2 - No options are specified for this example, so these entries are left blank. For an entry to be left blank, the associated column in the database table must be able to accept zero length (null) values.

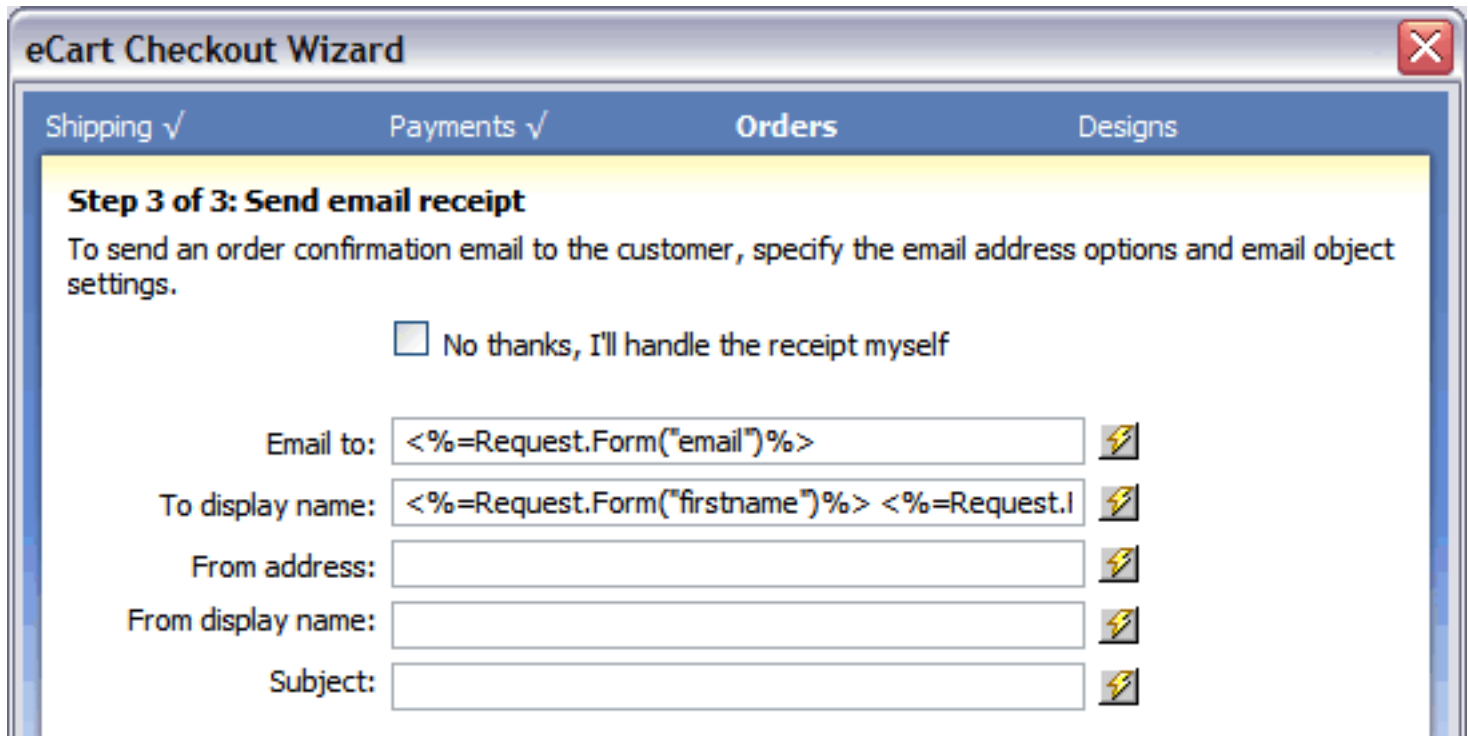
4. Click Next when you're done to proceed through the wizard.

You can modify this configuration after the Checkout Wizard has been completed by accessing the Store Cart Details in Database server behavior through the Server Behaviors panel. As well, you can also apply this server behavior outside of the Checkout Wizard specific to your own needs, or to apply more advanced configuration options not immediately available in the Checkout Wizard. See [Store Cart Details in Database server behavior](#) for more information.

Send email receipt

The Checkout Wizard configures the Universal Email server behavior to generate and send an order receipt to the customer's email address with all the order details.

The following details the general configuration requirements to support sending an email:



eCart Checkout Wizard

Shipping ✓ Payments ✓ **Orders** Designs

Step 3 of 3: Send email receipt

To send an order confirmation email to the customer, specify the email address options and email object settings.

No thanks, I'll handle the receipt myself

Email to: <%=Request.Form("email")%> ⚡

To display name: <%=Request.Form("firstname")%> <%=Request.I ⚡

From address: ⚡

From display name: ⚡

Subject: ⚡

No thanks, I'll handle the receipt myself: Enables or disables email receipt configuration through the Checkout Wizard.

Email to: Specifies the form field from the checkout form that the customer's email receipts is retrieved from (configured by default).

To display name: the email recipient header name that is displayed in the customer's email client. Defaults to the first and last name form fields retrieved from the checkout form.

From address: the email address that appears as the sender to the recipient of the email

From display name: the email sender header that describes the from address.

Subject: Sets the subject line of the email (e.g. Company XYZ Order Receipt)

Several email objects are directly supported to handle sending the email. The following sections detail the configuration options specific to each:


- [AspEmail](#)
- [AspMail](#)
- [CDONTS](#)
- [CDOSYS](#)
- [CFMAIL](#)
- [JMail](#)
- [PHPmail](#)
- [SoftArtisans SMTPmail](#)

You can modify this configuration after the Checkout Wizard has been completed by accessing the Universal Email server behavior through the Server Behaviors panel. As well, you can also apply this server behavior outside of the Checkout Wizard specific to your own needs, or to apply more advanced configuration options not immediately available in the Checkout Wizard. See Universal Email's help documentation for more information.

AspEmail

The following information is required for configuring your email application to use the AspEmail object:

Email object

Host: 
Port:
Server Login:
Password:

Host: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. `<%= "server_name"%>`).

Port: This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

Server Login and Password: These fields allow you to specify a username and password to access the SMTP server the mail object uses. Restricting SMTP access provides security against outside sources sending mail through your server. This feature is only available for the full version of the email object.

For more information and support for the AspEmail object, please refer to <http://support.persits.com/>

AspMail

The following information is required for configuring your email application to use the AspMail object:

Email object

Remote Host: 

Remote Host: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. `<%= "server_name"%>`).

For more information and support for the AspMail object, please refer to <http://www.serverobjects.com/comp/Aspmail4.htm>

CDONTS NewMail

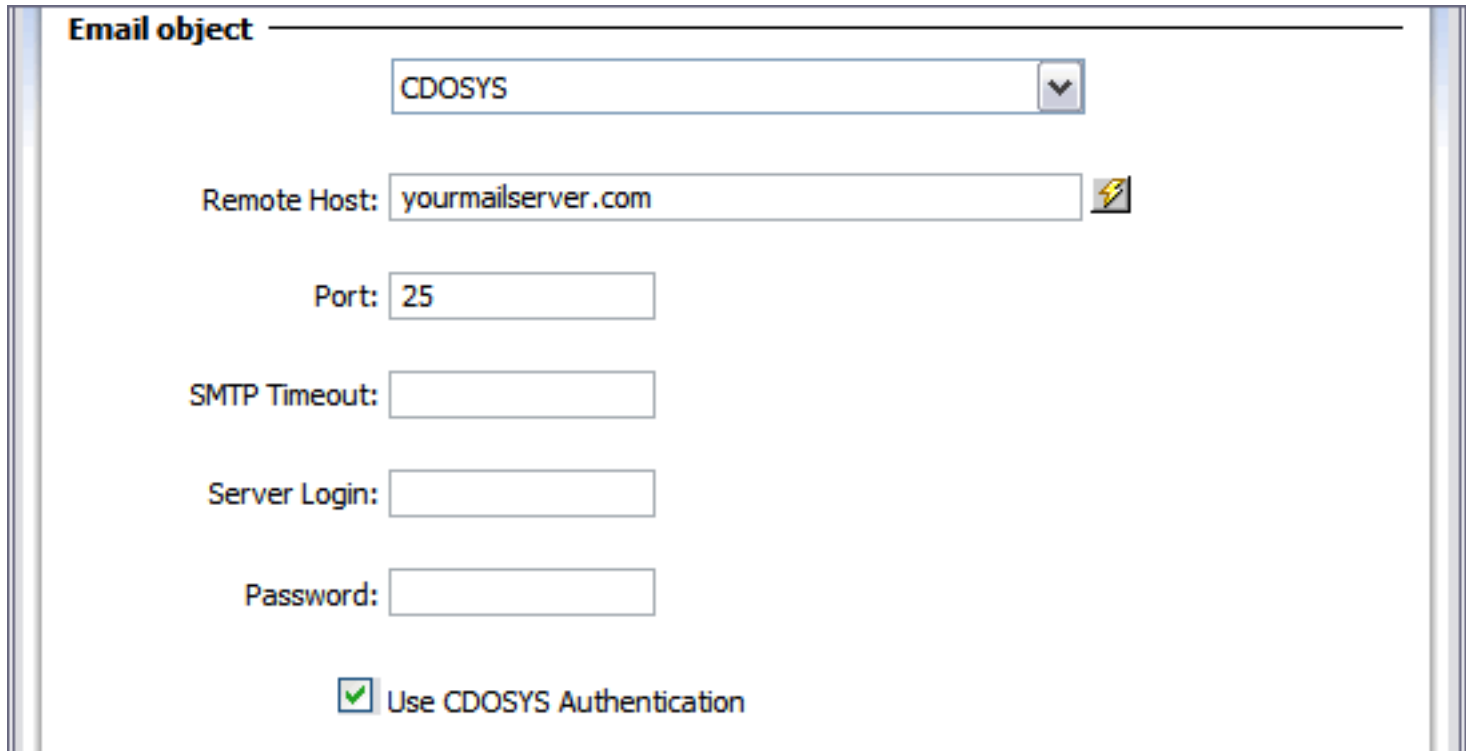
The CDONTS NewMail object is the CDO mail object utilized by Internet Information Server (IIS). It requires no additional configuration through Universal Email other than specification.

Email object
CDONTS NewMail 

For more information and support for the CDO Mail object, please refer to Microsoft's [TechNet](#) site.

CDOSYS

The following information is required for configuring your email application to use the CDOSYS object:



Email object

CDOSYS

Remote Host: yourmailserver.com

Port: 25

SMTP Timeout:

Server Login:

Password:

Use CDOSYS Authentication

Remote Host: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%= "server_name"%>).

Port: This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

SMTP Timeout: This sets the number of seconds which the email object should attempt to connect to a specified SMTP server to relay the generated email.

Server Login and Password: These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information and support for the CDOSYS object, please refer to Microsoft's [TechNet](#) site.

CFMAIL

The following information is required for configuring your email application to use the CFMAIL object:



The screenshot shows a configuration window titled "Email object". At the top, there is a dropdown menu with "CFMAIL" selected. Below this are three input fields: "Mail Server:" with a lightning bolt icon, "Port:", and "Timeout:". The "Mail Server:" field is the largest and is currently empty. The "Port:" and "Timeout:" fields are smaller and also empty.

Mail Server: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

Port: This specifies the designated port on the computer which the email object will use to send email.

Timeout: This sets the number of seconds which the email object should attempt to connect to a specified SMTP server to relay the generated email.

For more information and support for the CFMAIL object, please refer to Macromedia's Coldfusion Developer Center, located at:

<http://www.macromedia.com/desdev/mx/coldfusion/>

JMail

The following information is required for configuring your email application to use the w3 JMail object:

Email object

Mail Server:
Server Login:
Password:

Mail Server: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. `<%= "server_name"%>`).







Server Login and Password: These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information on the w3 JMail object, please refer to <http://www.dimac.net>

PHPmail

The following information is required for configuring your email application to use the native mail() functions available in PHP for both Mail for Windows and Mail for Linux:

Email object

Server Name: 
Port: 
Return Path: 
Organization: 
X-Mailer: 
Character Set: 

Server Name: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. <%= "server_name"%>).

Port: This specifies the designated port on the computer which the email object will use to send email. The default is 25 and will work for most configurations.

Return Path: Sets this header to contain definitive information about the email address and route back to the message's originator. Not required.

Organization: Sets the value for the Organization header class of the email. Not required.

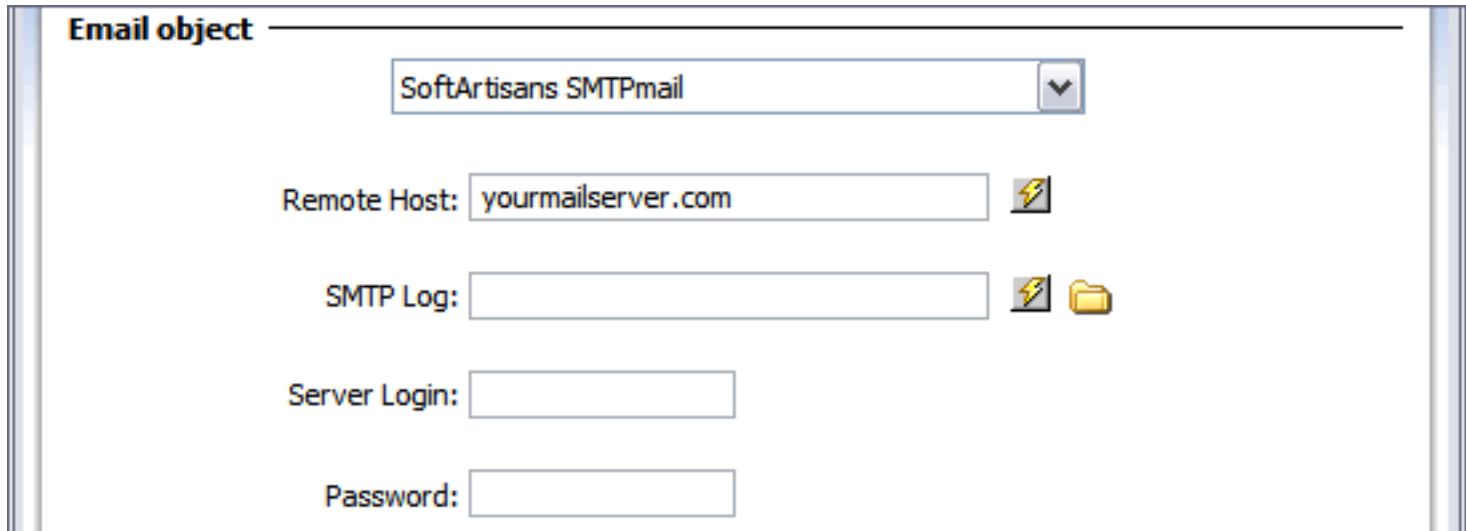
X-Mailer: Sets the header that describes the mailer program that was used to create this message. Not required.

Character Set: The character set to use for the body of the email. Default is iso-8859-1. Not required. For more information and support for the PEAR Mail object, please refer to:

<http://us3.php.net/mail>

SMTPmail

The following information is required for configuring your email application to use the SMTPmail object:



Email object

SoftArtisans SMTPmail

Remote Host: ⚡

SMTP Log: ⚡ 📁

Server Login:

Password:

Remote Host: This is the name of the SMTP server used by the object to distribute the generated email.

Valid entries for this include:

- ip address (e.g. 192.168.0.1)
- domain name (e.g. mail. server.com)
- server name, if recognized within your network

This information may also be retrieved using server side code (e.g. `<%= "server_name"%>`).

SMTP Log: This provides the name of the log file to which actions by the email object can be written. The location can be specified by using the browse button if it will be on the current machine or may be pulled from a specified recordset using the dynamic data function (this field will accept server side code).

Server Login and Password: These fields allow you to specify a username and password to access the SMTP server for use by the mail object. Restricting SMTP access provides security against outside sources sending mail through your server.

For more information and support for the SMTPmail object, please refer to: <http://support.softartisans.com/>

Designs

The Checkout Wizard directly configures the look and feel of the checkout process by selecting a style for the checkout form, as well as configuring all the necessary pages for the checkout workflow.

The following sections details the steps necessary to define the look and feel of the checkout form, as well as configure all the pages necessary to the checkout process.

- [Select design options](#)
 - [Specify pages to create](#)
-

Select design options

The Checkout Wizard specifies the look and feel of the checkout form created for your checkout process.

The following is a breakdown of each of the configuration settings available and how they affect the checkout form display. When you have made your selections, click the Next button to proceed to the next step:

- **Design:** Selects the design for the shopping cart layout. This configures the HTML structure and CSS used in the shopping cart display.
- **Color scheme:** Selects the color scheme applied to the shopping cart display from 11 available options.
- **Preview:** As well as displaying a sample based on your configuration options, clicking the preview image opens the sample design in a separate browser window.

eCart Checkout Wizard

Shipping & Taxes ✓ Payments ✓ Orders ✓ **Designs**

Step 1 of 2: Select design options
Choose a design to be added to your checkout page.

Design: Simple

Color scheme: Pacifica

PayPal Website Payments Pro US with Express Checkout

Preview

Fast, Secure Checkout with PayPal

Checkout with PayPal
Fast, easy, secure.

Save time. Checkout securely. Pay without sharing your financial information.

Pay by Credit Card

Credit card type: Visa

Credit card number

Expiration month

Expiration year

CVV code

Billing Information

First name

Last name

Email address

Phone number

Fax number

Address

Address line 2

City

State: Choose a state

Postal code

Country: Choose a country

Shipping Information

Same as billing address

Shipping first name

Shipping last name

Shipping email address

Shipping phone number

Shipping fax number

Shipping address

Shipping address line 2

Customer shipping city

Customer shipping state: Choose a state

Shipping postal code

Shipping country: Choose a country

Checkout

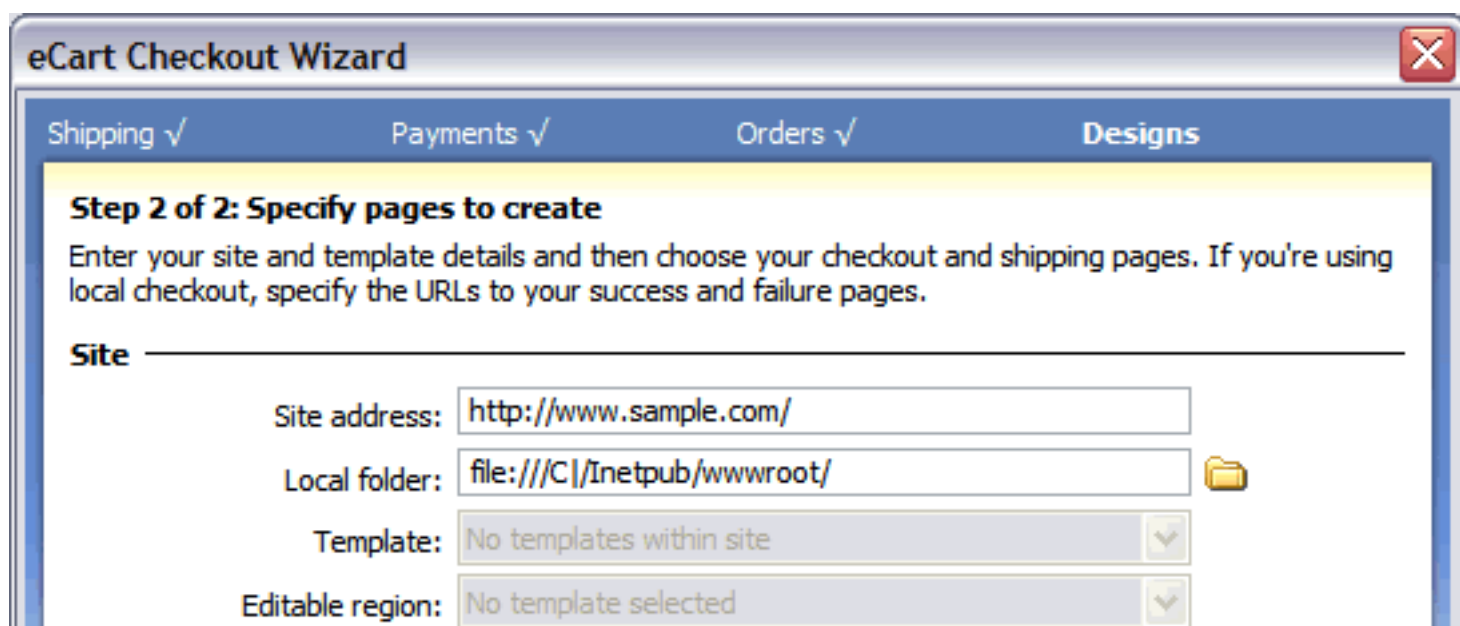
WEBASSIST

HELP BACK NEXT CANCEL




Specify pages to create

The Checkout Wizard specifies the look and feel of the checkout form created for your checkout process.

The following is a breakdown of each of the configuration settings available and how they affect the checkout form display. When you have made your selections, click the Next button to proceed to the next step:

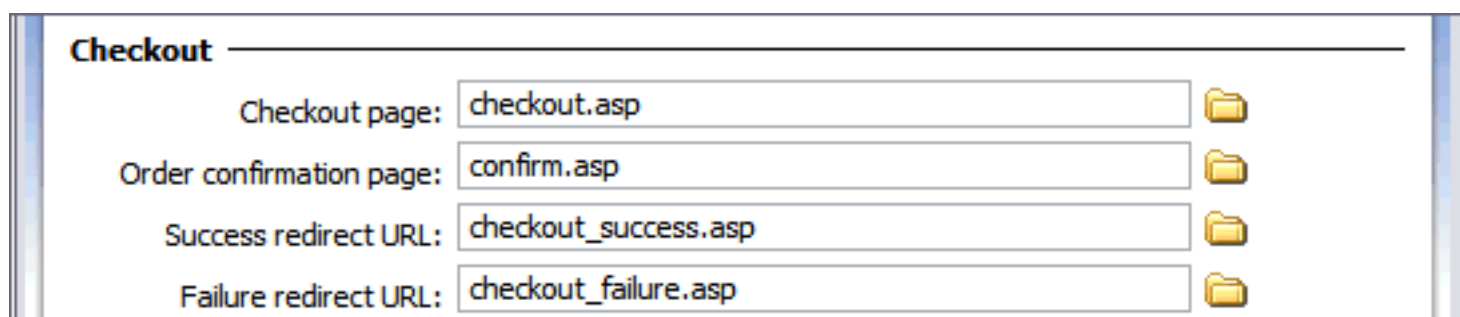


The screenshot shows the 'eCart Checkout Wizard' window with the 'Site' tab selected. The title bar includes 'Shipping ✓', 'Payments ✓', 'Orders ✓', and 'Designs'. The main content area is titled 'Step 2 of 2: Specify pages to create' and contains the following fields:





- Site address:
- Local folder: 
- Template: 
- Editable region: 

Site

- **Site address:** The domain the site is hosted at. This provides the root from which redirect URL's can be appropriately referenced for service providers (gateways, shipping) posting information back to your site.
- **Local folder:** The location where you are developing the pages locally. Provides context for links between pages that are generated.
- **Template:** If your site uses a template for its pages, you may select from available templates to be used for the new pages created by the wizard.
- **Editable region:** The location in the template specified above where the display content generated on the page should be inserted. This is typically a "main content" region.



The screenshot shows the 'Checkout' configuration section with the following fields:

- Checkout page: 
- Order confirmation page: 
- Success redirect URL: 
- Failure redirect URL: 

Checkout

- **Checkout page:** The page that contains the checkout form the customer fills out with their billing information to complete the transaction.
- **Order confirmation page:** A secondary page after submitting a completed checkout form with all required billing information. Allows the user to review data prior to final submission to the gateway.
- **Success redirect URL:** When using a local checkout service, this is the page that is redirected to upon confirmation that the transaction has processed successfully.
- **Failure redirect URL:** When using a local checkout service, this is the page that is redirected to upon confirmation that the transaction has NOT processed successfully.

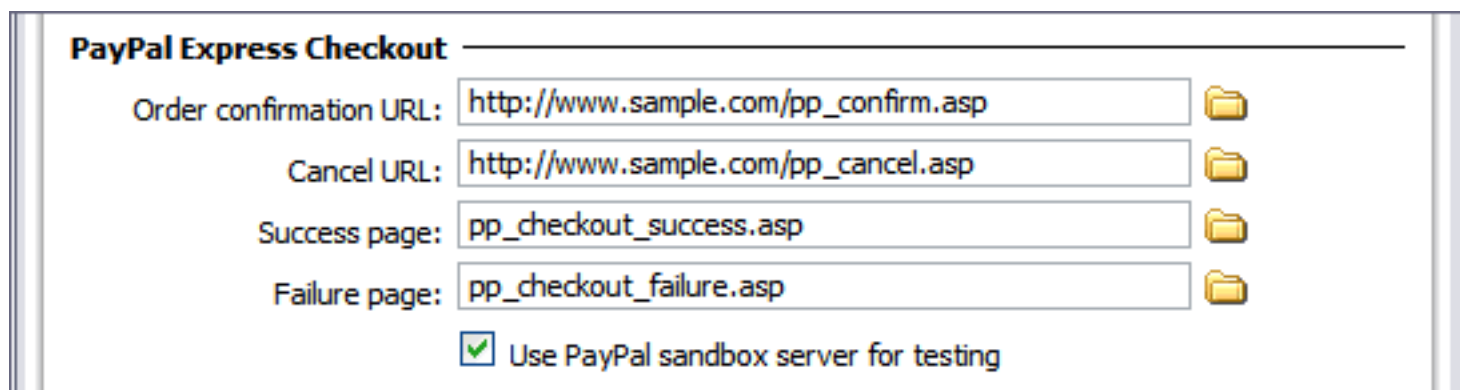


The screenshot shows the 'Shipping' configuration section with the following field:





- Failure redirect URL: 

Shipping

- **Failure redirect URL:** If a shipping service has been enabled, this option allows you to configure a redirect page if the shipping rate request is unsuccessful.



The screenshot shows the 'PayPal Express Checkout' configuration section with the following fields:

- Order confirmation URL: 
- Cancel URL: 
- Success page: 
- Failure page: 
- Use PayPal sandbox server for testing

PayPal Express Checkout

- This section is only applicable if the Express Checkout service has been enabled during the prior configuration process. This service has functionality specific to its process that requires its own page set to manage the processing component of checkout.
- **Order confirmation URL:** When Express Checkout is selected by a customer as a payment option, this is the page after logging in to PayPal that gives the customer an opportunity to review their order prior to completing the transaction.
- **Cancel URL:** The redirect location the customer is sent to if they decide to cancel the transaction prior to completing payment during the log in process on PayPal's site.
- **Success page:** When using the Express Checkout service, this is the page that is redirected to upon confirmation that the transaction has processed successfully.
- **Failure page:** When using the Express Checkout service, this is the page that is redirected to upon confirmation that the transaction has NOT processed successfully.

Note: The sandbox server testing environment will not work for ASP configurations if you do not have your SSL certificate installed on your website. Please refer to PayPal's developer center for more information.

Insert Checkout Form

Insert Checkout Form directly configures the look and feel of your checkout form, and allows you to directly configure the properties of your form specific to your payment gateway's requirements or your workflow requirements.

The following sections details the steps necessary to define the look and feel of the checkout form, as well as configure all the pages necessary to the checkout process.

- [Step 1 of 2: Choose form type](#)
- [Step 2 of 2: Select design options](#)

To access the eCart Insert Checkout Form wizard:

1. In Design View in Dreamweaver, open the page that is to hold the checkout form.
2. Access the eCart Insert Checkout Form wizard through one of the following methods:
 - eCart Insert panel



- Insert > WebAssist > eCart > Insert Checkout Form

The [next page](#) details the introduction step to begin configuring your checkout process using the eCart Insert Checkout Form wizard.

Use the Back and Next buttons to navigate the steps of the wizard, and click Finish at the end to complete the process and generate the configured form.

Step 1 of 2: Choose form type

Insert Checkout Form offers multiple configuration options for creating a checkout form specific to your needs. In addition to configuring the common local and remote checkout scenarios with available payment gateways, it also provides options for using hidden fields to process checkout information on a page posted from a different form, transferring a populated form to a secure (SSL) location, or creating a pre-checkout form that gathers information, but does not submit it immediately for processing.

Once you have configured your form, it is inserted into the page, and can then be further customized using Dreamweaver's WYSIWYG design capabilities.

This feature provides multiple checkout configurations for several merchant providers, and enables you to create and retain custom configurations as well. See [Gateway-specific configuration information](#) for details specific to the gateways supported in this feature.

eCart Insert Checkout Form

Step 1 of 2: Choose form type

Select your cart and the type of form you want to insert into your page.

Cart:

Form type:

- Local checkout – Process payments on your site
- Remote checkout – Process payments on provider's site
 - Provider: Edit
- Hidden form – Hidden fields populated from a form on another page
 - Form page:
 - Form name:
- SSL Transfer – Send cart contents to a secure URL
 - SSL URL:
- Pre-checkout form – Visible input fields

WEBASSIST

Cart: Select the eCart Object in your site specific to the shopping cart implementation for which this form is collecting information to process.

Form type: Select the type of form specific to the part of your checkout workflow you are creating. Available options are:

- **Local checkout:** Insert a checkout form that processes payments directly on your website. Upon completing the Insert Checkout Form wizard, the Local Checkout server behavior dialog is launched to configure support from available payment gateways. Please review the section [Creating a local checkout form](#) for more information regarding configuring local checkout support.
- **Remote checkout:** Inserts a checkout form that posts the contents to a specified provider. Select from a list of available providers, and configure the account info necessary to identify yourself to the gateway. Please review the section [Creating a remote checkout form](#) for more information regarding configuring remote checkout support.
- **Hidden form:** Inserts a checkout form that contains hidden fields populated by the customer information from another specified form. Select the page and the form on the specified page that the customer information is to be passed from. This is commonly used on a processing confirmation page where a customer may review their information prior to submitting.
- **SSL transfer:** Inserts a checkout form that is configured to post customer information to a specified SSL location.
- **Pre-checkout form:** Inserts a checkout form that only gathers information and is not configured to post to a specific location. Allows you to customize the form specific to your needs. Commonly used as the page to collect customer data and post it to a page that confirms the customer content and processes the data (such as the Hidden form implementation above).

Step 2 of 2: Select design options

Specify the look and feel of the checkout form created for your checkout process.

The following is a breakdown of each of the configuration settings available and how they affect the checkout form display. When you have made your selections, click Finish to insert the ofrm onto the page:

- **Design:** Selects the design for the shopping cart layout. This configures the HTML structure and CSS used in the checkout form.
- **Color scheme:** Selects the color scheme applied to the checkout from 11 available options.
- **Font:** Selects the default display font family for html text in your checkout form.
- **Button type:** Selects the type of buttons to be used in the form. Options are:
 - Image: uses image buttons that correspond to the selected Style.
 - Form: uses form buttons

Note: The PNG source files for all buttons are available in your Download Center. These files are available to modify the buttons provided in the shopping cart display using the Edit with Fireworks feature in Dreamweaver. See the Buttons section under Shopping cart display components for more information.

- **Payment:** Only available fro gateways the offer both credit card and eCheck payment options.
- **Shipping:** Includes or excludes shipping fields.
- **Preview:** As well as displaying a sample based on your configuration options, clicking the preview image opens the sample design in a separate browser window.

eCart Insert Checkout Form

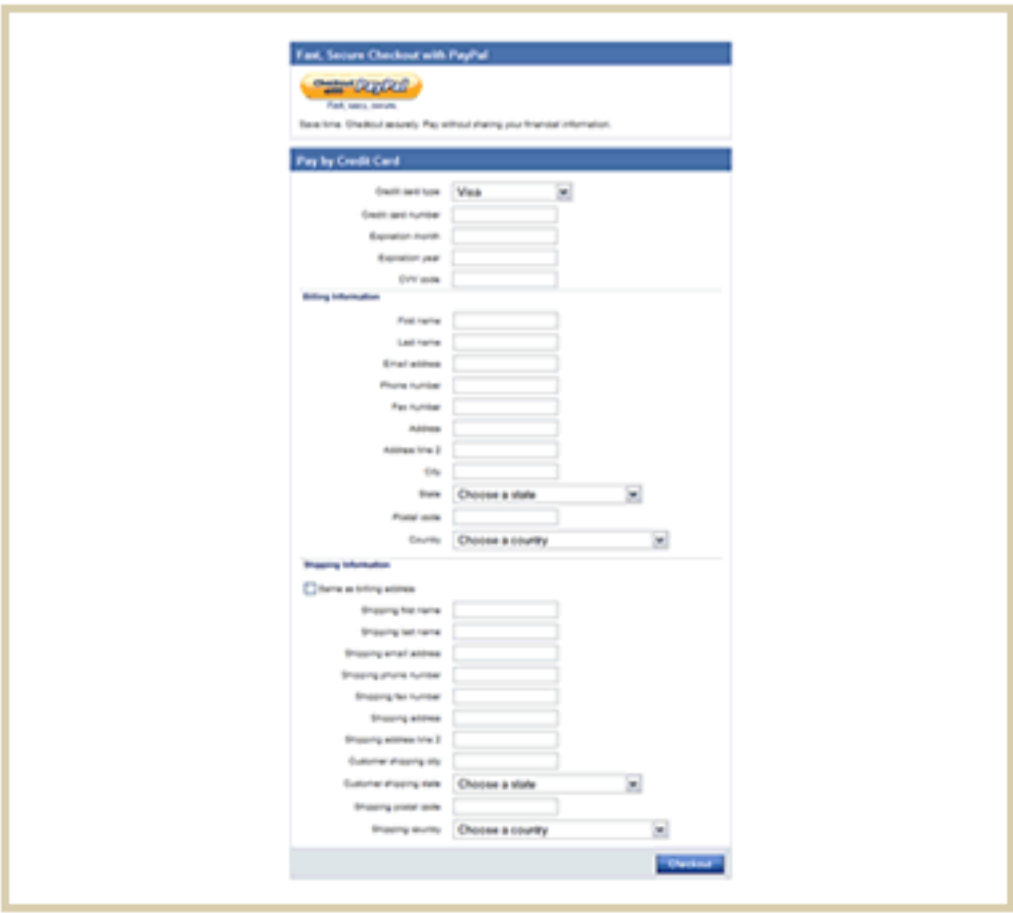
Step 2 of 2: Select design options

Select the design for your form and form field options. To preview your choices in a browser, click the preview image.

Design:	Clean	▼	Button type:	Image	▼
Color scheme:	Pacifica	▼	Payment:	Credit card	▼
Font:	Arial, Helvetica, sans-serif	▼	Shipping:	Include shipping fields	▼

Include PayPal Express Checkout button in form

Preview:



The preview shows a checkout form titled "Fast, Secure Checkout with PayPal". It includes a "Pay by Credit Card" section with fields for credit card type (Visa), number, expiration date, and CVV. Below that is a "Billing Information" section with fields for first and last name, email, phone, fax, address, city, state, and postal code. A "Shipping Information" section is also present, with a checkbox for "Same as billing address" and fields for shipping name, address, city, state, and postal code. A "Checkout" button is at the bottom right of the preview.

WEBASSIST

HELP BACK FINISH CANCEL

Creating a local checkout form

The checkout form gathers and submits all the necessary information to complete a transaction to your merchant gateway.

eCart directly supports local checkout, for five payment gateways, allowing you to complete your transactions without leaving your site. The gateways are:

- PayPal Website Payments Pro
 - Direct Payment
 - UK
 - Express Checkout
- Authorize.Net Advanced Integration Method (AIM)
- Authorize.Net SecureSource
- WebAssist Payment Gateway
- Linkpoint API
- PayFlow Pro

Each gateway has its own requirements that you need to be aware of to properly implement a checkout form to work with them. For more information specific to these payment gateways, please see the section [Local Checkout](#).

Aside from the Checkout Wizard, eCart provides three specific features to implement local checkout functionality for all the supported gateways. These are Insert Checkout Form, Local Checkout server behavior, and Local Checkout Response data bindings.

Insert Checkout Form facilitates the configuration of the variables necessary to pass credit card information to your merchant gateway within a single user interface, and then inserts the customized checkout form on your page. As well, you have the option of adding PayPal's Express Checkout option as an alternative checkout method to your customers in addition to your other payment options configured using eCart.

Local Checkout server behavior configures the necessary server-side code on your page to complete your transaction, including configuration of the variables passed from your checkout form, the shopping cart, and merchant account information, as well as the redirect destination on success and failure.

Data bindings are available to reference the data returned by your merchant gateway for use in your pages upon completion of the transaction.

The following sections detail each of these features:

- [Using Insert Checkout Form for local checkout](#)
 - [Applying the Local Checkout server behavior](#)
 - [Getting transaction details from the payment gateway](#)
-

Using Insert Checkout Form for local checkout

eCart ships with the Insert Checkout Form feature. This feature provides a simple interface for configuring a local checkout form specific to your needs to capture and pass credit card information directly to your merchant gateway to complete a transaction.

This feature works in conjunction with the [Local Checkout server behavior](#), which configures the necessary server-side code on the page to pass the information about your shopping cart, merchant account, and re-direct destination on success and failure to the gateway for processing. The server behavior configuration is launched automatically upon completing the Insert Checkout form wizard with local checkout selected as your form type.

Once you have configured your form, it is inserted into the page, and can then be further customized using Dreamweaver's WYSIWYG design capabilities.

This feature provides local checkout configurations for the following 4 merchant providers, and enables you to create and retain custom configurations as well. Links will take you to more specific information for each gateway:

- [PayPal Website Payments Pro](#)
 - [Direct Payment](#)
 - [Express Checkout](#)
- [Authorize.Net Advanced Integration method \(AIM\)](#)
- [PayFlow Pro](#)
- [WebAssist Payment Gateway](#)
- [LinkPoint API](#)

To access Insert Checkout Form:

The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- eCart Insert panel



- Insert > WebAssist > eCart > Insert Checkout Form

Note: More information on the Insert Checkout Form wizard is available [here](#).

Applying the Local Checkout server behavior

The Local Checkout server behavior configures the necessary server-side code to process a transaction from a checkout page. It works in conjunction with a checkout form created using the [Insert Checkout Form](#) feature, and the configuration dialog is launched immediately upon completing the Insert Checkout Form wizard.

Setting up the server behavior is accomplished through a five step wizard, the details of which are described below. These steps configure the variables specific to your payment gateway that need to be passed to it containing your merchant information, details about the transaction, and your credit card information. It also configures the redirect destination for a successful transaction and a failed transaction. Finally, it provides the option of configuring PayPal's Express Checkout merchant gateway as an alternate checkout method on your page.

See [Local Checkout](#) for details specific to the gateways supported in this feature.

To access the Local Checkout server behavior

The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- Server Behaviors > eCart > Checkout > Insert Checkout Form > Local Checkout configuration
- Server Behaviors > eCart > Checkout > Local Checkout

To configure the Local Checkout server behavior:

- Open the page that will submit your transaction to the merchant gateway.
- In the Server Behaviors panel, click the Plus (+) icon and select eCart > Checkout > Local Checkout

The Local Checkout dialog opens:

The dialog box is titled "Local Checkout" and shows "Step 1 of 5: Enter your general payment information". It contains a text box with instructions: "Select the payment gateway you would like to use. If you do not already have a merchant account with the chosen gateway, [sign up today!](#)". Below this are several fields: "Payment gateway:" with a dropdown menu set to "PayPal Direct Payment"; "Cart:" with a dropdown menu set to "eCart1"; "Trigger:" with a dropdown menu set to "any form post" and a lightning bolt icon; "Success redirect URL:" and "Failure redirect URL:" with empty text boxes and folder icons. At the bottom are buttons for "HELP", "BACK", "NEXT", "FINISH", and "CANCEL".

- From the payment gateway list, select your payment gateway.
- Select the eCart object on the page specific to the shopping cart for which the transaction is being processed.
- Select the trigger on the page that will submit the transaction to your gateway.
- Enter the Success and Failure redirect destinations that the end user will be directed to upon processing the transaction. This can be a relative URL or an http:// location.
- Click Next.
- The second step is to enter the required fields specific to your merchant gateway. These are required variables that must be passed in order to complete the transaction.

The dialog box is titled "Local Checkout" and shows "Step 2 of 5: Specify the required field values". It contains a text box with instructions: "The following fields are required by the payment gateway you have chosen. When you are done, click Finish to use the default configuration or click Next to adjust optional settings." Below this is a table with two columns: "Field name" and "Value".

Field name	Value
Username	test@sample.com
Password	abc123
Signature	asdas
PaymentAction	Sale
Currency	USD
IPAddress	<%=Request.ServerVariables("REMO...>
CreditCardType	<%=Request.Form("cc_type")%>
CreditCardNumber	<%=Request.Form("cc_number")%>
ExpMonth	<%=Request.Form("exp_month")%>
ExpYear	<%=Request.Form("exp_year")%>
FirstName	<%=Request.Form("firstname")%>

Below the table is a "Value:" text field containing "test@sample.com" and a "Description:" text field containing "Merchant account username with PayPal." At the bottom are buttons for "HELP", "BACK", "NEXT", "FINISH", and "CANCEL".

Variable information is updated by selecting the variable in the pane and entering its corresponding info in the Value textfield. The majority of variables already contain the required default values and do not need to be updated. Only your account specific information is required.

Descriptions of each are available at the base of the display pane by selecting the variable.

Information specific to these variables can be found in the documentation for your merchant gateway. See [Local Checkout](#) for more information.

Upon completing configuration of the variables, click Next.

- The third step is configuration of the variables that pass the shopping cart information.

The dialog box is titled "Local Checkout" and shows "Step 3 of 5: Specify order details". It contains a text box with instructions: "Click Add (+) to insert available fields for the selected gateway. Order detail fields will be repeated for each shopping cart row according to the gateway specifications." Below this are "Order Summary:" and "Order Details:" sections. The "Order Summary:" section has "Charge amount:" with a text field containing "<%=eCart1.GrandTotal()%>" and "Order description:" with an empty text field. The "Order Details:" section has "Add" and "Delete" icons. Below this is a table with two columns: "Field name" and "Value".

Field name	Value
Amount{x+1}	<%=eCart1.DisplayInfo("Price")%>
Name{x+1}	<%=eCart1.DisplayInfo("Name")%>
Quantity{x+1}	<%=eCart1.DisplayInfo("ID")%>
Tax{x+1}	<%=eCart1.DisplayInfo("Quantity")%>

Below the table is a "Field name:" text field containing "Amount{x+1}" and a "Value:" text field containing "<%=eCart1.DisplayInfo("Price")%>". Below that is a "Description:" text field containing "The item cost." At the bottom are buttons for "HELP", "BACK", "NEXT", "FINISH", and "CANCEL".

Each of these variables comes pre-populated with the server-side code specific to passing the details in your shopping cart. It is recommended that you use these default values and do not attempt to modify this configuration unless you are an experienced developer with an understanding of the architecture of eCart.

Charge amount passes the shopping cart total.

The Order ID for your transaction to uniquely identify it is based of the current server session.

The pane at the bottom contains the data for your shopping cart columns specific to each item in the cart. This configuration checks the configured column values for the specified fields and loops through all the items currently in the cart to pass the line item details to the gateway.

The default configuration contains the total cost for a quantity of a given item, the description of the items, the ID for the items, and the total number of each unique item within the shopping cart.

Additional information to pass can be added by entering it in the Field name: and Value: textfields and clicking the Plus (+) icon. Fields can be removed by selecting them within the pane and selecting the Minus (-) icon.

Upon completing configuration of the variables, click Next.

- The fourth step is to configure the optional fields that can be passed to the gateway specific to your transaction.

The dialog box is titled "Local Checkout" and shows "Step 4 of 5: Specify optional field values". It contains a text box with instructions: "Set the following fields as necessary. Refer to your payment gateway's documentation to determine which fields are applicable." Below this are "Add" and "Delete" icons. Below the icons is a table with two columns: "Field name" and "Value".

Field name	Value
MerchantSessionID	<%=Session.SessionID%>
Payer	<%=Request.Form("email")%>
Street2	<%=Request.Form("street2")%>
Phone	<%=Request.Form("phone")%>
CVV2	<%=Request.Form("cvv")%>
StartMonth	
IssueNumber	
ItemTotal	<%=eCart1.TotalColumn("TotalPrice", "")%>
ShippingTotal	<%=eCart1.GetShipping()%>
HandlingTotal	<%=eCart1.GetCharges() - eCart1.GetDiscount...>
TaxTotal	<%=eCart1.GetTax()%>

Below the table is a "Field name:" text field containing "MerchantSessionID" and a "Value:" text field containing "<%=Session.SessionID%>". Below that is a "Description:" text field containing "Session ID". At the bottom are buttons for "HELP", "BACK", "NEXT", "FINISH", and "CANCEL".

Variable information is updated by selecting the variable in the pane and entering its corresponding info in the Value textfield. Most variables already contain the required default values and do not need to be updated.

Information specific to these variables can be found in the documentation for your merchant gateway. See [Local Checkout](#) for more information.

Upon completing configuration of the variables, click Next.

- The final step of the wizard is the option to add PayPal's Express Checkout as an additional payment option on the page.

The dialog box is titled "Local Checkout" and shows "Step 5 of 5: Setup PayPal Express Checkout". It contains a text box with instructions: "Set the following fields as necessary; fields marked with an asterisk (*) are mandatory. Refer to PayPal's documentation to determine which other fields are applicable." Below this is a checkbox labeled "Use PayPal Express Checkout" which is checked. Below the checkbox is a "Trigger:" dropdown menu set to "Button: PayPal_ExpressCheckout_button pressed" with a lightning bolt icon. Below this is a table with two columns: "Field name" and "Value".

Field name	Value
Username*	test@sample.com
Password*	abc123
Signature*	asdas
Token	
Currency*	USD
OrderTotal*	<%=eCart1.GrandTotal()%>
ReturnURL*	
CancelURL*	
FailureURL	

Below the table is a "Value:" text field containing "test@sample.com" and a "Description:" text field containing "Merchant account username with PayPal." Below that is a checkbox labeled "Use PayPal sandbox server for testing" which is checked. At the bottom are buttons for "HELP", "BACK", "NEXT", "FINISH", and "CANCEL".

If you wish to add this option on the page, click the *Use PayPal Express Checkout* checkbox.

Select the trigger on the page that initiates payment processing through PayPal.

Variable information is updated by selecting the variable in the pane and entering its corresponding info in the Value textfield. Some variables already contain the required default values and do not need to be updated.

Be sure to enable the Sandbox to process test transactions and avoid posting unnecessary transactions and refunds through your live account. You can re-access the server behavior to enable your live account.

Please refer to the section detailing [PayPal Express Checkout Authentication](#) for more information on configuration of variables specific to PayPal's Express Checkout payment gateway.

Click Finish to apply the server behavior to the page.

Please note that each payment gateway has specific requirements to complete transactions and that implementing this feature is not the only step in setting up your payment gateway to process transactions. Please refer to the [Local Checkout](#) section to find more information pertinent to setting up your payment gateway for processing.

Getting transaction details from the payment gateway

All merchant gateways supported for local checkout in eCart update the current session with the information pertinent to the processing of the transaction.

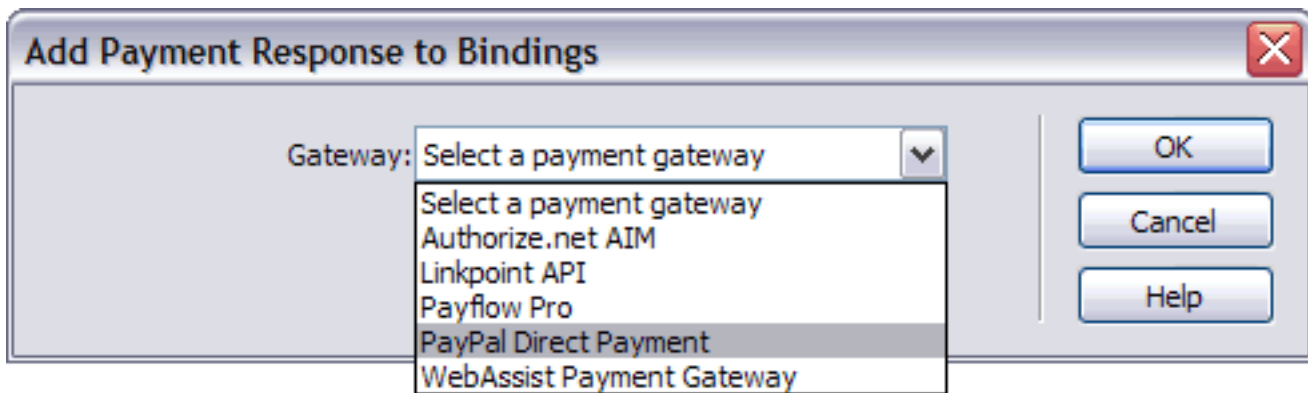
This allows you to retrieve any information you submitted to the gateway to process the transaction, as well as additional info specific to the processing of the transaction.

Bindings are available to apply to the page specific to the variables passed back by the gateway. You may use these bindings for your purposes in storing and displaying transaction data. These bindings are made available to use on the page for a specific gateway.

To make gateway data bindings available for use in a given page:

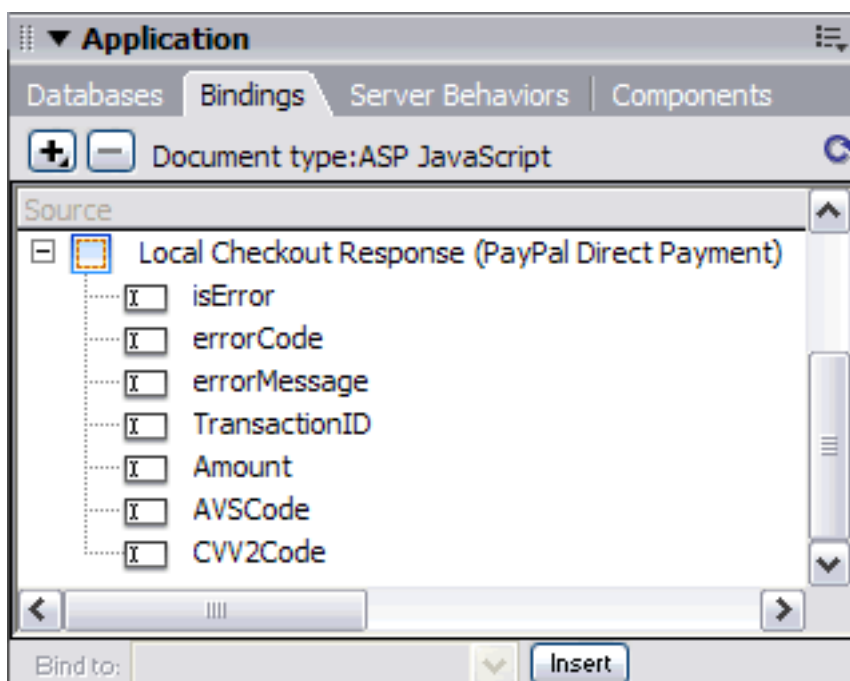
1. Open the page on which you wish to use transaction data returned from the payment gateway.
2. In the Bindings panel, click the plus button and select Local Checkout Response

The Add Payment Response to Bindings dialog opens:



3. Select your payment gateway from the list available.
4. Click OK.

The complete list of available bindings for information returned by the specified gateway is now available from your Bindings panel to use on the page. They may be inserted on the page for display, or bound as value references for elements, server behaviors, or server-side code on the page.



Creating a remote checkout form

The checkout form gathers and submits all the necessary information to complete a transaction to your merchant gateway.

eCart ships with the Insert Checkout Form feature. It facilitates the configuration of the variables necessary to pass to your merchant gateway within a single user interface, and then inserts the customized form on your page.

This feature comes with pre-configured options for several merchant gateways that supports posting to their hosted checkout forms.

You may also create your own checkout form specific to your needs.

Typically, the method of the form is *POST*, and submits this information as defined by the values of the form's fields.

The action of the form is set to the `https://` location of the merchant gateway's hosted form.

Information not controlled by the customer is passed using hidden form fields, such as the merchant account number and the total of the cart.

Information provided by the customer to the gateway is specified using text fields, checkboxes, or lists.

It is possible to create a basic checkout form that submits only the minimum required information to the gateway to complete the transaction. All additional information is then gathered by the merchant's hosted form as necessary.

This type of form can be submitted directly within the shopping cart display to replace the checkout button provided, allowing you to initiate the checkout process directly within the shopping cart display. Each merchant provider supported in eCart has a Basic configuration available for use in the Insert Search Form feature for this purpose.

Transaction data is returned to you by the merchant provider through a form POST. Data bindings are available for you to capture this information on a destination page.

Further PayPal IPN subscribers can re-populate the shopping cart based on this POST using the Get Contents From Form server behavior.

The following is a list of the pages in this section specific to helping you with the checkout form options discussed above, with links directly to each:

- [Using Insert Checkout Form for remote checkout](#)
 - [Customizing a remote checkout form](#)
 - [Getting transaction details from the payment gateway](#)
 - [Re-populating the shopping cart based on a form POST](#)
-

Using Insert Checkout Form for remote checkout

eCart ships with the Insert Checkout Form feature. This feature provides a simple interface for configuring a checkout form specific to your needs.

Once you have configured your form, it is inserted into the page, and can then be further customized using Dreamweaver's WYSIWYG design capabilities.

This feature provides checkout configurations for 6 merchant providers, and enables you to create and retain custom configurations as well.

See [Remote Checkout](#) for details specific to the gateways supported in this feature.

To access Insert Checkout Form:

The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- eCart Insert panel



- Insert > WebAssist > eCart > Insert Checkout Form

To configure and insert a checkout form:

- In Design View, place your cursor in the location that the checkout form is to be inserted.
- In the Insert panel, click the Insert Checkout Form icon.

The eCart Insert Checkout Form dialog opens:

eCart Insert Checkout Form

Step 1 of 2: Choose form type

Select your cart and the type of form you want to insert into your page.

Cart: eCart1

Form type:

- Local checkout – Process payments on your site
- Remote checkout – Process payments on provider's site
 - Provider: PayPal Website Payments Standard
- Hidden form – Hidden fields populated from a form on another page
 - Form page:
 - Form name: Select a form
- SSL Transfer – Send cart contents to a secure URL
 - SSL URL:
- Pre-checkout form – Visible input fields

WEBASSIST

HELP BACK NEXT CANCEL

- Select the eCart Object specific to this shopping cart checkout.
- Select Remote checkout for the Form type.
- From the payment gateway list, select one of the available checkout form configurations, or select Custom > New to create a new one. Custom configurations can also be saved from modifications made to provided configurations.
- Click the Edit button.

The Configure Checkout Form dialog opens:

Configure Checkout Form

Form action: https://www.PayPal.com/cgi-bin/webscr

Form Fields

+ Add - Delete

Label	Name	Default	Display
Paypal Member ID	business		hidden
Amount	amount_{x+1}	<%= WA_eCart_FormatNu...	hidden
Item Names	item_name_{x...	<%=eCart1.DisplayInfo("Na...	hidden
Item Number	item_number_...	<%=eCart1.DisplayInfo("ID...	hidden
Quantity	quantity_{x+1}	<%=eCart1.DisplayInfo("Q...	hidden
First Option Name	on0_{x+1}		hidden
First Option Value	os0_{x+1}		hidden
Second Option Name	on1_{x+1}		hidden
Second Option Value	os1_{x+1}		hidden
Amount	amount_{y+2}	<%=eCart1.GrandTotal() - ...	hidden

Label:

Name:

Default:

Display as: text field

WEBASSIST

HELP OK CANCEL

The figure above shows the PayPal With Line Items configuration.

The Form action: field at the bottom configures the action attribute of the form and establishes the http:// location that the form should POST the information to.

Within the display window are the variables that need to be passed to the merchant gateway to complete the transaction. Each variable is passed through the form POST using form elements specific to each. Each has four parameters that require configuration:

- Label:** specifies a visible text label associated to the form element submitted to the gateway. If this form element is visible to the customer and configured by their entry or selection, this label is inserted in the form to the left of the form element to identify it within the form.
- Name:** specifies the name of the variable being passed to the gateway. This applies to the name attribute of the form element that corresponds to the variable, and identifies it within the form POST. Variable names are specific to your merchant gateway configuration; see [Remote Checkout](#).
- Default:** provides the default value of a form field for a given variable. The value can be defined statically for unchanging information (e.g. account id), but can also use server-side code to dynamically retrieve information (e.g. cart total)
- Display as:** selects the type of form element inserted in the checkout form for a given variable. Options are radio fields, text fields, or select lists. Keep in mind that select lists allow you to select from multiple values accepted for a variable, but the individual options are not configured in this interface. You will need to update the list directly in Design View with any necessary options.

- To add a variable to the configuration, enter the necessary parameters for a variable within the form as described above, and click the Add (+) button. the variable will be updated within the display pane.
- To remove a variable, select it within the display panel and click the Delete button.
- To update an existing variable, select it within the display pane. Its configuration populates the parameters at the top of the interface. Make any necessary changes and click the display pane to update them.
- Use the methods outlined in steps 5-7 to configure your checkout form to your needs. You should note that at a minimum, you will need to update the variable specific to your merchant account id to be sure that you identify your account to the merchant gateway.
- Once you have completed configuration, click OK. You are prompted to save the customized configuration.
- Enter a name for the configuration and click OK. The new configuration is now available to insert as a checkout form on your page.
- Select the configuration to be inserted on the page, the cart name specific to the eCart object for your shopping cart implementation, and click OK.

The checkout form is inserted on the page.

Note: You cannot re-inspect a checkout form once it has been inserted, but can easily update variables and form elements to your needs by updating the form directly in Dreamweaver's WYSIWYG.

Customizing a remote checkout form

Your merchant provider will require that information sent to the gateway meets with the architecture of their system and is configured appropriately to work within the settings of your merchant account.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Please review the documentation specific to your merchant gateway to determine what information is necessary in your checkout form specific to your merchant account configuration.

Guidelines

The Insert Checkout Form feature creates a basic form that contains all the form elements necessary to submit a checkout.

The form is inserted into the page using a simple table structure to organize all the visible form elements and their labels.

Once the form has been inserted, you can customize the look and feel specific to your needs using Dreamweaver's design features.

The action attribute of the form needs to point to the `http://` location of your merchant gateway's hosted form to process the transaction.

You can add, remove, or modify the provided form elements as necessary. Insert Checkout Form can only create form fields of the type hidden, text field, or select list, so you may wish to modify these after the insertion.

Some variables have options to be configured that the user can select from a list, such as payment type. Insert Checkout Form cannot specify multiple options within its configuration interface for a given variable.

Variables that contain this type of information need to have the options added to the select list manually in order for the selection to pass the appropriate information based on a selection in the list. Select list options can be modified directly using the Property Inspector when a given list is selected on the page.

The value of an option needs to correspond to the available values associated to a given variable within your merchant gateway's architecture.

Insert Checkout Form also inserts a simple form button to submit the form. You may remove this and use an image form field if you don't want to use a form button. Image fields can be inserted into the form through the Dreamweaver menu *Insert > Form > Image Field*.

Note: Checkout button images specific to shopping cart displays are inserted into the `WA_eCart/Images` folder in the root of your site when you have used the eCart Display Manager to create a shopping cart display.

Getting transaction details from the payment gateway

All merchant gateways supported for remote checkout in eCart can be configured to POST transaction details to an http:// location through the merchant interface. See more information pertinent to your specific gateway in the section [Remote Checkout](#).

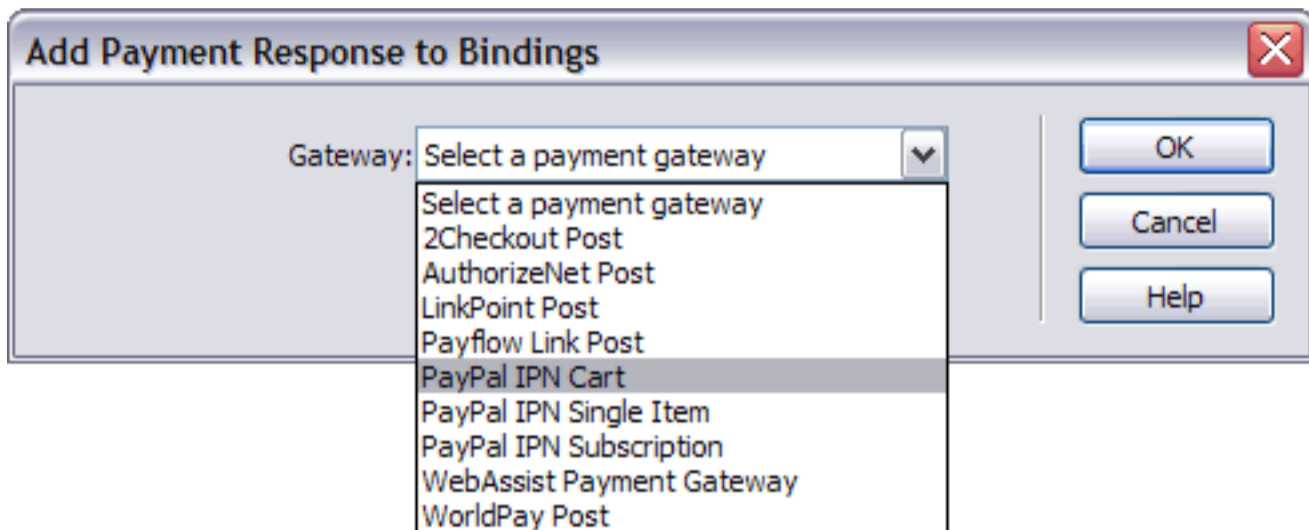
This allows you to retrieve any information you submitted to the gateway to process the transaction, as well as additional info specific to the processing of the transaction.

Bindings are available to apply to the page specific to the variables passed back by the gateway. You may use these bindings for your purposes in storing and displaying transaction data. These bindings are made available to use on the page for a specific gateway.

To make gateway data bindings available for use in a given page:

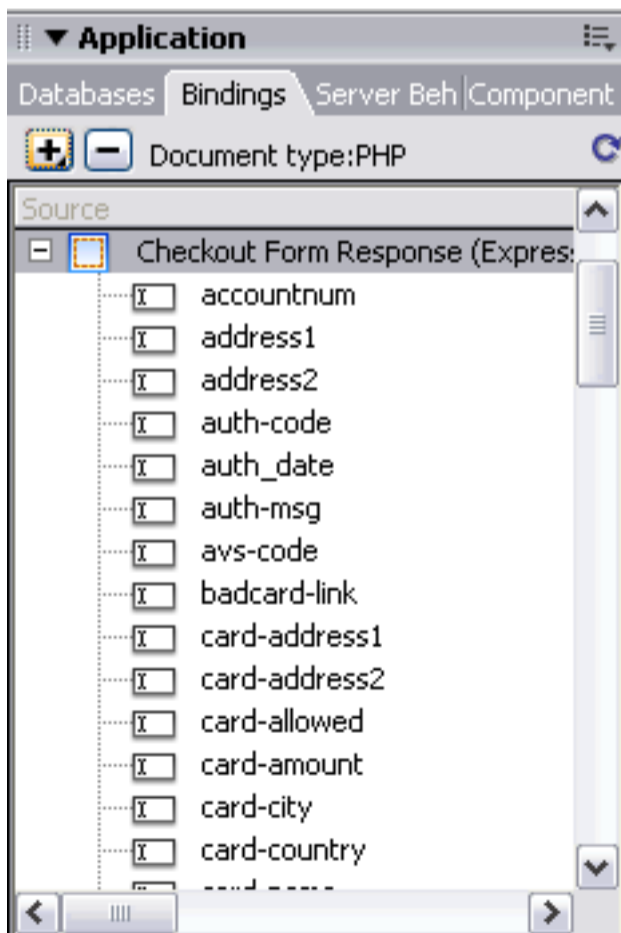
1. Open the page on which you wish to use transaction data returned from the payment gateway.
2. In the Bindings panel, click the plus button and select Checkout Form Response

The Add Payment Response to Bindings dialog opens:



3. Select your payment gateway configuration from the list available.
4. Click OK.

The complete list of available bindings for information returned by the specified gateway is now available from your Bindings panel to use on the page. They may be inserted on the page for display, or bound as value references for elements, server behaviors, or server-side code on the page.



Re-populating the shopping cart based on a form POST

The Get Contents From Form server behavior takes the data passed back from the merchant gateway and uses it to re-populate the shopping cart.

Currently, only PayPal is supported directly in eCart for re-population of shopping cart contents in this manner. PayPal's Instant Payment Notification (IPN) service posts not only the high level information for the transaction, but the line item details contained within the shopping cart. When the Get Contents From Form server behavior is run on the page, it correlates the line item details with the appropriate columns in the shopping cart.

Since the information for each cart column is available on the return post, the contents of the shopping cart can be re-created once that data has returned.

A custom option is available to configure support for gateways not directly supported in eCart.

The configure button allows you to correlate information returned to columns within the shopping cart specific to the type of POST returned by IPN. This data can be used for a shopping cart display for an order receipt, maintaining the shopping cart contents if a transaction fails, or even inserting shopping cart contents into the database.

Re-population of the shopping cart is triggered by a specified event on the page.

To apply the Get Contents From Form server behavior:

1. Open the page that will receive the POST from your merchant gateway.
2. In the Server Behaviors panel, click the plus button and select WA eCart > Add to Cart > Get Contents From Form

The Get Contents From Form dialog opens:

Get Contents From Form

Trigger: Valid Paypal Transaction

Cart: eCart1

Payment gateway: PayPal IPN Cart

Description: Creates server side code to capture returned data from a PayPal transaction including authorization status as well as credit card holder and shipping information. Auction information, if any, is also returned.

WEBASSIST

HELP OK CANCEL

3. Select the trigger on the page that will run the server-side code for this server behavior.
4. Select the shopping cart name that correlates to the eCart Object specific to your shopping cart implementation.
5. Select the IPN configuration specific to your needs from the list available.
6. If necessary, click Configure to modify the given IPN configuration specific to your needs.

The following shows the default configuration for PayPal available for re-populating the cart.

Configure Transaction Details

Cart Column	Value
ID	<%=Request.Form("item_number"+(displayInde...
Name	<%=Request.Form("item_name"+(displayIndex...
Description	
Weight	
Quantity	<%=Request.Form("quantity"+(displayIndex+1...
Price	<%=parseFloat(Request.Form("mc_gross_" + (di...

Value: <%=Request.Form("item_number"+(displayIndex

Update

OK Cancel Help

Specialized server-side code is provided for the ID, Name, Quantity, and Price columns. This code takes the line item details information, extracts values specific to line items, and correlates them to the given columns.

7. Click OK.

The server behavior is applied to the page.

Gateway-specific configuration information

This section provides resources specific to each merchant gateway directly supported by eCart.

Gateway information is divided up in to the following checkout options:

- [Local Checkout](#)
 - [Remote Checkout](#)
-

Local Checkout

This section provides resources specific to each merchant gateway directly supported by eCart for local checkout.

Each provides the location of the help documentation provided by each merchant specific to configuring the submission and retrieval of transaction information from the gateway.

In addition, the minimum requirements for sending checkout information is provided.

Information is available for each of the following supported merchant gateways:

- [PayPal Website Payments Pro](#)
 - [Authorize.Net Advanced Integration method \(AIM\)](#)
 - [Payflow Pro](#)
 - [WebAssist Payment Gateway](#)
 - [LinkPoint API](#)
 - [eWay XML](#)
-

PayPal Website Payments Pro

PayPal offers two methods of checking out under their Website Payments Pro service. The following sections detail each, and how they are supported within eCart:

- [Direct Payment](#)
 - [Express Checkout](#)
-

Direct Payment

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the Direct Payment merchant gateway available from PayPal.

The complete set of variables specific to Direct Payment is available directly from PayPal in their documentation, available in the following location:

https://www.paypal.com/IntegrationCenter/ic_direct-payment.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Direct Payment, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read PayPal's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for Direct Payment that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to PayPal.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your Direct Payment documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, Direct Payment responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Express Checkout

eCart has specific features available to configure PayPal's Express Checkout merchant services for use on your pages, as well as an alternative payment option to other payment gateways configured for your site.

Express Checkout uses a slightly different implementation strategy than other local checkout services. It uses the following server behaviors to allow your customers to login in to their PayPal account, authenticate and return information specific to the buyer to your site, and then process the transaction for that buyer upon returning. These steps are accomplished in turn by the following three server behaviors:

- [PayPal Express Checkout Authentication](#)
- [Get PayPal Express Checkout Payer Profile](#)
- [Process PayPal Express Checkout Transaction](#)

In addition, PayPal provides the following resources specific to the Express Checkout service to aid in your configuration of the service:

https://www.paypal.com/IntegrationCenter/ic_direct-payment.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Express Checkout, but you are responsible for the proper configuration of your merchant account to take advantage of optional features available through the Express checkout API.

Be sure to read PayPal's documentation to ensure that your implementation works within their acceptable parameters and requirements.

PayPal Express Checkout Authentication

PayPal Express Checkout Authentication refers to a request made to the PayPal site when a payment is to be made using the Express Checkout service that requires a PayPal user to login and verify their PayPal account. This verification then returns that information to your website and allows you initiate the payment process. The server behavior can be applied directly through the server behaviors panel, or it can be applied in the Local Checkout server behavior by enabling and configuring Express Checkout in step 5 of the wizard.

To access the PayPal Express Checkout Authentication server behavior

The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- Server Behaviors > eCart > Checkout > PayPal Express Checkout > Authentication
- Server Behaviors > eCart > Checkout > Local Checkout

To configure the PayPal Express Checkout Authentication server behavior:

1. Open the page that will initiate the checkout process.
2. In the Server Behaviors panel, click the Plus (+) icon and select eCart > Checkout > Local Checkout

The PayPal Express Checkout Authentication dialog opens:

Field name	Value
Username*	test@sample.com
Password*	abc123
Signature*	asdas
Token	
Currency*	USD
OrderTotal*	<%=eCart1.GrandTotal()%>
ReturnURL*	
CancelURL*	
FailureURL	
BuyerEmail	

3. Select the trigger on the page that will submit the authentication request to the gateway.
4. Select the eCart object on the page specific to the shopping cart for which the transaction is being processed.
5. Specify the required fields specific to your merchant gateway. These are required variables that must be passed in order to complete the transaction.

Variable information is updated by selecting the variable in the pane and entering its corresponding info in the Value textfield. Some variables already contain the required default values and do not need to be updated.

The following variables are required for configuration:

Username: The email address associated to your PayPal merchant account.

Password: The password for your PayPal merchant account.

Signature: The API Signature associated to your account that identifies you to the gateway. More information on retrieving this information is available at https://www.paypal.com/IntegrationCenter/ic_certificate.html

Currency: The currency that the transaction is to be processed in. Defaults to USD.

OrderTotal: The transaction amount to be processed.

ReturnURL: The http location on your website that the authentication information is to be returned to. This is also the location where the [Get PayPal Express Checkout Payer Profile](#) is applied to continue the payment based on the information returned.

CancelURL: The http location where customers who cancel the transaction while on PayPal's site are to be re-directed to.

6. If you are testing your configuration using PayPal's sandbox and the account is not live, you may specify this within your configuration by checking *Use PayPal sandbox server for testing*.
7. Click Finish to apply the server behavior to the page.

Note: The sandbox server testing environment will not work for ASP configurations if you do not have your SSL certificate installed on your website. Please refer to PayPal's developer center for more information.

Get PayPal Express Checkout Payer Profile

This server behavior retrieves the information specific to a customer once they have authenticated on the PayPal site. The authentication process is initiated by the PayPal Express Checkout Authentication server behavior, which is configured to redirect to a location where the payer profile server behavior is applied to continue the checkout process.

To access the Get PayPal Express Checkout Payer Profile server behavior

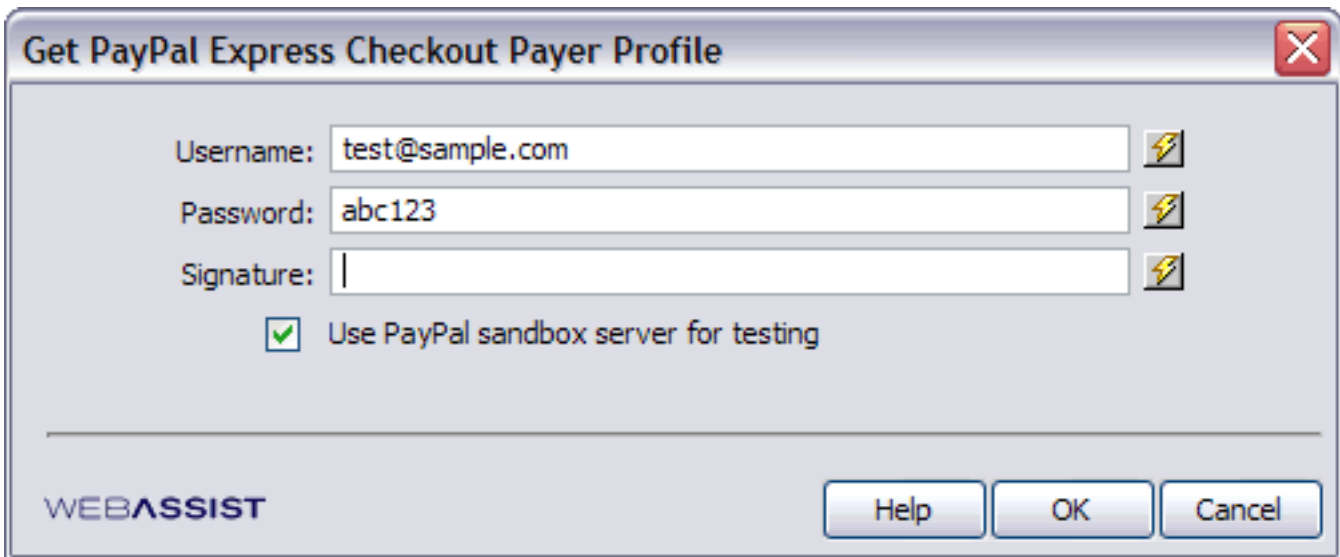
The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- Server Behaviors > WA eCart > Checkout > PayPal Express Checkout > Get Payer Profile

To configure the Get PayPal Express Checkout Payer Profile server behavior:

1. Open the page that will initiate the checkout process.
2. In the Server Behaviors panel, click the Plus (+) icon and select WA eCart > Checkout > PayPal Express Checkout > Get Payer Profile

The Get PayPal Express Checkout Payer Profile dialog opens:



3. **Username:** The email address associated to your PayPal merchant account.

Password: The password for your PayPal merchant account.

Signature: The API Signature associated to your account that identifies you to the gateway. More information on retrieving this information is available at https://www.paypal.com/IntegrationCenter/ic_certificate.html

4. If you are testing your configuration using PayPal's sandbox and the account is not live, you may specify this within your configuration by checking *Use PayPal sandbox server for testing*.
5. Click Finish to apply the server behavior to the page.

Note: The sandbox server testing environment will not work for ASP configurations if you do not have your SSL certificate installed on your website. Please refer to PayPal's developer center for more information.

Process PayPal Express Checkout Transaction

This server behavior completes the payment process once a customer has been authenticated on PayPal's site and their payer profile has been retrieved using the [Get PayPal Express Checkout Payer Profile server behavior](#). The transaction information is then submitted to PayPal to complete the payment.

To access the Process PayPal Express Checkout Transaction server behavior

The Insert Checkout Form feature is accessible in the following locations in Dreamweaver:

- Server Behaviors > WA eCart > Checkout > PayPal Express Checkout > Process Transaction

To configure the Process PayPal Express Checkout Transaction server behavior:

1. Open the page that will complete the checkout process.
2. In the Server Behaviors panel, click the Plus (+) icon and select WA eCart > Checkout > PayPal Express Checkout > Process Transaction

The PayPal Express Checkout Authentication dialog opens:

Process PayPal Express Checkout Transaction

Step 1 of 3: Enter your general payment information
Please enter some information about your PayPal Express Checkout account. If you don't have an account, [sign up today!](#)

Cart name: WA_eCart1
Trigger: any form post
Username: test@sample.com
Password: abc123
Signature: |
Token: <%=Session("PayPal_ECO_Token")%>
Payer ID: <%=Session("PayPal_ECO_PayerID")%>
Success redirect URL:
Failure redirect URL:
 Use PayPal sandbox server for testing

WEBASSIST Help < Back Next > Finish Cancel

3. Select the eCart object on the page specific to the shopping cart for which the transaction is being processed.
 4. Select the trigger on the page that will submit the authentication request to the gateway.
 5. In the Username field, specify the email address associated to your PayPal merchant account.
 6. In the Password field, specify the password for your PayPal merchant account.
 7. In the Signature field, specify the API Signature associated to your account that identifies you to the gateway. More information on retrieving this information is available at https://www.paypal.com/IntegrationCenter/ic_certificate.html
 8. The Token and Payer ID fields should be pre-populated with the necessary code to retrieve the Payer Token and ID based on the [Get PayPal Express Checkout Payer Profile server behavior](#) that is configured in conjunction with this checkout process.
 9. Specify the redirect locations based on a successful or unsuccessful transaction processing.
 10. If you are testing your configuration using PayPal's sandbox and the account is not live, you may specify this within your configuration by checking *Use PayPal sandbox server for testing*.
- Note:* The sandbox server testing environment will not work for ASP configurations if you do not have your SSL certificate installed on your website. Please refer to PayPal's developer center for more information.
11. Click Next to proceed to step two.

Process PayPal Express Checkout Transaction

Step 2 of 3: Specify order details
Select a field name to modify it. Order detail fields will be repeated for each shopping cart row according to the gateway specifications.

Order Summary:
Charge amount: <%=WA_eCart1.GrandTotal()%>
Order description:

Order Details:

Field name	Value
Amount{x+1}	<%=WA_eCart1.DisplayInfo("Price")%>
Name{x+1}	<%=WA_eCart1.DisplayInfo("Name")%>
Number{x+1}	<%=WA_eCart1.DisplayInfo("ID")%>
Quantity{x+1}	<%=WA_eCart1.DisplayInfo("Quantity")%>
Tax{x+1}	

Field name: Amount{x+1} Value: <%=WA_eCart1.DisplayInfo("Price")%>
Description: The item cost.

WEBASSIST Help < Back Next > Finish Cancel

12. This step is pre-configured to pass the line item details to PayPal based on the contents of your shopping cart. Advanced users familiar with the code being used may choose to modify this configuration based on their needs, but less experienced users should use the default values provided.
13. Click next to proceed to step 3.

Process PayPal Express Checkout Transaction

Step 3 of 3: Specify optional field values
Set the following fields as necessary. Refer to the Express Checkout documentation to determine which fields are applicable.

Field name	Value
PaymentAction	Sale
Currency	USD
ItemTotal	<%=WA_eCart1.TotalColumn("TotalPrice","")%>
TaxTotal	<%=WA_eCart1.GetCharges() - (-1*WA_eCart...
SH_Name	<%=Request.Form("sh_name")%>
SH_Street1	<%=Request.Form("sh_street1")%>
SH_Street2	<%=Request.Form("sh_street2")%>
SH_CityName	<%=Request.Form("sh_cityname")%>
SH_StateOrProvince	<%=Request.Form("sh_state_province")%>
SH_PostalCode	<%=Request.Form("sh_postcode")%>
SH_Country	<%=Request.Form("sh_country")%>
ShippingTotal	

Field name: PaymentAction Value: Sale
Description: How you want to obtain payment: Authorization, Order, or Sale

WEBASSIST Help < Back Next > Finish Cancel

14. Additional information can be configured to be passed to PayPal specific to your needs. More information on available fields for configuration can be found at https://www.paypal.com/IntegrationCenter/ic_pro_home.html
15. Click Finish to apply the server behavior to the page.

Authorize.Net Advanced Integration Method (AIM)

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the Authorize.Net merchant gateway for the Advanced Integration Method (AIM).

The complete set of variables specific to Authorize.Net's AIM is available directly from Authorize.Net in their documentation when you have set up an Authorize.Net account. This information is available when you login to their website in the following location:

<https://secure.authorize.net/>

In the *Help* section of the site (accessible through the left hand navigation) click the *Documentation and Reference Guides* link at the bottom of the page, and then click on the Advanced Integration Method (AIM) link in the following page.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Authorize.Net, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read Authorize.Net's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Server Requirements

Specific to your web application server model, the following components/libraries need to be installed in order to support using AIM in your website. You will need to confirm with your hosting provider that your server supports the following for your model:

- ASP - Msxml2.0
- PHP - curl library

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for Authorize.Net that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to Authorize.Net.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your AIM documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, Authorize.Net responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Authorize.Net also performs a form POST containing the information pertinent to your transaction to an http:// location. You must enable the POST of this information and configure the passback location in your Authorize.Net account in the following location for this information to be returned to your site:

1. Go to <https://secure.authorize.net/> and login with your merchant account information.
2. In the left navigation, click the *Settings* link.
3. Under the *Transaction Response* section, click the *Relay Response* link.
4. Specify the http:// location that the transaction details should be posted to upon completion of the transaction

Note: Your merchant account has a security setting that checks the http:// location it POST's transaction details to. Therefore, for your checkout post to be accepted by the gateway, you must specify the http:// location of the response page. This is configured in the following location:

1. Go to <https://secure.authorize.net/> and login with your merchant account information.
 2. In the left navigation, click the *Settings* link.
 3. Under the *Transaction Response* section, click the *Response/Receipt URLs* link.
 4. Specify the http:// location that the transaction details should be posted to upon completion of the transaction
-

Payflow Pro

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the Payflow Pro merchant gateway available from PayPal.

The complete set of variables specific to Payflow Pro is available directly from PayPal in their documentation when you have set up an Payflow Pro account. This information is available in the following location:

https://www.paypal.com/IntegrationCenter/ic_payflowpro.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Payflow Pro, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read PayPal's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Server Requirements

The appropriate SDK needs to be downloaded, installed and configured on your web application server. You will need to work with your hosting provider to confirm that your application server is configured correctly for your site to process transactions using this gateway.

The SDK's are available for download through your [PayPal Manager](#) in the Downloads section once you have signed up for an account with PayPal. It is included in the Payflow Pro Software Development Kit for Windows.

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for Payflow Pro that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to Verisign.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your Payflow Pro documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, Payflow Pro responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

WebAssist Payment Gateway

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the WebAssist Payment Gateway merchant gateway available from WebAssist.

The complete set of variables specific to WebAssist Payment Gateway is available directly from WebAssist in their documentation when you have set up an WebAssist Payment Gateway account. This information is available upon logging in to your merchant interface by clicking the Documentation link under Integration and Testing. The file specific to configuring local checkout is called *Remote Client Specifications*.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using WebAssist Payment Gateway, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read WebAssist Payment Gateway's documentation to ensure that your implementation works within the acceptable parameters and requirements.

Server Requirements

Specific to your web application server model, the following components/libraries need to be installed in order to support using WebAssist Payment Gateway in your website. You will need to confirm with your hosting provider that your server supports the following for your model:

- ASP - Msxml2.0
- PHP - curl library

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for WebAssist Payment Gateway that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to WebAssist Payment Gateway.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your WebAssist Payment Gateway documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, WebAssist Payment Gateway responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

LinkPoint API

The complete set of variables specific to LinkPoint API is available directly from LinkPoint in their documentation when you have set up a LinkPoint API account. This information is available when you logon to LinkPoint Central. From the top navigation, access the Support section. On the redirected page, look under Manuals > LinkPoint API. You'll have the option of getting the help in HTML or PDF format.

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the LinkPoint API merchant gateway available from LinkPoint.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using LinkPoint API, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read LinkPoint's documentation to ensure that your implementation works within their acceptable parameters and requirements.

In addition, the following resource may prove valuable in testing your Linkpoint account and provides a variety of resources specific to Linkpoint testing. Please note that this site is not supported directly by Linkpoint services:

<http://www.sgserror.com/staging.php>

Server Requirements

The LpiCom_6_0 object needs to be installed on your web application server. You will need to work with your hosting provider to confirm that this object is installed on your web server for your site to process transactions using this gateway.

Because LinkPoint API is dependent on this COM object, LinkPoint API is not supported on PHP hosted on a Unix server.

The COM control is available for download through your LinkPoint Manager in the Downloads section once you have signed up for an account with LinkPoint. It is included in the LinkPoint API Software Development Kit for Windows.

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for LinkPoint API that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to LinkPoint.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your LinkPoint API documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, LinkPoint API responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

eWay XML

The complete set of variables specific to eWay XML is available directly from eWay in their documentation on their website:

<http://www.eway.com.au/support/xml.aspx>

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the eWay XML merchant gateway available from eWay.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using eWay XML, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read eWay's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Server Requirements

eWay requires that you use a secure connection (HTTPS) to transmit transaction data. This means you will need to get an SSL certificate to secure at a minimum, the checkout page on your website.

Submitting checkout information

The [Insert Checkout Form](#) feature of eCart provides a Local Checkout configuration for eWay XML that directly configures and inserts a checkout form on your page, which can then be customized to meet your needs.

The [Local Checkout server behavior](#) configures the server-side code necessary on your checkout page to pass your merchant information, shopping cart data, and customer payment information directly to eWay.

Local Checkout details the minimum variable requirements for you during the configuration process. As well, all optional variables are available for configuration. Please consult your eWay XML documentation to determine which optional variables may be necessary for you to configure specific to your development needs.

Retrieving transaction information from the payment gateway

When you submit a transaction for processing, eWay XML responds to the same page and returns the transaction information directly to your server session, allowing you to capture the transaction results for use on your site.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Remote Checkout

This section provides resources specific to each merchant gateway directly supported by eCart for remote checkout.

Each provides the location of the help documentation provided by each merchant specific to configuring the submission and retrieval of transaction information from the gateway.

In addition, the minimum requirements for sending checkout information is provided, as well as the necessary steps to configure your account to send transaction information back to your website.

Information is available for each of the following supported merchant gateways:

- [2Checkout](#)
 - [Authorize.Net Simple Integration method \(SIM\)](#)
 - [WebAssist Payment Gateway](#)
 - [PayPal Website Payments Standard](#)
 - [LinkPoint Connect](#)
 - [Payflow Link](#)
 - [WorldPay](#)
 - [eWay Shared](#)
-

2Checkout

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the 2Checkout merchant gateway.

The complete set of variables specific to 2Checkout's merchant gateway are available directly from 2Checkout in their documentation when you have set up a 2Checkout account. This information is available when you login to their website in the following location:

<https://sellers.2checkout.com/cgi-bin/sellersarea/cartpassback.2c>

Important: Be sure to configure your 2Checkout account to Test Mode to allow you to test your checkout implementations without processing any real transactions.

Submitting checkout information

To initiate payment processing using 2Checkout, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

2Checkout is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use 2Checkout's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to 2Checkout, as defined in the 2Checkout Basic configuration:

- **2Checkout hosted form URL:**
<https://www.2checkout.com/cgi-bin/sbuyers/cartpurchase.2c>
- **sid:** your 2Checkout.com account number
- **cart_order_id:** unique order id from your cart or other program. The configurations provided in eCart use the SessionID as this unique identifier, but can be changed specific to your architecture.
- **total:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.

Please refer to 2Checkout's documentation for the complete set of variables available for 2Checkout configuration.

Retrieving transaction information from the payment gateway

2Checkout performs a form POST containing the information pertinent to your transaction to an http: location. You must enable the POST of this information and configure the passback location in your 2Checkout account in the following location for this information to be returned to your site:

<https://sellers.2checkout.com/cgi-bin/sellersarea/cartdetails.2c>

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially. Please refer to the help documentation referenced at the top of this page for a complete list of the variables available.

Authorize.Net Simple Integration Method (SIM)

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the Authorize.Net merchant gateway.

eCart directly supports Authorize.Net's Simple Integration Method (SIM) for processing transactions.

The complete set of variables specific to Authorize.Net's SIM is available directly from Authorize.Net in their documentation when you have set up an Authorize.Net account. This information is available when you login to their website in the following location:

<https://secure.authorize.net/>

Click on the *Documentation and Reference Guides* link at the bottom of the page, and then click on the Simple Integration Method (SIM) link in the following page.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Authorize.Net, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read Authorize.Net's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using Authorize.Net, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

Authorize.Net is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use Authorize.Net's hosted checkout form to gather any additional information on their website.

Insert Checkout Form provides three pre-configured checkout forms for your convenience that be inserted on the page and configured to your needs. These are:

- **Authorize.Net AIM:** provides the minimum information necessary to initiate checkout. All necessary transaction information is gathered using Authorize.Net's hosted checkout form.
- **Authorize.Net Credit Card:** configures all possible variables that can be passed to Authorize.Net for credit card payment processing, except form fields for credit card numbers and CV2 (these can be added if necessary, but should only be used if your checkout form is located in a secure website location, e.g. https://). This pre-populates the checkout form hosted on Authorize.Net's website to complete the transaction.
- **Authorize.Net ECheck:** configures all possible variables that can be passed to Authorize.Net for ECheck payment processing, except form fields for banking routing and account information (these can be added if necessary, but should only be used if your checkout form is located in a secure website location, e.g. https://). This pre-populates the checkout form hosted on Authorize.Net's website to complete the transaction.
- **SecureSource Credit Card:** configures all possible variables that can be passed to Authorize.Net for SecureSource credit card payment processing, except form fields for credit card numbers and CV2 (these can be added if necessary, but should only be used if your checkout form is located in a secure website location, e.g. https://). This pre-populates the checkout form hosted on Authorize.Net's website to complete the transaction.
- **SecureSource ECheck:** configures all possible variables that can be passed to Authorize.Net for SecureSource ECheck payment processing, except form fields for banking routing and account information (these can be added if necessary, but should only be used if your checkout form is located in a secure website location, e.g. https://). This pre-populates the checkout form hosted on Authorize.Net's website to complete the transaction.

The following is the minimum information necessary to pass to Authorize.Net, as defined in the Authorize.Net Basic configuration:

- **Authorize.Net hosted form URL:**
https://secure.authorize.net/gateway/transact.dll
- **x_login:** your Authorize.Net account number
- **txnkey:** the txnkey specific to your Authorize.Net account. You must request this key from Authorize.Net through the merchant interface.
- **x_fp_sequence:** a unique number specific to the transaction. Used to create transaction fingerprint.
- **x_fp_timestamp:** this is the timestamp of the fingerprint of the transaction.
- **x_hash:** this is the fingerprint of the transaction, which is created using a set of functions that hashes the x_login, txnkey, x_fp_sequence, and x_fp_timestamp into a secure string, which Authorize.Net receives and validates your identity as a merchant.
- **x_test_request:** tells the gateway whether the transaction submitted is a test transaction or a live transaction. For all provided configurations, this variable is set to TRUE to allow you to perform testing for the implemented form. When pushing the site live and processing live transactions, the value for this field should be set to FALSE.
- **x_show_form:** tells the gateway if it should display the checkout form when redirected to Authorize.Net's website to complete the transaction. All configurations provided by eCart have a value for this field of PAYMENT_FORM which displays the form for checkout.
- **x_relay_response:** tells the gateway to POST transaction information to an http:// location configured within your merchant account directly through Authorize.Net's website. The default value for all provided configurations is TRUE, enabling this feature of the merchant account.
- **x_amount:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.
- **x_method:** sets whether the payment method is by credit card or by check. The available variables are CC and ECHECK. The Basic and Complete configurations have a default of CC, and the ECheck configuration has a default of ECHECK. You may also customize your checkout form to pass this value from a select list form element, selecting the payment method prior to redirecting to the Authorize.Net hosted form. If this value is not set, the option will be provided in Authorize.Net's hosted form.
- **x_type:** sets the type of transaction being passed to the merchant gateway. All configurations have a default of AUTH_CAPTURE, the type necessary to complete a payment transaction.

Please refer to Authorize.Net's SIM documentation for the complete set of variables available for checkout configuration.

Note: The provided Authorize.Net configurations support the **x_hash** submission requirement. Supporting files are inserted into your site at the time you insert a checkout form using the Insert Checkout Form feature, and are located in a subdirectory of your WA_eCart folder. Keep this in mind when transferring files to your webserver as they are required for the checkout to work appropriately.

Further, PHP users need to ensure that their server supports *mhash* for their checkout form to work appropriately with Authorize.Net. Information regarding *mhash* is available at:

<http://us3.php.net/mhash>

Retrieving transaction information from the payment gateway

Authorize.Net performs a form POST containing the information pertinent to your transaction to an http:// location. You must enable the POST of this information and configure the passback location in your Authorize.Net account in the following location for this information to be returned to your site:

1. Go to <https://secure.authorize.net/> and login with your merchant account information.
2. In the left navigation, click the *Settings* link.
3. Under the *Transaction Response* section, click the *Relay Response* link.
4. Specify the http:// location that the transaction details should be posted to upon completion of the transaction

Note: Your merchant account has a security setting that checks the http:// location it POST's transaction details to. Therefore, for your checkout post to be accepted by the gateway, you must specify the http:// location of the response page. This is configured in the following location:

1. Go to <https://secure.authorize.net/> and login with your merchant account information.
2. In the left navigation, click the *Settings* link.
3. Under the *Transaction Response* section, click the *Response/Receipt URLs* link.
4. Specify the http:// location that the transaction details should be posted to upon completion of the transaction

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the Authorize.Net gateway.

WebAssist Payment Gateway

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from WebAssist's WebAssist Payment Gateway WebAssist Payment Gateway merchant gateway.

The complete set of variables specific to WebAssist Payment Gateway is available directly from WebAssist in their documentation when you have set up an WebAssist Payment Gateway account. This information is available upon logging in to your merchant interface by clicking the Documentation link under Integration and Testing. The file specific to configuring local checkout is called *Integration Specifications*. The pertinent portion of the document is *Section 2. Direct Method*.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using WebAssist Payment Gateway, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read WebAssist Payment Gateway's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using WebAssist Payment Gateway, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

WebAssist Payment Gateway is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use WebAssist Payment Gateway's hosted checkout form to gather any additional information on their website.

The profile comes pre-configured to pass standard eCart implementations' data to WebAssist Payment Gateway for processing. The only data required to be entered is your *Merchant ID*, which should be configured in the *publisher_name* variable. This is necessary to identify your merchant account to the gateway.

Please refer to the WebAssist Payment Gateway documentation for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

WebAssist Payment Gateway performs a form POST containing the information pertinent to your transaction to an http:// location. To pass this information back to your site, you must pass the POST URL to the gateway at the time the transaction is submitted.

- For successful transactions, the POST URL is set in the variable *success-link*
- For failed transactions, the POST URL is set in the variable *badcard-link*

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the WebAssist Payment Gateway gateway.

PayPal Website Payments Standard

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the PayPal Website Payments Standard merchant gateway.

The complete set of variables specific to PayPal is available in their Integration documentation on their website in the following location:

https://www.paypal.com/IntegrationCenter/ic_standard_home.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form to work with PayPal, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read PayPal's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using PayPal, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

PayPal is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use PayPal's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to PayPal:

- **PayPal hosted form URL:**
<https://www.PayPal.com/cgi-bin/webscr>
- **cmd:** configures the transaction configuration necessary for the type of form. The Basic configuration has a default of `x_click`, but this changes depending on what information is being passed from your form, as well as if you are passing line-item details. See PayPal's documentation referenced above for more information.
- **business:** the email address unique to your Paypal account that identifies it to the gateway.
- **item_name:** this is the descriptive text associated to the transaction. The default for the Basic configuration is the static descriptive text "online order".
- **currency_code:** tells the gateway what currency the transaction should be processed in. The default value is USD.
- **amount:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.

Please refer to PayPal's documentation referenced above for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

PayPal's Instant Payment Notification (IPN) service performs a form POST containing the information pertinent to your transaction to an `http://` location. You must enable IPN and configure the passback location in your PayPal account in the following location for this information to be returned to your site:

1. Go to https://www.paypal.com/us/cgi-bin/webscr?cmd=_profile-ipn-notify and login with your merchant account information.
2. Click the *Edit* button to update your IPN configuration.
3. Check the box to enable IPN for your account.
4. Specify the `http://` location that the transaction details should be posted to upon completion of the transaction.
5. Click *Save* to update your profile

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the PayPal gateway.

In addition, PayPal can be configured to accept line item detail information and pass it back to the specified IPN page. This makes it possible to re-populate the shopping cart directly on your website using this information. See [Re-populating the shopping cart based on a form POST](#) for more information on support for this configuration.

LinkPoint Connect

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the LinkPoint merchant gateway.

eCart directly supports LinkPoint's Connect gateway for processing transactions.

The complete set of variables specific to LinkPoint Connect is available directly from LinkPoint in their documentation when you have set up a LinkPoint Connect account. This information is available when you logon to LinkPoint Central. From the top navigation, access the Support section. On the redirected page, look under Manuals > LinkPoint Connect. You'll have the option of getting the help in HTML or PDF format.

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using LinkPoint, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read LinkPoint's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using LinkPoint Connect, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

LinkPoint is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use LinkPoint's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to LinkPoint:

- **LinkPoint hosted form URL:**
`https://www.linkpointcentral.com/lpc/servlet/lppay`
- **storename:** your LinkPoint account number
- **chargetotal:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.
- **mode:** identifies the processing mode for the configuration. The Basic configuration specifies *fullpay*, which has the LinkPoint form collect all transaction information other than the basic info passed from the checkout form. The Complete and Recurring configurations use *payplus*, which allows you to pass all transaction information accepted by the gateway. Please read the documentation of all available mode types and their corresponding variables to determine what information you want to pass to LinkPoint specific to your transaction.
- **txntype:** sets the type of transaction being passed to the merchant gateway. All configurations have a default of sale, the type necessary to complete a payment transaction.
- **description:** passes descriptive text for the transaction. Pre-configured to send the *FullDetails* string available from the eCart object which provides the name, price and quantity of the items passed from the shopping cart.

Notes: You will need to reference the `http://` location of your checkout form within LinkPoint Central to verify that the submission is coming from an authorized server. Login to LinkPoint Central, click the *Customization* link, and click *Settings* in the left hand navigation pane. In the top section, specify the `http://` location of the *Order Submission Form*.

As well, you should set up your account to test mode to test processing transactions. This is configured in LinkPoint Central in the *Admin* section under *Live/Test Mode*.

Please refer to LinkPoint's Connect documentation for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

LinkPoint performs a form POST containing the information pertinent to your transaction to an `http://` location. You must enable the POST of this information and configure the passback location in your LinkPoint account in the following location for this information to be returned to your site:

1. Go to LinkPoint Central and login with your merchant account information.
2. In the top navigation, click the *Customization* link.
3. In the left navigation, click *Settings*.
4. Specify the `http://` location that the transaction details should be posted to upon completion of the transaction for both the Confirmation and the Failure pages.
5. Select *URL is a CGI script* to enable the appropriate POST to get the information sent to the specified pages.
6. In addition, if *AutoForwarding* is selected, the destination pages for the POST are redirected to, and can display the outcome of the transaction on the return to your website.

The [WA Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the LinkPoint Connect help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the LinkPoint gateway.

Payflow Link

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from PayPal's Payflow merchant gateway.

eCart directly supports the Payflow Link gateway for processing transactions.

The complete set of variables specific to Payflow Link is available directly from PayPal in their documentation when you have set up an Payflow account. This information is available in the following location:

https://www.paypal.com/IntegrationCenter/ic_Payflowlink.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form using Payflow, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read Payflow's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using Payflow, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

Payflow is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use Payflow's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to Payflow:

- **Payflow hosted form URL:**
<https://Payflowlink.paypal.com>
- **LOGIN:** your Payflow account login name that identifies you to the gateway
- **PARTNER:** tells the gateway the name of the reseller from which you purchased your account. This information should have been provided with your login information for your account. The default value is *VeriSign*.
- **AMOUNT:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.
- **TYPE:** A single letter that identifies the type of transaction. The default value for all configurations provided is *S* for Sale.
- **DESCRIPTION:** passes descriptive text for the transaction. By default, its configured to use the FullDetails calculation available in the eCart object to pass the ID, price and quantity of all items in the cart as a string.

Please refer to the Payflow Link documentation for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

Payflow performs a form POST containing the information pertinent to your transaction to an http:// location. You must enable the POST of this information and configure the passback location in your Payflow account in the following location for this information to be returned to your site:

1. To configure Payflow Link, log in to the PayPal Manager at <https://manager.paypal.com>.
2. Navigate to Service Settings > Payflow Link and click on the Configuration page. For information about configuration, click Help on that page.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the Payflow Link gateway.

WorldPay

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the WorldPay merchant gateway.

The complete set of variables specific to WorldPay is available in their integration documentation on their website in the following location:

http://support.worldpay.com/kb/integration_guides/junior/integration/help/sjig.html

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form to work with WorldPay, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read WorldPay's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using WorldPay, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

The Basic, Complete, and Shipping configurations directly support the WorldPay Junior checkout implementation, which is specific to completing the transaction using a form POST.

In addition, the Line items configuration supports WorldPay Select Checkout and passes line item information to the shopping cart, including charges applied through rules, but cannot pass discounts (See [Release Notes](#) for more information).

WorldPay is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use WorldPay's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to WorldPay:

- **WorldPay hosted form URL:**
<https://select.worldpay.com/wcc/purchase>
- **instId:** the WorldPay ID specific to your account
- **cartId:** a unique ID passed through the form to identify this purchase. The default configuration passes the SessionID of the current browser session.
- **amount:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.
- **currency:** tells the gateway what currency the transaction should be processed in. The default value is USD.
- **desc:** Passes descriptive text for the transaction. By default, its configured to use the FullDetails calculation avail abel in the eCart object to pass the ID, price and quantity of all items in the cart as a string.
- **testMode:** tells the gateway that the current transaction is a test transaction to the system so that no real payment processing takes place. All provided configurations have this parameter set to "100" to enable test transactions by default.

Please refer to WorldPay's documentation referenced above for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

WorldPay's service can be configured to perform a form POST containing the information pertinent to your transaction to an <http://> location. You must enable this feature and configure the passback location in your WorldPay account in the following location for this information to be returned to your site:

1. Go to <https://select.worldpay.com/wcc/admin> and login with your merchant account information.
2. In the lower right corner of the page, click the *Configuration options* link.
3. Check the box to enable IPN for your account.
4. Check the *Callback enabled* box to activate the return-POST feature.
5. Directly above the checkbox in the *Callback URL* field, enter the <http://> location to which the form Post should be directed to.
6. At the bottom of the page, click *SAVE CHANGES* to update your configuration.

The [WA Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the WorldPay gateway.

Note: Support for callback from WorldPay Select Checkout's service does not directly support the passback of line item details at this time, but will be added in a future release.

eWay Shared

The following is a breakdown of the information necessary to configure eCart to work with information architecture specific to submitting and retrieving information from the eWay Shared merchant gateway.

The complete set of variables specific to eWay is available in their integration documentation on their website in the following location:

<http://www.eway.com.au/support/shared.aspx>

The merchant gateway support provided within eCart is designed to facilitate the configuration of your checkout form to work with eWay, but you are responsible for the proper configuration of your merchant account to take advantage of these features.

Be sure to read eWay's documentation to ensure that your implementation works within their acceptable parameters and requirements.

Submitting checkout information

To initiate payment processing using eWay, you must submit a form to their website containing the information specific to your transaction.

The [Insert Checkout Form](#) feature of eCart directly configures and inserts a checkout form on your page for this purpose, which can then be customized to meet your needs.

eWay is flexible in the configuration of your checkout form, in that you can gather all the information on your website using a created form and pass it to their gateway for processing, or you can submit only the minimum information required to initiate the transaction and use eWay's hosted checkout form to gather any additional information on their website.

The following is the minimum information necessary to pass to eWay:

- **eWay hosted form URL:**
<https://www.eway.com.au/gateway/payment.asp>
- **ewayCustomerID:** the eWay ID specific to your account
- **ewayTotalAmount:** total amount to be billed. The provided checkout configurations retrieve this value from the GrandTotal value in your eCart implementation.
Note: All test transactions submitted to eWay need to have a Total of 10.50\$ submitted to the gateway or the gateway will reject it.

Please refer to eWay's documentation referenced above for the complete set of variables available for checkout configuration.

Retrieving transaction information from the payment gateway

eWay's service can be configured to perform a form POST containing the information pertinent to your transaction to an <http://> location. You must enable this feature in your account and configure the passback location in the eWay Business Center on their website.

The [Local Checkout Response](#) data bindings provides all of the variables passed back to the specified page through the Bindings panel. These bindings can be applied to the page for you to use and store the transaction information on your website.

The variables passed back to your site using this method are the same variables that can be passed to the website initially, as well as information specific to the processing of the transaction that are generated during processing.

Please refer to the help documentation referenced at the top of this page for a complete list of the variables available for retrieval from the WorldPay gateway.

Implementing Shipping Services

With eCart, you can set up a shipping service to retrieve a rate request and automatically add that shipping charge to your shopping cart. The shipping charge is calculated in real-time based on the criteria you supply, including the weight and size of the items shipped, as well as the destination and method of shipping chosen.

eCart supports the following shipping services:

- [Australia Post](#)
- [DHL](#)
- [FedEx](#)
- [UPS](#)
- [US Postal Service](#)

Shipping rate requests are configured as part of the Checkout Wizard. You may use the server behaviors detailed in the following section to modify the shipping rate requests created when you have used the Checkout wizard, or to create an custom shipping rate request outside of the guided process in the Checkout wizard.

Shipping services should be set up prior to the shopping cart display so that the proper shipping charges can be included in the cart display. A shipping service appears as a rule in the shopping cart object and as a charge in the shopping cart itself.

Configuring the eCart Australia Post Shipping Rate Request

The Australia Post Shipping Rate Request offered in eCart returns shipping quotes in real-time from the Australia Post eDeliver Delivery Rate Calculator. The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

Applying this server behavior updates the eCart Object automatically to apply the shipping rate retrieved to the cart total. See [Discounts, Charges, Taxes and Shipping](#) for more information.

To add this service to your shopping cart, you'll need to supply a range of both general information (such as where to redirect if the request succeeds or fails) and cart specific information (including which form elements the shipping destination details should come from). For these reasons, it's important to set up the database schema and page forms prior to incorporating the shipping service.

The Australia Post Shipping Rate Request may be added to the site through any page that includes a shopping cart; typically the service is added before the shopping cart display is inserted into the page, but after any required forms have been included in the site. A wizard type interface guides you through the process of setting up the service. The following section details the steps for configuring an eCart Shipping Rate Request from Australia Post.

To access the Australia Post Shipping Rate Request wizard:

Shipping rate requests are implemented as a server behavior. To begin the configuration process, open the Server Behaviors panel and choose Add (+). From the list, choose eCart > Shipping > Australia Post.

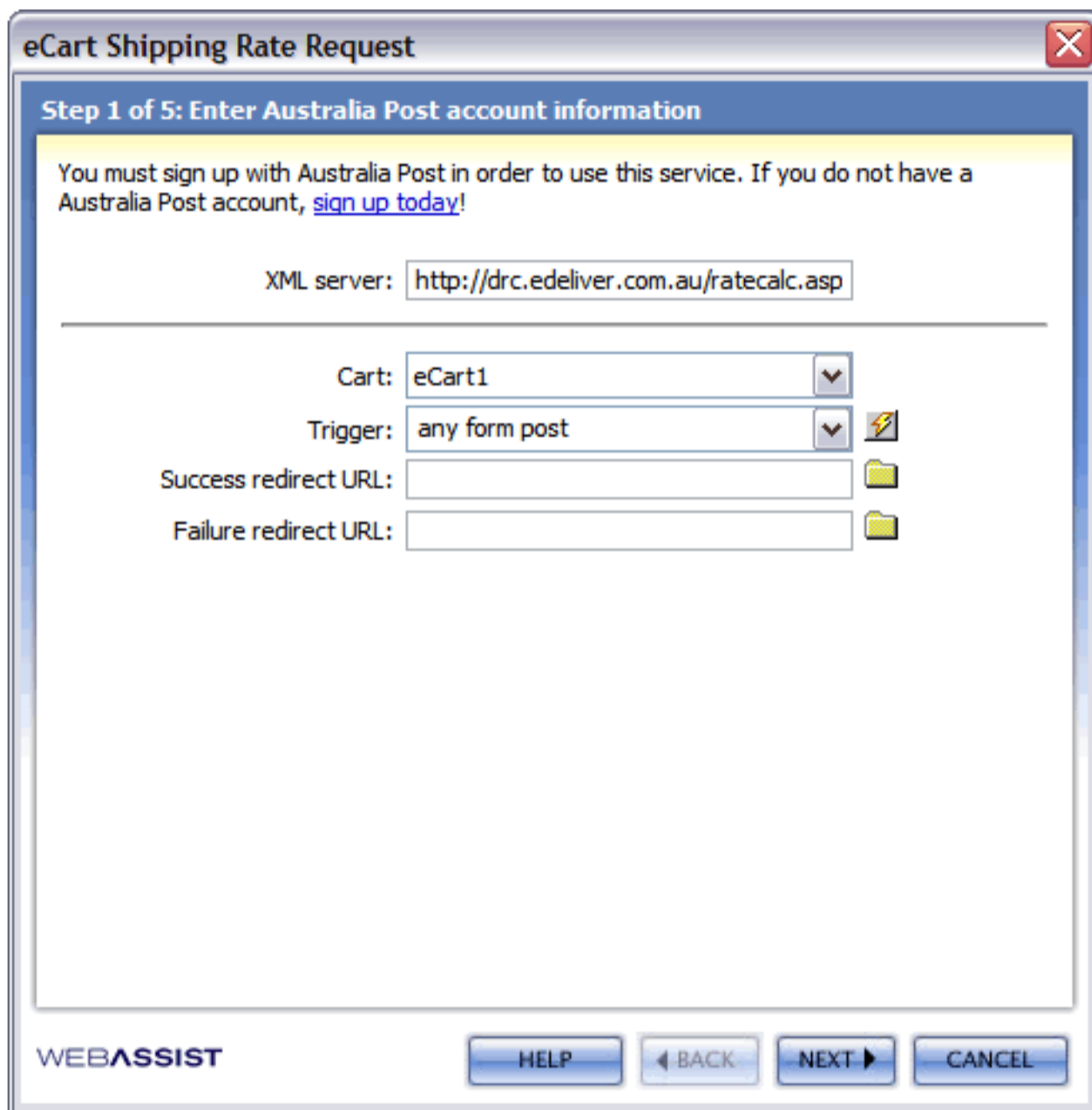
To use the Australia Post Shipping Rate Request wizard:

1. Open a page to insert the code to query the shipping service and return the shipping charge.

If you are gathering the customer shipping information from a form, apply the eCart Australia Post Shipping server behavior to the page specified as the action in that form, which is often the checkout page. If you are retrieving the shipping information from a database, place the server behavior on a page containing the recordset information prior to the checkout page.

2. From the Server Behavior panel, choose Add (+) and, from the list, select eCart > Shipping > Australia Post.

The eCart Shipping Rate Request wizard opens:



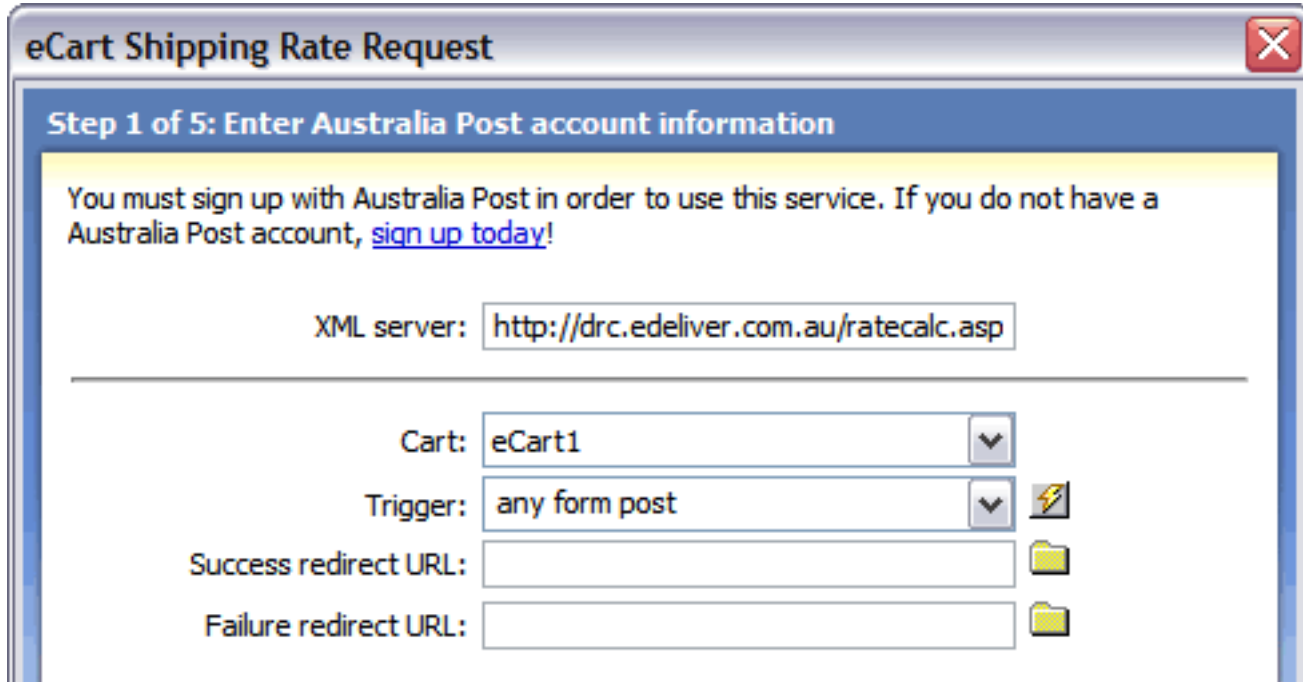
The screenshot shows a wizard window titled "eCart Shipping Rate Request" with a close button in the top right corner. The main content area has a blue header "Step 1 of 5: Enter Australia Post account information" and a yellow background with the text: "You must sign up with Australia Post in order to use this service. If you do not have a Australia Post account, [sign up today!](#)". Below this, there is a text input field for "XML server:" containing the URL "http://drc.edeliver.com.au/ratecalc.asp". A horizontal line separates this from the configuration options: "Cart:" with a dropdown menu showing "eCart1", "Trigger:" with a dropdown menu showing "any form post" and a lightning bolt icon, "Success redirect URL:" with an empty text box and a folder icon, and "Failure redirect URL:" with an empty text box and a folder icon. At the bottom left is the "WEBASSIST" logo, and at the bottom right are four buttons: "HELP", "BACK", "NEXT", and "CANCEL".

3. The wizard gathers the information necessary to communicate with the shipping service through its five steps:
 - o [Enter Australia Post account information](#)
 - o [Enter your shipment information](#)
 - o [Select your shipping origin](#)
 - o [Select your shipping destination](#)
 - o [Set your package information](#)
 - o [Choose special services per package \(optional\)](#)
4. Use the Back and Next buttons to navigate through each of these sections.
5. Upon completion of the wizard, click Finish to apply the code to the shopping cart object and the server behavior that queries the shipping provider.

The wizard creates and activates a rule in the Shipping tab of the eCart Object. You can modify the rule by selecting it and choosing Edit.

Step 1 - Enter Australia Post Account Information

In the first step of the eCart Shipping Rate Request wizard, you'll need to enter general information about how the shipping service should be initiated and redirected.



eCart Shipping Rate Request

Step 1 of 5: Enter Australia Post account information

You must sign up with Australia Post in order to use this service. If you do not have a Australia Post account, [sign up today!](#)

XML server:

Cart:

Trigger:

Success redirect URL:

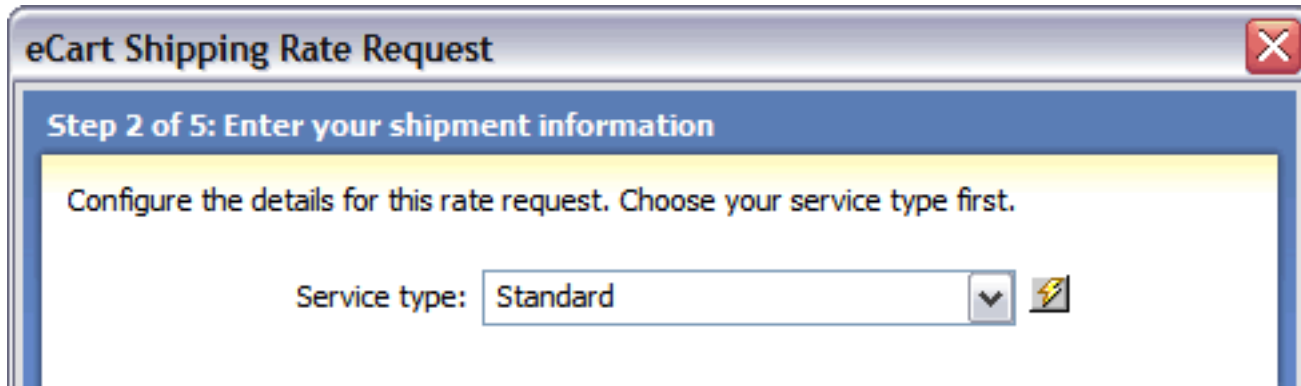
Failure redirect URL:

The following describes each of the configuration settings available and how they affect the shipping configuration. When you have made your selections, click the Next button to proceed to the following step:

- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).
- **Cart:** This is the name of the eCart Object specific to your shopping cart implementation associated with this request. This field is only updateable if you have multiple shopping cart implementations applied to the page the rate request is being applied to.
- **Trigger:** Select the trigger that will call the shipping service. The following triggers are available:
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the query to the shipping service. This is typically used when a form on a different page contains information needed by the shipping service.
 - **Before Page Load:** The query to the service is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the query to the shipping service. This is used typically when applying the server behavior to the same page that contains a form with the necessary shipping information.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
- **Success redirect URL:** The path to the page you want displayed if the shipping service returns a shipping quote successfully. Typically this field is left blank except for testing purposes.
- **Failure redirect URL:** The path to the page you want displayed if the shipping service returns an error. Typically this field is left blank except for testing purposes or to inform customers of a communications problem gathering the shipping quote.

Step 2 - Enter your shipment information

In step 2 of the wizard, you're asked to specify the types of services you'd like. Each of the choices allow you to chose either static values or dynamic values.



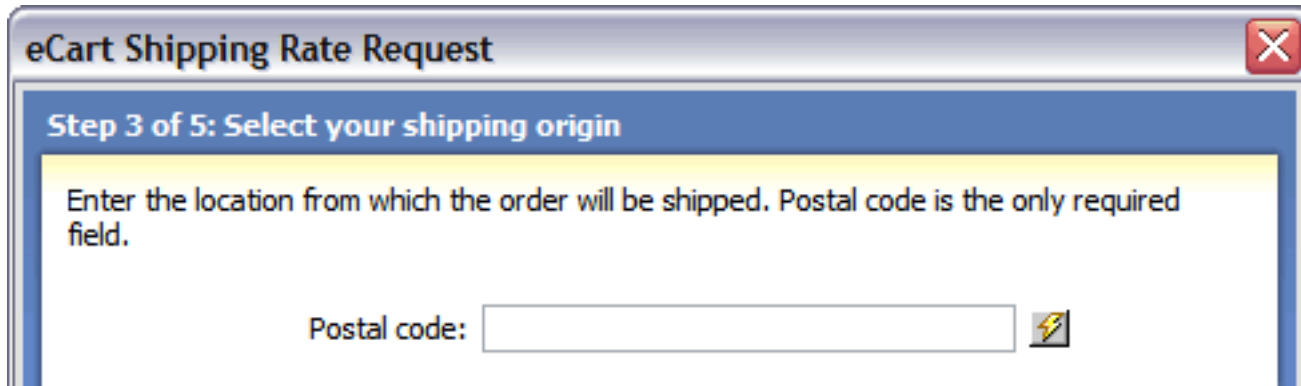
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button in the top right corner. Inside the dialog, there is a blue header bar that reads "Step 2 of 5: Enter your shipment information". Below the header, a yellow highlighted area contains the text: "Configure the details for this rate request. Choose your service type first." Below this text is a form field labeled "Service type:" with a dropdown menu showing "Standard" and a small icon of a lightning bolt to its right.

The following describes the service settings available and how they affect the shipping service options. When you have made your selections, click the Next button to proceed to the following step:

- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the service option, insert the Australia Post Service Types snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
-

Step 3 - Select your shipping origin

For Australia Post to determine the proper rate quote, you must indicate the shipping location. If the online store ships from different locations for a different item, the address values may be assigned using dynamic data.



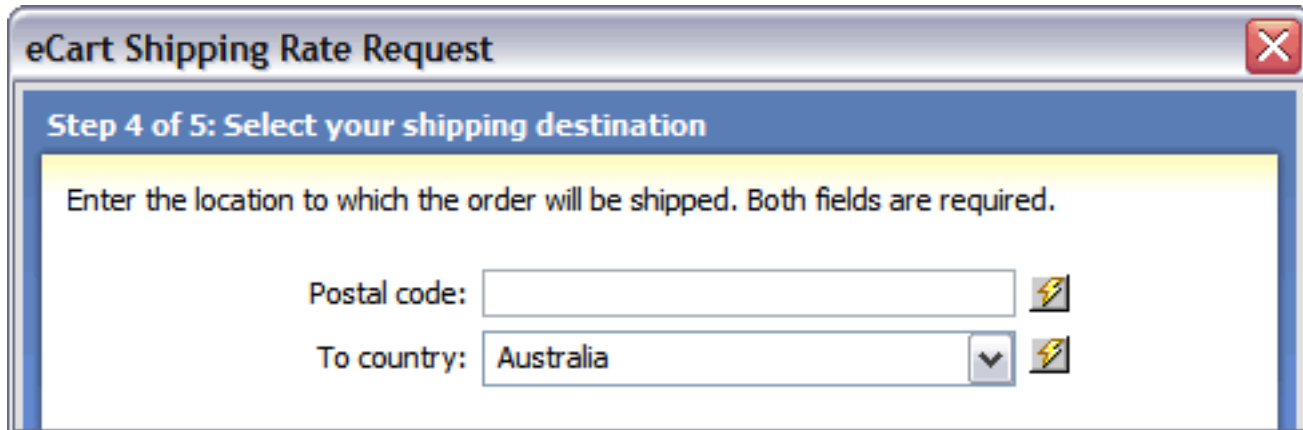
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main content area has a blue header that reads "Step 3 of 5: Select your shipping origin". Below the header, a yellow highlighted box contains the text: "Enter the location from which the order will be shipped. Postal code is the only required field." At the bottom of the dialog, there is a text input field labeled "Postal code:" followed by a lightning bolt icon, which is used to open the Dynamic Data dialog box.

The following describes the shipping origin settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
-

Step 4 - Select your shipping destination

Australia Post requires the shipping destination to determine the correct rate quote. Typically, the required values are assigned to form elements filled out at run-time by the shopper.



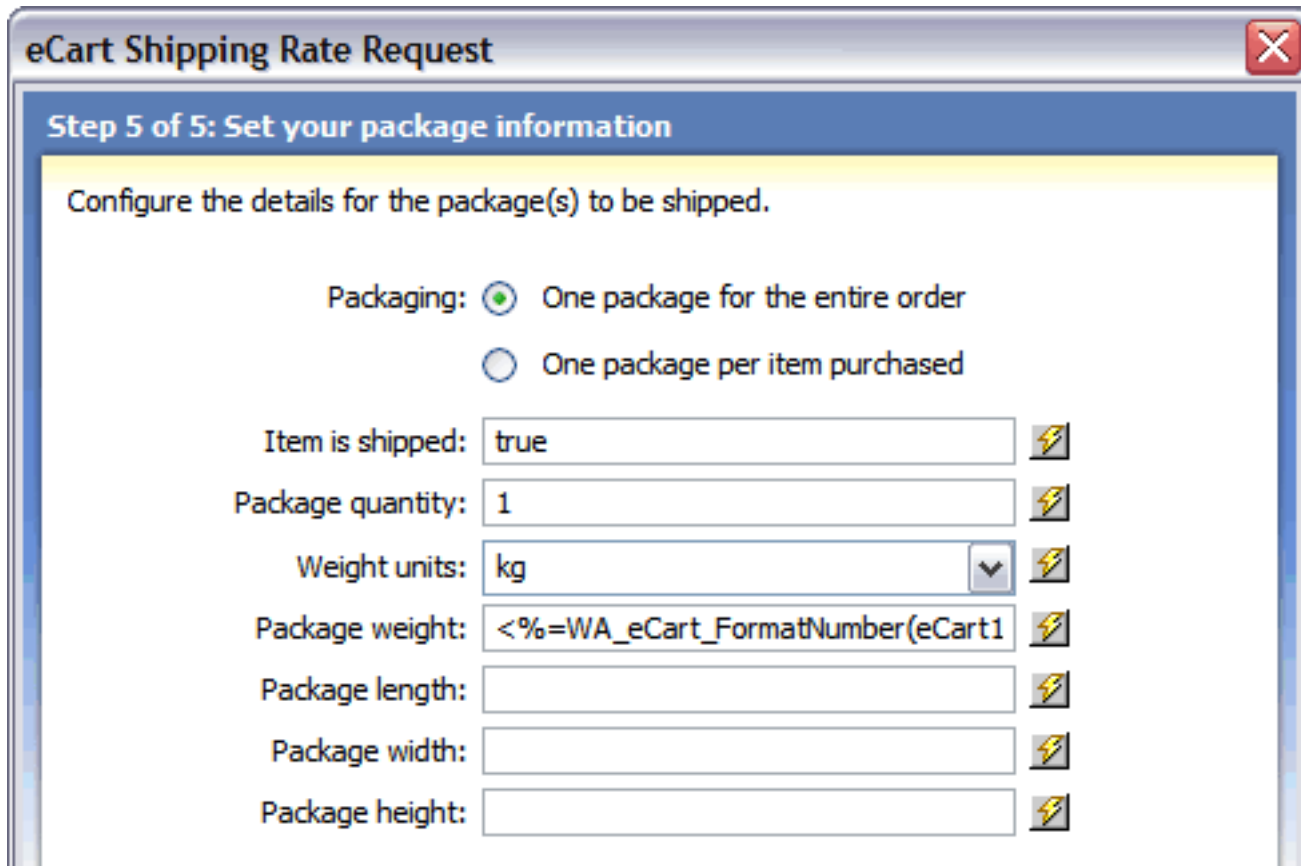
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 4 of 5: Select your shipping destination". Below this, a yellow highlighted area contains the instruction: "Enter the location to which the order will be shipped. Both fields are required." There are two input fields: "Postal code:" followed by an empty text box and a lightning bolt icon; and "To country:" followed by a dropdown menu showing "Australia" and a lightning bolt icon.

The following describes each of the shipping destination settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Postal code:** Enter the postal code that your packages are shipped to. Select the lightning bolt to open the Dynamic Data dialog box and select the form field in the checkout form used to retrieve this information from the shopper, or another data binding from which this information is readily available.
 - **To country:** Enter or choose the country where your packages are shipped to. Select the lightning bolt to open the Dynamic Data dialog box and select the form field in the checkout form used to retrieve this information from the shopper, or another data binding from which this information is readily available. The provided Shipping Countries snippet can also be used to insert a list in the checkout form that can be bound in the dynamic data dialog, allowing customers to select the country.
-

Step 5 - Set your package information

In step 5 of the wizard, you're asked to specify the details of the packages to be shipped. You can choose between setting up one package for the entire order or a separate package for every item. Every field on this step of the wizard is required.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button in the top right corner. The main heading is "Step 5 of 5: Set your package information". Below this, a yellow banner contains the text "Configure the details for the package(s) to be shipped." The form includes two radio buttons for "Packaging": "One package for the entire order" (selected) and "One package per item purchased". Below are seven input fields, each with a lightning bolt icon to its right: "Item is shipped:" with the value "true"; "Package quantity:" with the value "1"; "Weight units:" with a dropdown menu showing "kg"; "Package weight:" with the value "<%=WA_eCart_FormatNumber(eCart1"; "Package length:"; "Package width:"; and "Package height:".

The following describes each of the package settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
 - **Item is shipped:** Determines whether the item in the cart is to be shipped via Australia Post or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
 - **Package quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
 - **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment.
 - **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
 - **Package length:** Sets the length of the package to be shipped.
 - **Package width:** Sets the width of the package to be shipped.
 - **Package height:** Sets the height of the package to be shipped.
-

Configuring the eCart DHL Shipping Rate Request

The DHL Shipping Rate Request offered in eCart returns shipping quotes in real-time from DHL's XML developer tools for ShipIT and International. The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

Applying this server behavior updates the eCart Object automatically to apply the shipping rate retrieved to the cart total. See [Discounts, Charges, Taxes and Shipping](#) for more information.

To add this service to your shopping cart, you'll need to supply a range of both general information (such as your user id and account number) and cart specific information (including which form elements the shipping destination details should come from). For these reasons, it's important to set up the database schema and page forms prior to incorporating the shipping service.

The DHL Shipping Rate Request may be added to the site through any page that includes a shopping cart; typically the service is added before the shopping cart display is inserted into the page, but after any required forms have been included in the site. A wizard type interface guides you through the process of setting up the service. The following section details the steps for configuring an eCart Shipping Rate Request from DHL.

To access the DHL Shipping Rate Request wizard:

Shipping rate requests are implemented as a server behavior. To begin the configuration process, open the Server Behaviors panel and choose Add (+). From the list, choose eCart > Shipping > DHL.

To use the DHL Shipping Rate Request wizard:

1. Open a page to insert the code to query the shipping service and return the shipping charge.

If you are gathering the customer shipping information from a form, apply the eCart DHL Shipping server behavior to the page specified as the action in that form, which is often the checkout page. If you are retrieving the shipping information from a database, place the server behavior on a page containing the recordset information prior to the checkout page.

2. From the Server Behavior panel, choose Add (+) and, from the list, select eCart > Shipping > DHL.

The eCart Shipping Rate Request wizard opens:

The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main content area has a blue header "Step 1 of 5: Enter DHL account information". Below the header is a yellow warning box with the text: "You must sign up with DHL in order to use this service. If you do not have a DHL account, [sign up today!](#)".

Below the warning box are several input fields:

- User ID:
- Password:
- Domestic shipping key:
- International shipping key:
- Account number:
- XML server:

Below these fields is a horizontal line. Underneath the line are more fields:

- Cart: (with a dropdown arrow)
- Trigger: (with a dropdown arrow and a lightning bolt icon)
- Success redirect URL: (with a folder icon)
- Failure redirect URL: (with a folder icon)

At the bottom left of the dialog is the "WEBASSIST" logo. At the bottom right are four buttons: "HELP", "BACK", "NEXT", and "CANCEL".

3. The wizard gathers the information necessary to communicate with the shipping service through its five steps:
 - o [Enter DHL account information](#)
 - o [Enter your shipment information](#)
 - o [Select your shipping origin](#)
 - o [Select your shipping destination](#)
 - o [Set your package information](#)
4. Use the Back and Next buttons to navigate through each of these sections.
5. Upon completion of the wizard, click Finish to apply the code to the shopping cart object and the server behavior that queries the shipping provider.

The wizard creates and activates a rule in the Shipping tab of the eCart Object. You can modify the rule by selecting it and choosing Edit.

Step 1 - Enter DHL Account Information

In the first step of the eCart Shipping Rate Request wizard, you'll need to enter general information about how the shipping rate request service should be initiated and redirected.

eCart Shipping Rate Request

Step 1 of 5: Enter DHL account information

You must sign up with DHL in order to use this service. If you do not have a DHL account, [sign up today!](#)

User ID:

Password:

Domestic shipping key:

International shipping key:

Account number:

XML server:

Cart:

Trigger:

Success redirect URL:

Failure redirect URL:

The following describes each of the configuration settings available and how they affect the shipping configuration. When you have made your selections, click the Next button to proceed to the following step:

- **User ID:** Enter the User ID associated with your DHL shipping account.
- **Password:** Enter the password associated to your account's User ID.
- **Domestic shipping key:** Enter the domestic shipping key for your XML ShipIT account. If you do not have this information, contact a DHL account representative at XMLRequests@dhl.com
- **International shipping key:** Enter the domestic shipping key for your XML International account. If you do not have this information, contact a DHL account representative at XMLRequests@dhl.com
- **Account number:** Enter your DHL account number.
- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).
- **Cart:** This is the name of the eCart Object specific to your shopping cart implementation associated with this request. This field is only updateable if you have multiple shopping cart implementations applied to the page the rate request is being applied to.
- **Trigger:** Select the trigger that will call the shipping service. The following triggers are available:
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the query to the shipping service. This is typically used when a form on a different page contains information needed by the shipping service.
 - **Before Page Load:** The query to the service is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the query to the shipping service. This is used typically when applying the server behavior to the same page that contains a form with the necessary shipping information.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
- **Success redirect URL:** The path to the page you want displayed if the shipping service returns a shipping quote successfully. Typically this field is left blank except for testing purposes.
- **Failure redirect URL:** The path to the page you want displayed if the shipping service returns an error. Typically this field is left blank except for testing purposes or to inform customers of a communications problem gathering the shipping quote.

Step 2 - Enter your shipment information

In step 2 of the wizard, you're asked to specify the types of services you'd like. Each of the choices allow you to chose either static values or dynamic values.

The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. Below the title bar is a blue header with the text "Step 2 of 5: Enter your shipment information". A yellow background area contains the instruction: "Configure the details for this rate request. Choose your service type first, then enter some information on how this shipment will be billed. Some options will effect the shipping cost." Below this are several input fields, each with a lightning bolt icon to its right, indicating dynamic data binding:

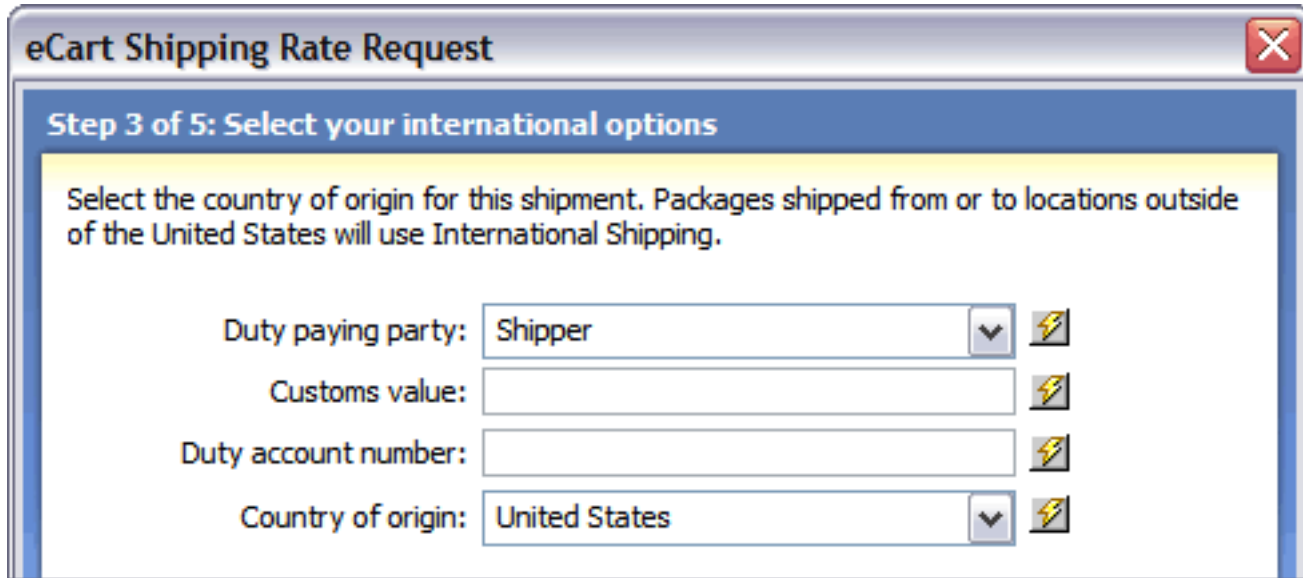
- Service type: Express (dropdown menu)
- Billing party: Shipper (dropdown menu)
- Billing account: (text input)
- COD value: (text input)
- COD payment: Not Applicable (dropdown menu)
- Trace: (text input)
- Shipping date: (text input)

The following describes the service settings available and how they affect the shipping service options. When you have made your selections, click the Next button to proceed to the following step:

- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the service option, insert the DHL Service Types snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
 - **Billing party:** Selects the appropriate party to be billed. This can be set statically or retrieved from a dynamic data binding.
 - **Billing account:** Selects the DHL account to be billed. This can be set statically or retrieved from a dynamic data binding.
 - **COD value:** Sets the value associated to Cash on Delivery payments. This field cannot be 0 if a COD payment type is selected, or the COD payment setting is ignored. This can be set statically or retrieved from a dynamic data binding.
 - **COD payment:** Sets the options for available Cash on Delivery payments. This can be set statically or retrieved from a dynamic data binding.
 - **Trace:** Sets tracking information specific to your rate request. This is not processed by DHL when the rate request is submitted, but is returned in the rate request response for your tracking purposes. This can be retrieved from a dynamic data binding.
 - **Shipping date:** Sets the date the package is to be picked up by DHL. This can be set statically or retrieved from a dynamic data binding.
-

Step 3 - Select your international options

For DHL to determine the proper rate quote based on customs and duty restrictions, you must indicate the country of origin. If the online store ships from different locations for a different item, the values may be assigned using dynamic data.



eCart Shipping Rate Request [Close]

Step 3 of 5: Select your international options

Select the country of origin for this shipment. Packages shipped from or to locations outside of the United States will use International Shipping.

Duty paying party: [v] [⚡]

Customs value: [⚡]

Duty account number: [⚡]

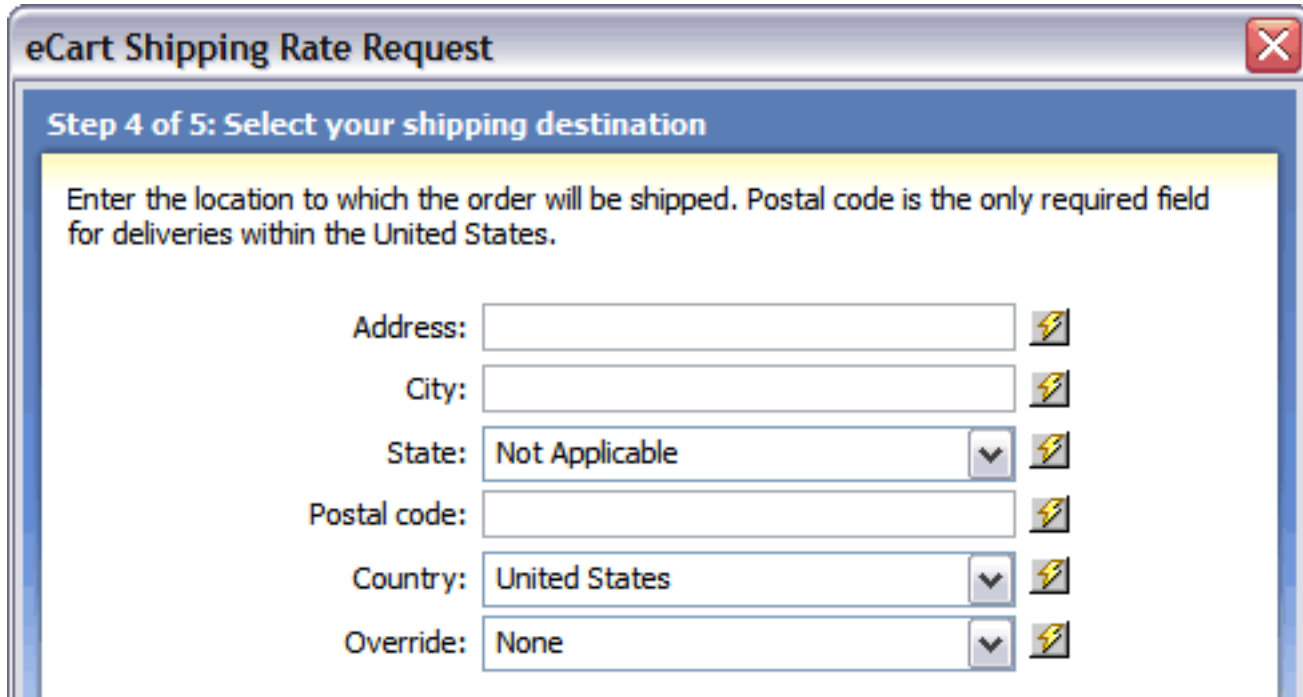
Country of origin: [v] [⚡]

The following describes the international shipping settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Duty paying party:** Sets the party responsible for paying any duty associated with the international order, affecting the rate total. This can be set statically or retrieved from a dynamic data binding.
 - **Customs value:** Sets the value of the shipment to determine the customs charges. This can be set statically or retrieved from a dynamic data binding, typically the subtotal of the items included in the shipment.
 - **Duty account number:** Sets the account number for duty collection. This can be set statically or retrieved from a dynamic data binding.
 - **Country of origin:** Sets the country the shipment originates from, determining if the order is international or not. This can be set statically or retrieved from a dynamic data binding.
-

Step 4 - Select your shipping destination

DHL requires the shipping destination to determine the correct rate quote. Typically, the required values are assigned to form elements filled out at run-time by the shopper.



eCart Shipping Rate Request [Close]

Step 4 of 5: Select your shipping destination

Enter the location to which the order will be shipped. Postal code is the only required field for deliveries within the United States.

Address: ⚡

City: ⚡

State: ⚡

Postal code: ⚡

Country: ⚡

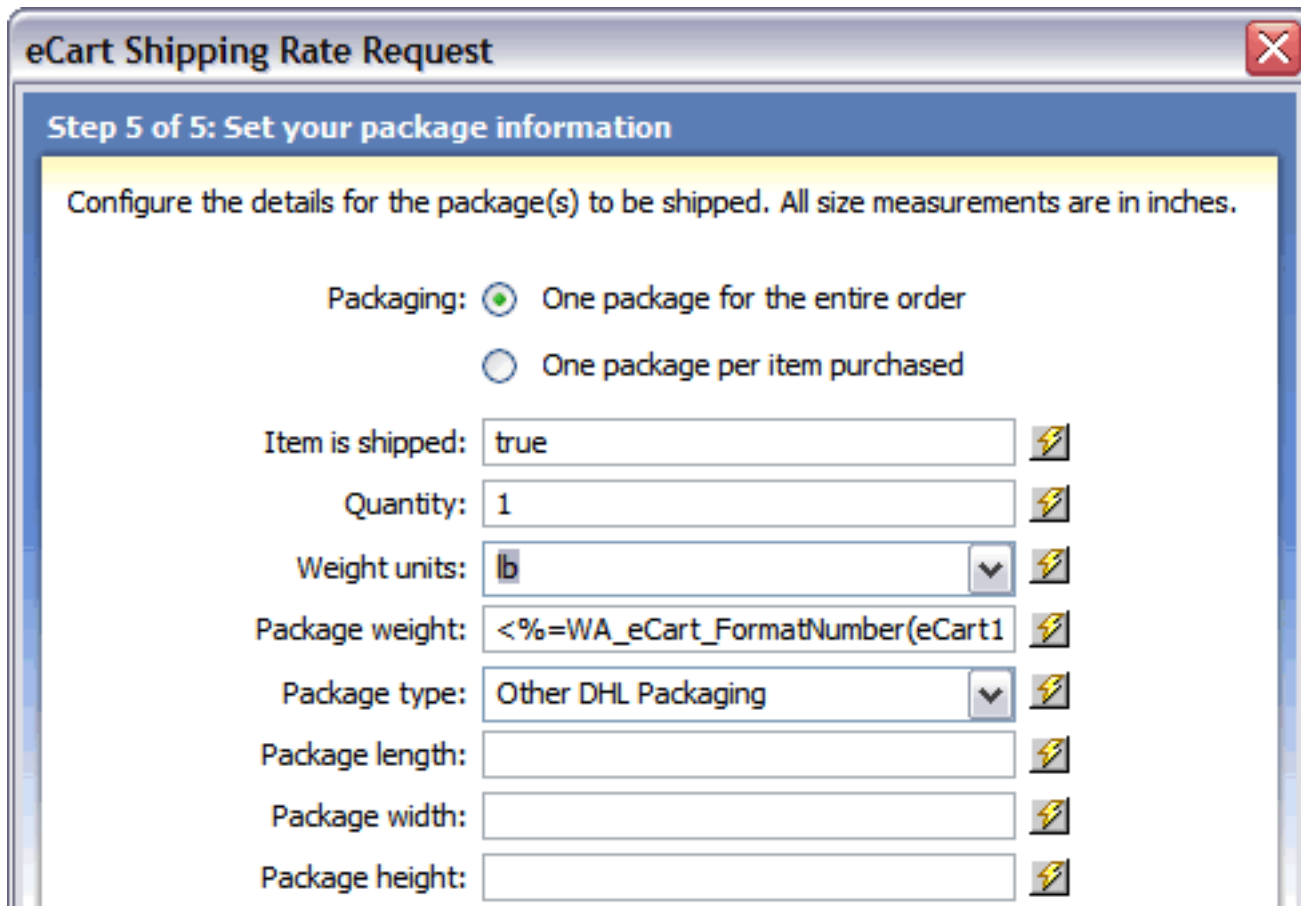
Override: ⚡

The following describes each of the shipping destination settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Address:** Enter the street address the packages are to be shipped to. This can be retrieved from a dynamic data binding, typically a checkout form field.
 - **City:** Enter the city the packages are to be shipped to. This can be retrieved from a dynamic data binding, typically a checkout form field.
 - **State:** Enter the state the packages are to be shipped to. This can be retrieved from a dynamic data binding, typically a checkout form field. A form field listing all the states is available in the Snippets panel under *WA eCart > Shipping*. Insert the Shipping States snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
 - **Postal code:** Enter the postal code that your packages are shipped to. Select the lightning bolt to open the Dynamic Data dialog box and select the form field in the checkout form used to retrieve this information from the shopper, or another data binding from which this information is readily available.
 - **Country:** Enter or choose the country where your packages are shipped to. Select the lightning bolt to open the Dynamic Data dialog box and select the form field in the checkout form used to retrieve this information from the shopper, or another data binding from which this information is readily available. A form field listing all the countries is available in the Snippets panel under *WA eCart > Shipping*. Insert the Shipping Countries snippet in the shipping form prior to accessing the server behavior, and select that form list's name from the Dynamic Data dialog.
 - **Override:** Selects the appropriate override scenario if the postal code entered is not available in the system, or if a delivery option is selected that doesn't meet the rate request requirements. This allows a rate to still be generated, which may be incorrect, but does cause the request to fail. This can be set statically or retrieved from a dynamic data binding.
-

Step 5 - Set your package information

In step 5 of the wizard, you're asked to specify the details of the packages to be shipped. You can choose between setting up one package for the entire order or a separate package for every item. Every field on this step of the wizard is required.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading is "Step 5 of 5: Set your package information". Below this, a yellow banner contains the text: "Configure the details for the package(s) to be shipped. All size measurements are in inches." The form contains the following fields:

- Packaging:** Two radio buttons. The first is selected and labeled "One package for the entire order". The second is labeled "One package per item purchased".
- Item is shipped:** A text input field containing "true" and a lightning bolt icon.
- Quantity:** A text input field containing "1" and a lightning bolt icon.
- Weight units:** A dropdown menu showing "lb" and a lightning bolt icon.
- Package weight:** A text input field containing the code "<%=WA_eCart_FormatNumber(eCart1" and a lightning bolt icon.
- Package type:** A dropdown menu showing "Other DHL Packaging" and a lightning bolt icon.
- Package length:** An empty text input field and a lightning bolt icon.
- Package width:** An empty text input field and a lightning bolt icon.
- Package height:** An empty text input field and a lightning bolt icon.

The following describes each of the package settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is shipped:** Determines whether the item in the cart is to be shipped via Australia Post or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment.
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type :** Determines the type of packaging used to make the shipment.
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Configuring the eCart FedEx Shipping Rate Request

The FedEx shipping service offered in eCart returns shipping quotes in real-time from the FedEx Web service. The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

Applying this server behavior updates the eCart Object automatically to apply the shipping rate retrieved to the cart total. See [Discounts, Charges, Taxes and Shipping](#) for more information.

To add this service to your shopping cart, you'll need to supply a range of both general information (such as your FedEx customer number and meter number) and cart specific information (including which form elements the shipping destination details should come from). For these reasons, it's important to set up your FedEx account as well as the database schema and page forms prior to incorporating the shipping service.

The FedEx shipping service may be added to the site through any page that includes a shopping cart; typically the service is added before the shopping cart display is inserted into the page, but after any required forms have been included in the site. A wizard type interface guides you through the process of setting up the service. The following section details the steps for configuring the eCart FedEx Shipping feature.

To access the eCart FedEx Shipping Rate Request wizard:

The eCart FedEx Shipping feature is implemented as a server behavior. To begin the configuration process, open the Server Behaviors panel and choose Add (+). From the list, choose eCart > Shipping > FedEx.

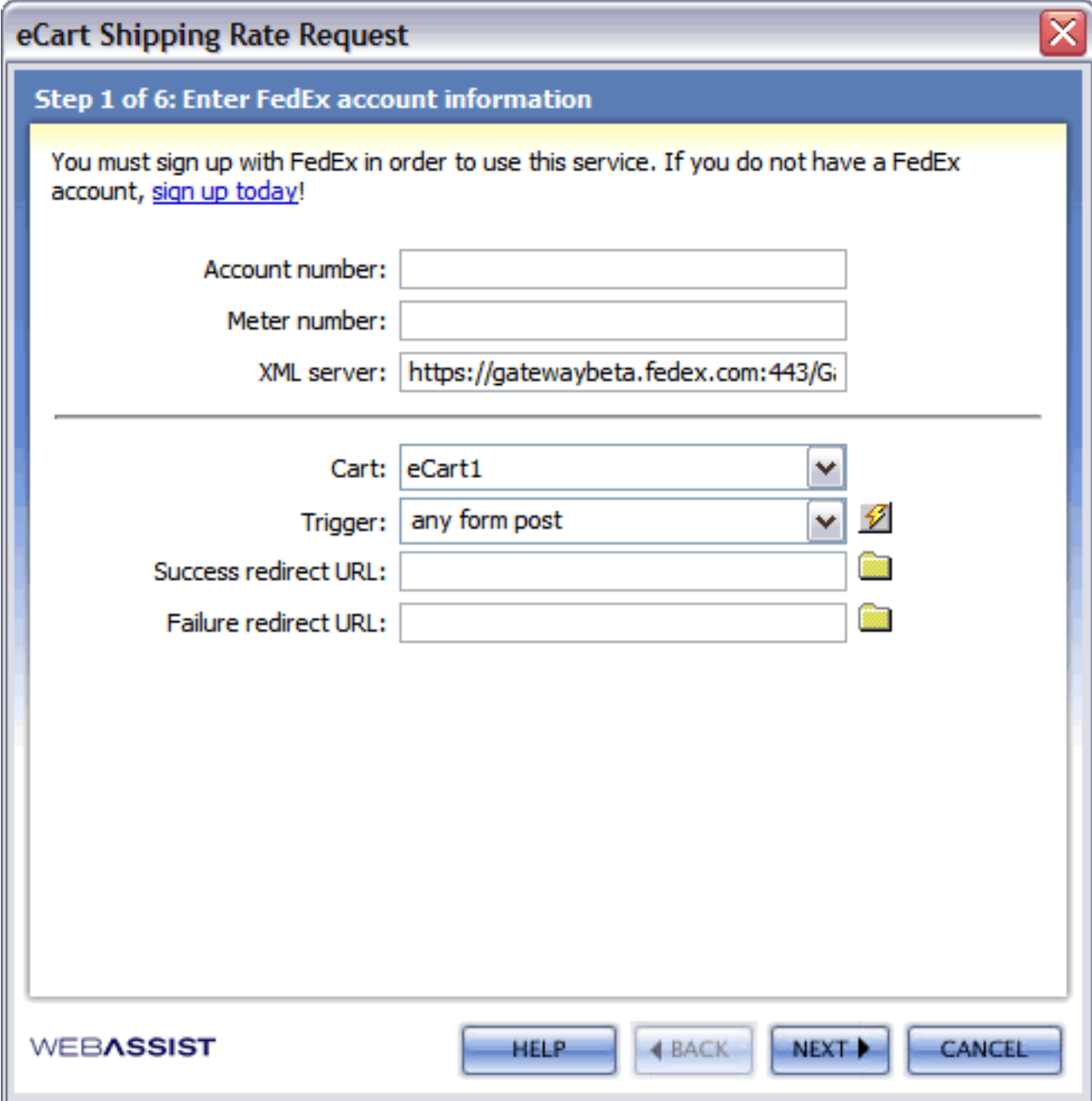
To use the eCart FedEx Shipping Rate Request wizard:

1. Open a page to insert the code to query the shipping service and return the shipping charge.

If you are gathering the customer shipping information from a form, apply the eCart FedEx Shipping server behavior to the page specified as the action in that form, which is often the checkout page. If you are retrieving the shipping information from a database, place the server behavior on a page containing the recordset information prior to the checkout page.

2. From the Server Behavior panel, choose Add (+) and, from the list, select eCart > Shipping > FedEx.

The eCart Shipping Rate Request wizard opens:



The screenshot shows a wizard window titled "eCart Shipping Rate Request" with a close button in the top right corner. The window has a blue header bar that says "Step 1 of 6: Enter FedEx account information". Below the header, there is a yellow background with the text: "You must sign up with FedEx in order to use this service. If you do not have a FedEx account, [sign up today!](#)".

The form contains the following fields:

- Account number:
- Meter number:
- XML server:

Below a horizontal line, there are four more fields:

- Cart: (dropdown menu)
- Trigger: (dropdown menu with a lightning bolt icon)
- Success redirect URL: (with a folder icon)
- Failure redirect URL: (with a folder icon)

At the bottom left of the window is the "WEBASSIST" logo. At the bottom right are four buttons: "HELP", "BACK", "NEXT", and "CANCEL".

3. The wizard gathers the information necessary to communicate with the shipping service through its six steps:

- o [Enter FedEx account information](#)
- o [Enter your shipment information](#)
- o [Select your shipping origin](#)
- o [Select your shipping destination](#)
- o [Set your package information](#)
- o [Choose special services per package \(optional\)](#)

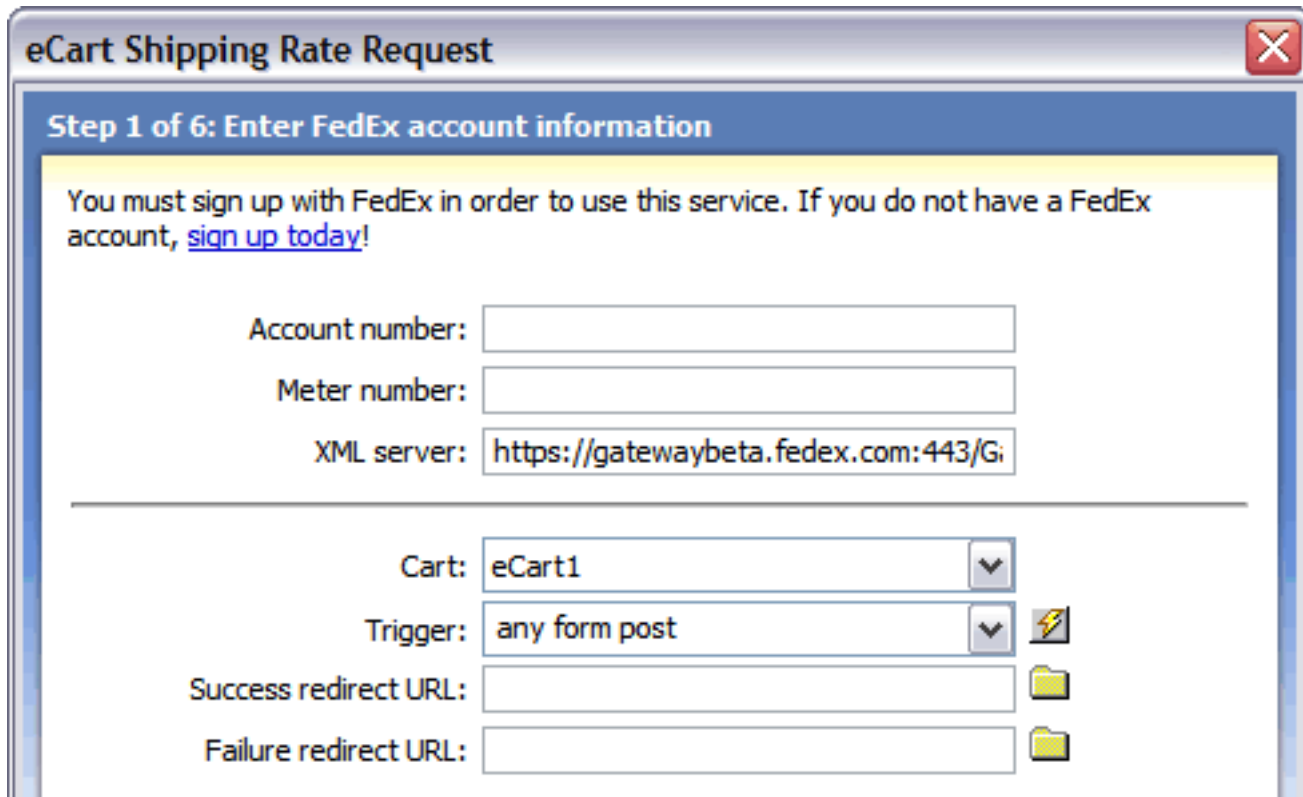
4. Use the Back and Next buttons to navigate through each of these sections.

5. Upon completion of the eCart FedEx Shipping, click Finish to apply the code to the shopping cart object and the server behavior that queries the shipping provider.

The wizard creates and activates a rule in the Shipping tab of the eCart Object. You can modify the rule by selecting it and choosing Edit.

Step 1 - Enter FedEx Account Information

In the first step of the eCart FedEx Shipping Rate Request wizard, you'll need to enter general information about your account as well as about how the shipping service should be initiated and redirected.



eCart Shipping Rate Request

Step 1 of 6: Enter FedEx account information

You must sign up with FedEx in order to use this service. If you do not have a FedEx account, [sign up today!](#)

Account number:

Meter number:

XML server:

Cart:

Trigger:

Success redirect URL:

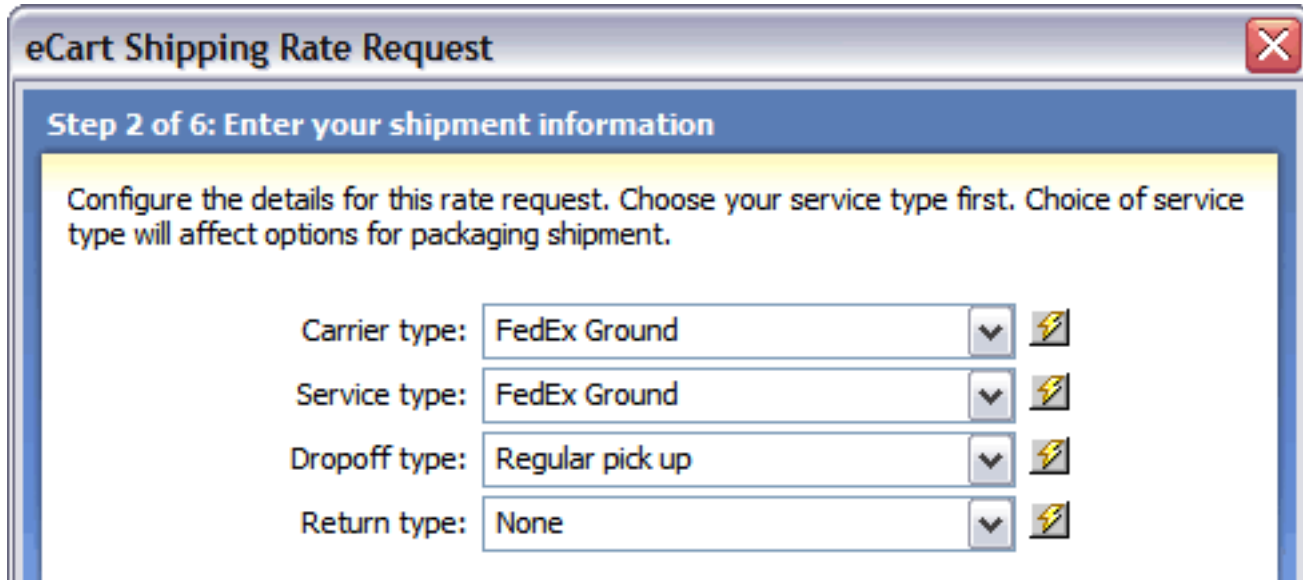
Failure redirect URL:

The following describes each of the configuration settings available and how they affect the shipping configuration. When you have made your selections, click the Next button to proceed to the following step:

- **Account number:** Enter the FedEx account number assigned to your account.
- **Meter number:** Enter the meter number you want to use for your account.
- **XML server:** Enter the full path to the secure XML server for your account (should be available by default).
- **Cart:** This is the name of the eCart Object specific to your shopping cart implementation associated with this request. This field is only updateable if you have multiple shopping cart implementations applied to the page the rate request is being applied to.
- **Trigger:** Select the trigger that will call the shipping service. The following triggers are available:
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the query to the shipping service. This is typically used when a form on a different page contains information needed by the shipping service.
 - **Before Page Load:** The query to the service is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the query to the shipping service. This is used typically when applying the server behavior to the same page that contains a form with the necessary shipping information.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
- **Success redirect URL:** The path to the page you want displayed if the shipping service returns a shipping quote successfully. Typically this field is left blank except for testing purposes.
- **Failure redirect URL:** The path to the page you want displayed if the shipping service returns an error. Typically this field is left blank except for testing purposes or to inform customers of a communications problem gathering the shipping quote.

Step 2 - Enter your shipment information

In step 2 of the wizard, you're asked to specify the types of services you'd like. Each of the choices allow you to chose either static values or dynamic values. It's important that you make your Carrier type selection first as that changes the Service type options available.



eCart Shipping Rate Request [X]

Step 2 of 6: Enter your shipment information

Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment.

Carrier type:	FedEx Ground	▼	⚡
Service type:	FedEx Ground	▼	⚡
Dropoff type:	Regular pick up	▼	⚡
Return type:	None	▼	⚡

The following describes each of the configuration settings available and how they affect the shipping service options. When you have made your selections, click the Next button to proceed to the following step:

- **Carrier type:** Chooses the primary carrier method, either FedEx Ground or FedEx Express. To allow the customer to choose the option, insert the FedEx Carrier Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
 - **Service type:** Selects the specific kind of service desired. To allow the customer to choose the option, insert the FedEx Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
 - **Dropoff type:** Determines the type of drop-off desired. Options are Regular pick up, Request courier, Drop box, Business service center, and Station; typically these are set to one type by the shipper.
 - **Return type:** Sets the type of return allowed. Options are Non-return, Non-return or online label, Return manager, Print return label, Email label, and Express/Call Tag. Again, these are typically set to one value by the shipper.
-

Step 3 - Select your shipping origin

For FedEx to determine the proper rate quote, you must indicate the shipping location. If the online store ships from different locations for different item, the address values may be assigned to dynamic data.



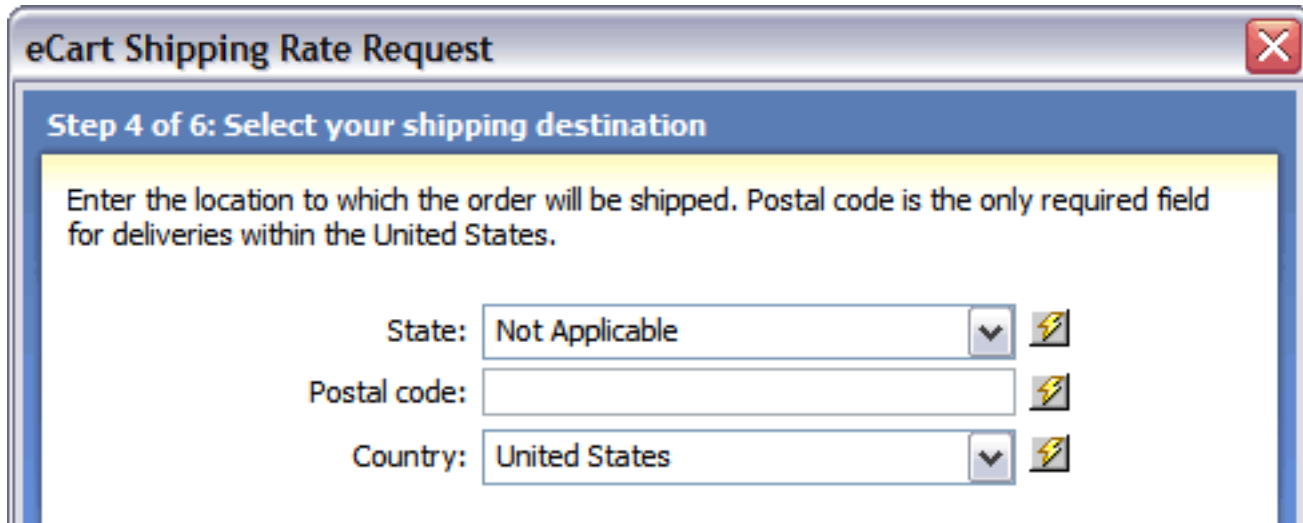
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 3 of 6: Select your shipping origin". Below the heading, there is a yellow highlighted area with the text: "Enter the location from which the order will be shipped. Postal code is the only required field for shipments within the United States." Below this text are three input fields: "State:" with a dropdown menu showing "Not Applicable" and a lightning bolt icon; "Postal code:" with a text input field and a lightning bolt icon; and "Country:" with a dropdown menu showing "United States" and a lightning bolt icon.

The following describes each of the shipping origin settings available. When you have made your selections, click the Next button to proceed to the following step:

- **State:** Enter or choose the state or province where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
 - **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
 - **Country:** Enter or choose the country where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
-

Step 4 - Select your shipping destination

FedEx requires the shipping destination to determine the correct rate quote. Typically, the required values are assigned to form elements filled out at run-time by the shopper.



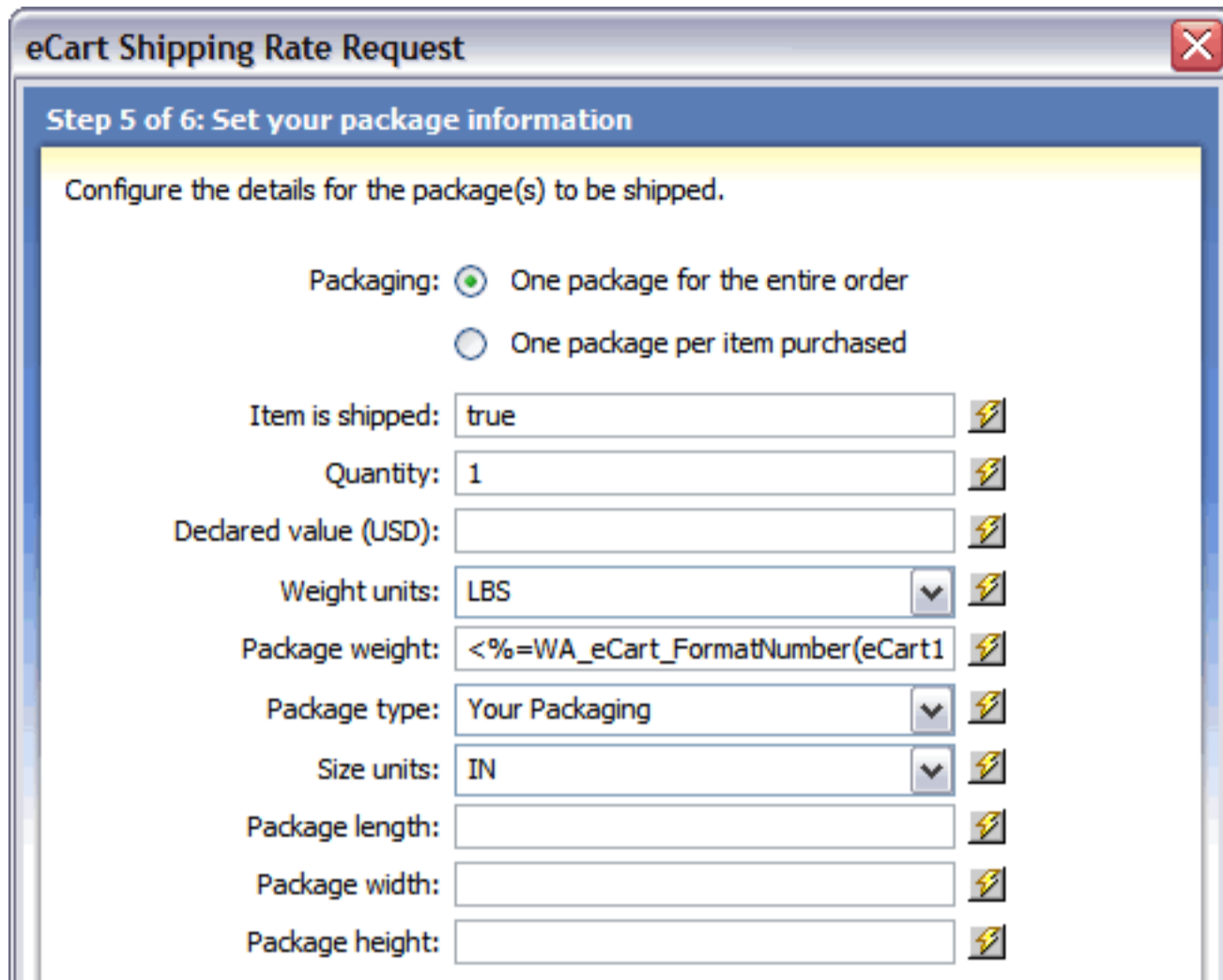
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 4 of 6: Select your shipping destination". Below the heading, there is a yellow highlighted area with the text: "Enter the location to which the order will be shipped. Postal code is the only required field for deliveries within the United States." Below this text are three form elements: "State:" with a dropdown menu showing "Not Applicable" and a lightning bolt icon; "Postal code:" with a text input field and a lightning bolt icon; and "Country:" with a dropdown menu showing "United States" and a lightning bolt icon.

The following describes each of the shipping destination settings available. When you have made your selections, click the Next button to proceed to the following step:

- **State:** Enter or choose the state or province where your packages are shipped to. Insert the provided FedEx Shipping Destination State/Province snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
 - **Postal Code:** Enter the postal code that your packages are shipped to. Insert the provided FedEx Shipping Destination Postal Code snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
 - **Country:** Enter or choose the country where your packages are shipped to. Insert the provided FedEx Shipping Destination Country snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
-

Step 5 - Set your package information

In step 5 of the wizard, you're asked to specify the details of the packages to be shipped. You can choose between setting up one package for the entire order or a separate package for every item. Every field on this step of the wizard is required.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. Below the title bar, there is a blue header bar that says "Step 5 of 6: Set your package information". Underneath, a yellow banner reads "Configure the details for the package(s) to be shipped." The main area contains the following fields:

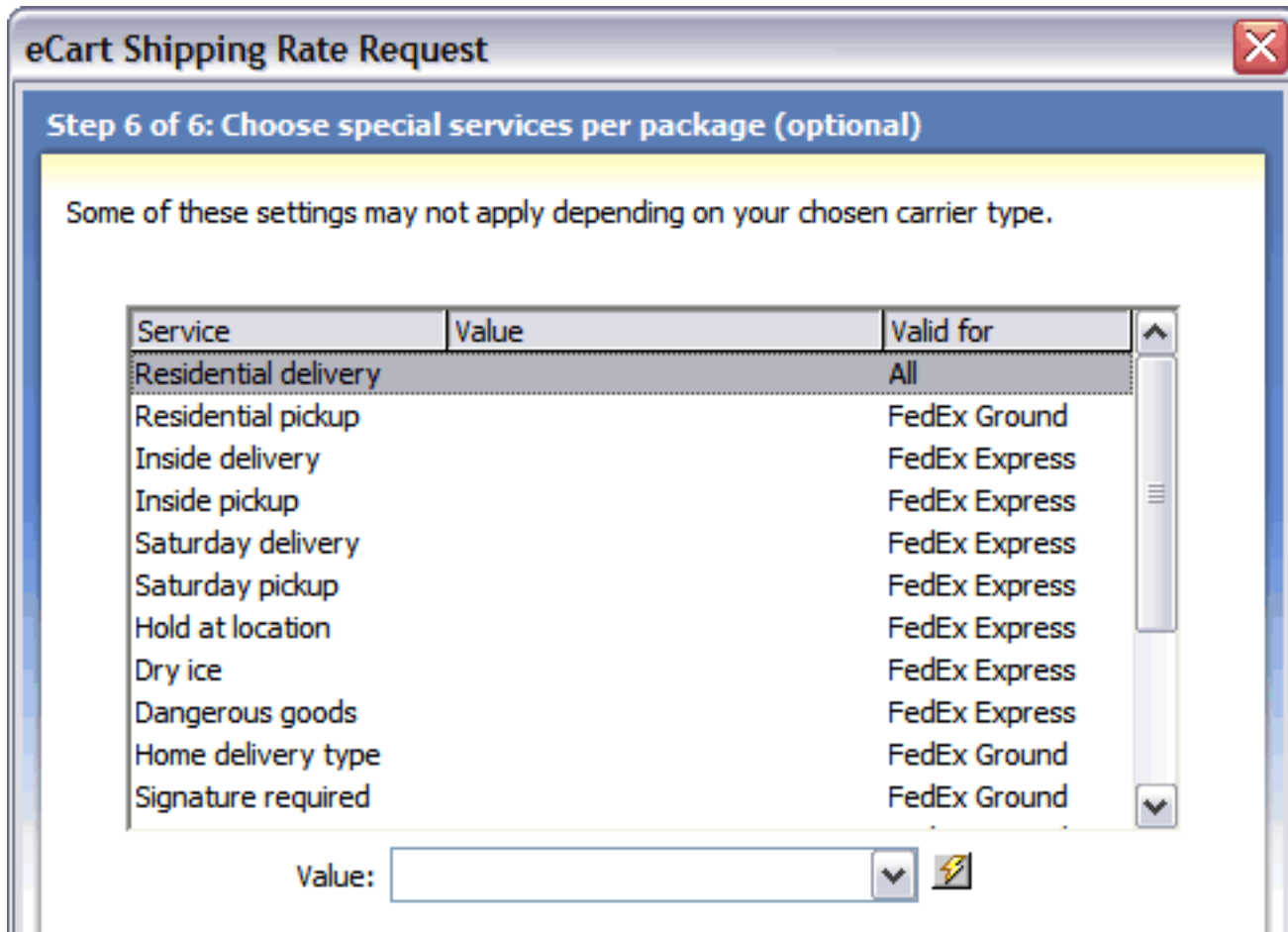
- Packaging:** Two radio buttons. The first is selected and labeled "One package for the entire order". The second is labeled "One package per item purchased".
- Item is shipped:** A text input field containing "true" and a lightning bolt icon.
- Quantity:** A text input field containing "1" and a lightning bolt icon.
- Declared value (USD):** An empty text input field and a lightning bolt icon.
- Weight units:** A dropdown menu showing "LBS" and a lightning bolt icon.
- Package weight:** A text input field containing "<%=WA_eCart_FormatNumber(eCart1" and a lightning bolt icon.
- Package type:** A dropdown menu showing "Your Packaging" and a lightning bolt icon.
- Size units:** A dropdown menu showing "IN" and a lightning bolt icon.
- Package length:** An empty text input field and a lightning bolt icon.
- Package width:** An empty text input field and a lightning bolt icon.
- Package height:** An empty text input field and a lightning bolt icon.

The following describes each of the package settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is shipped:** Determines whether the item in the cart is to be shipped via FedEx or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Quantity:** Sets the number of packages used for the shipment. If items are sent individually, this can be set as 1. Otherwise, you may wish to use dynamic data to determine how the shipment is to be broken up if there is an option for multiple packages to be sent.
- **Declared value (USD):** Sets the declared value of each object in the cart. If the items have separate values and they are shipped separately, use the Dynamic Data dialog box to assign this field to the proper column; to combine values, use the Dynamic Data dialog box to select a calculated shopping cart column like TotalPrice.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used. Choose the Your Packaging option if you use non-FedEx packaging; to use FedEx packaging choose one of the following: FedEx Envelope, FedEx Pak, FedEx Express Box, FedEx Tube, FedEx 10Kg Box, or FedEx 25Kg Box. If you previously selected FedEx Ground as your service, only the Your Packaging option is available.
- **Size units:** Declares the type of size units to be used in calculating the total size of the shipment. Standard values are either IN (inches) or CM (centimeters).
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.



Step 6 - Choose special services per package (Optional)

In the final step of the eCart Shipping Rate Request wizard, you're given an opportunity to set values for optional special services. Not all services are available for all service types (FedEx Ground and FedEx Express).



Some of these settings may not apply depending on your chosen carrier type.

Service	Value	Valid for
Residential delivery		All
Residential pickup		FedEx Ground
Inside delivery		FedEx Express
Inside pickup		FedEx Express
Saturday delivery		FedEx Express
Saturday pickup		FedEx Express
Hold at location		FedEx Express
Dry ice		FedEx Express
Dangerous goods		FedEx Express
Home delivery type		FedEx Ground
Signature required		FedEx Ground

Value:  

The following describes each of the optional service settings available. When you have made your selections, click the Finish button to complete the wizard:

- **Residential delivery:** Sets whether residential delivery is chosen. Available for all services.
- **Residential pickup:** Sets whether residential pickup is chosen. Available for FedEx Ground only.
- **Inside delivery:** Sets whether inside delivery is chosen. Available for FedEx Express only.
- **Saturday delivery:** Sets whether Saturday delivery is chosen. Available for FedEx Express only.
- **Hold at location:** Sets whether the hold at location service is chosen. Available for FedEx Express only.
- **Dry ice:** Sets whether the dry ice option is chosen. Available for FedEx Express only.
- **Dangerous goods:** Sets whether the dangerous goods option is chosen. Acceptable values are Accessible or Inaccessible; available for FedEx Express only.
- **Home delivery type:** Sets whether home delivery is chosen. Available for FedEx Ground only.
- **AOD:** Sets whether the Acknowledge On Delivery option is chosen. Available for FedEx Ground only.
- **Autopod:** Sets whether Automatic Proof Of Delivery option is chosen. Available for FedEx Ground only.
- **Non-standard package:** Sets whether non-standard package option is chosen. Available for FedEx Ground only.
- **COD:** Sets whether the Cash on Delivery option is chosen. Available for all services.
- **COD amount:** Sets whether the COD amount option is chosen. Available for FedEx Ground only. Acceptable values are Any, Guaranteed Funds or Cash.

Configuring eCart UPS Shipping Rate Request

The UPS shipping service offered in eCart returns shipping quotes in real-time from the UPS Web service. The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

Applying this server behavior updates the eCart Object automatically to apply the shipping rate retrieved to the cart total. See [Discounts, Charges, Taxes and Shipping](#) for more information.

To add this service to your shopping cart, you'll need to supply a range of both general information (such as your UPS user ID and password) and cart specific information (including which form elements the shipping destination details should come from). For these reasons, it's important to set up your UPS account as well as the database schema and page forms prior to incorporating the shipping service.

The UPS shipping service may be added to the site through any page that includes a shopping cart; typically the service is added before the shopping cart display is inserted into the page, but after any required forms have been included in the site. A wizard type interface guides you through the process of setting up the service. The following section details the steps for configuring the eCart UPS Shipping Rate Request feature.

To access the eCart UPS Shipping Rate Request wizard:

The eCart UPS Shipping feature is implemented as a server behavior. To begin the configuration process, open the Server Behaviors panel and choose Add (+). From the list, choose eCart > Shipping > UPS.

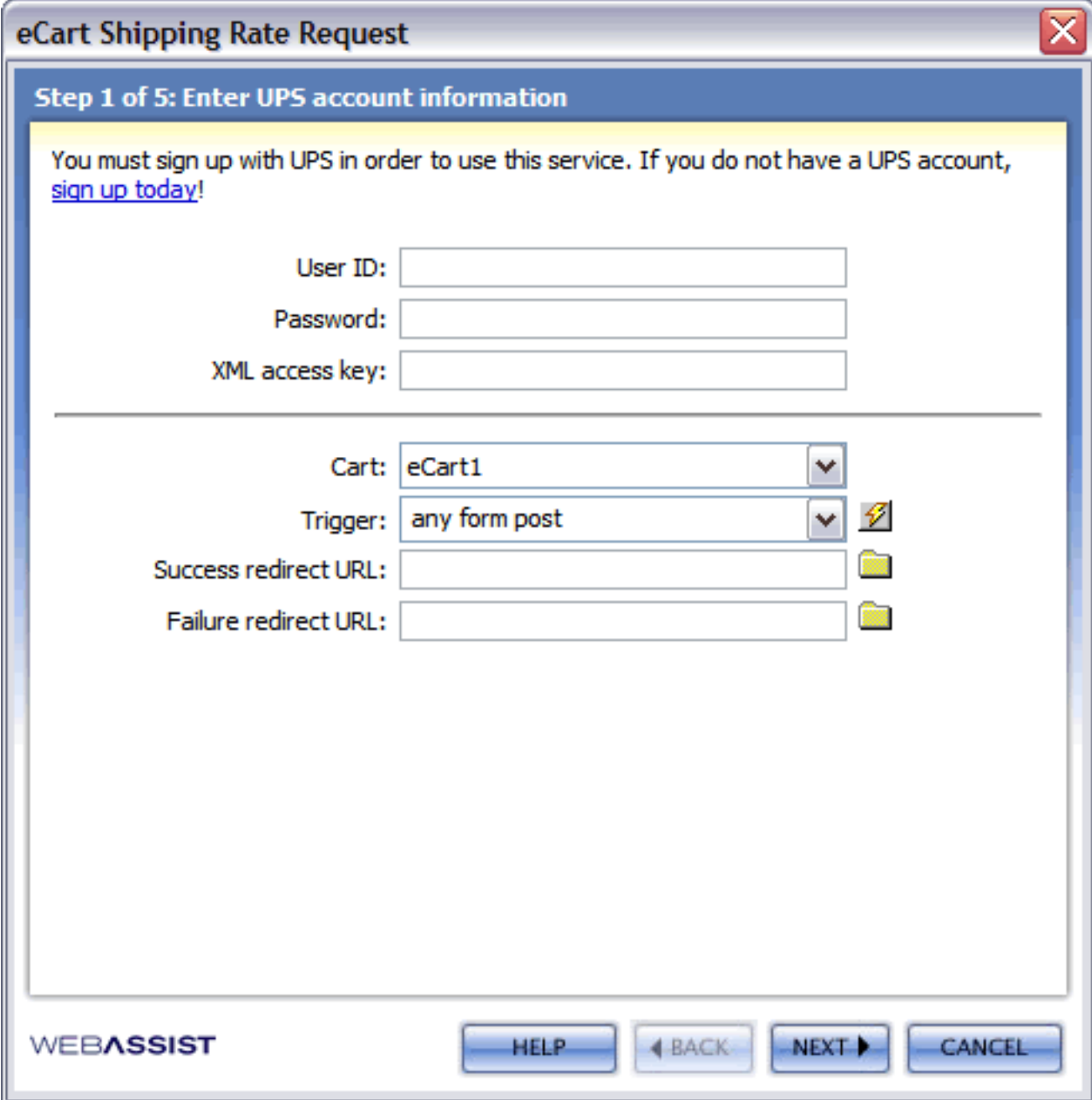
To use the eCart UPS Shipping Rate Request wizard:

1. Open a page to insert the code to query the shipping service and return the shipping charge.

If you are gathering the customer shipping information from a form, apply the eCart UPS Shipping Rate Request server behavior to the page specified as the action in that form, which is often the checkout page. If you are retrieving the shipping information from a database, place the server behavior on a page containing the recordset information prior to the checkout page.

2. From the Server Behavior panel, choose Add (+) and, from the list, select eCart > Shipping > UPS.

The eCart UPS Shipping Rate Request wizard opens:



The screenshot shows a wizard window titled "eCart Shipping Rate Request" with a close button in the top right corner. The window has a blue header bar that says "Step 1 of 5: Enter UPS account information". Below the header, there is a yellow background area with the text: "You must sign up with UPS in order to use this service. If you do not have a UPS account, [sign up today!](#)".

Below this text are three input fields:

- User ID:
- Password:
- XML access key:

A horizontal line separates these fields from the configuration section below. The configuration section includes:

- Cart: (dropdown menu)
- Trigger: (dropdown menu with a lightning bolt icon)
- Success redirect URL: (with a folder icon)
- Failure redirect URL: (with a folder icon)

At the bottom of the window, there is a "WEBASSIST" logo on the left and four buttons: "HELP", "BACK", "NEXT", and "CANCEL".

3. The wizard gathers the information necessary to communicate with the shipping service through its five steps:

- o [Enter UPS account information](#)
- o [Enter your shipment information](#)
- o [Select your shipping origin](#)
- o [Select your shipping destination](#)
- o [Set your package information](#)

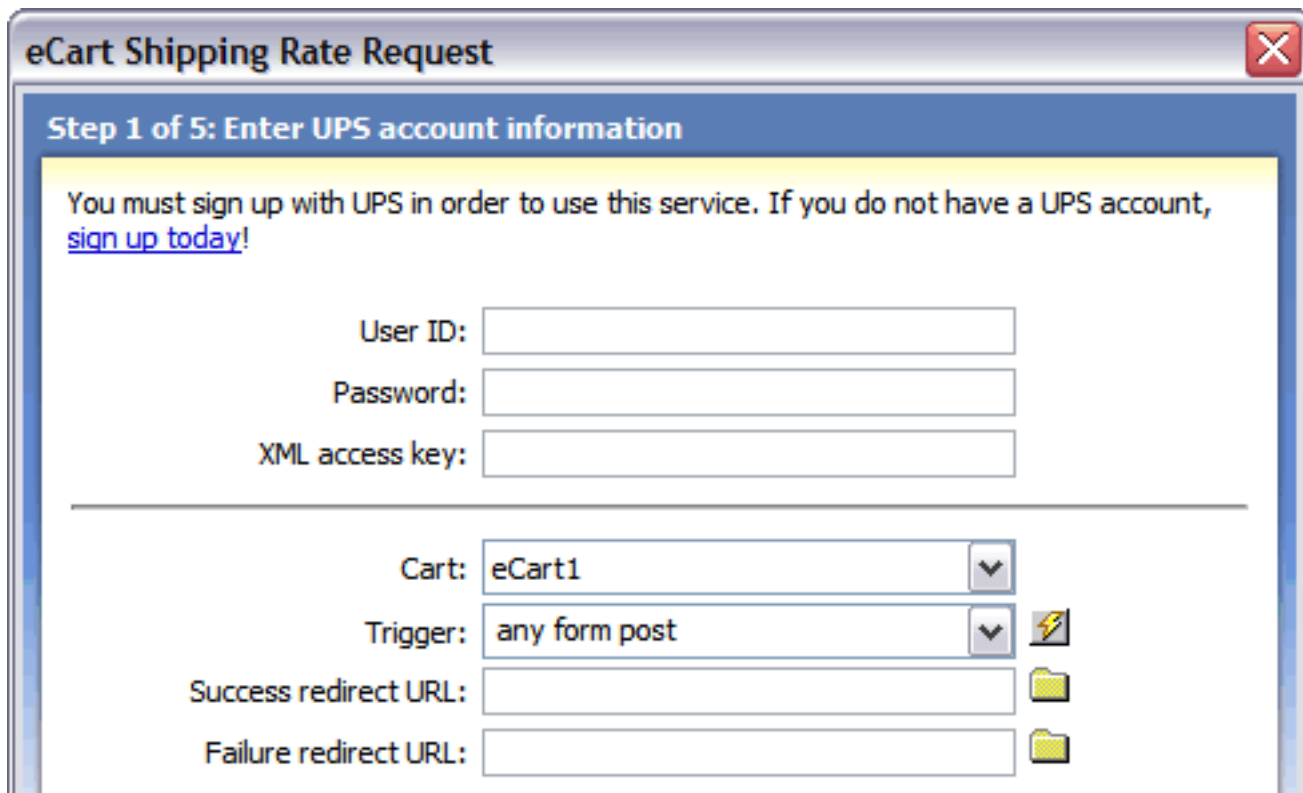
4. Use the Back and Next buttons to navigate through each of these sections.

5. Upon completion of the eCart UPS Shipping Rate Request, click Finish to apply the code to the shopping cart object and the server behavior that queries the shipping provider.

The wizard creates and activates a rule in the Shipping tab of the eCart Object. You can modify the rule by selecting it and choosing Edit.

Step 1 - Enter UPS account information

In the first step of the eCart UPS Shipping Rate Request wizard, you'll need to enter general information about your account as well as about how the shipping service should be initiated and redirected.

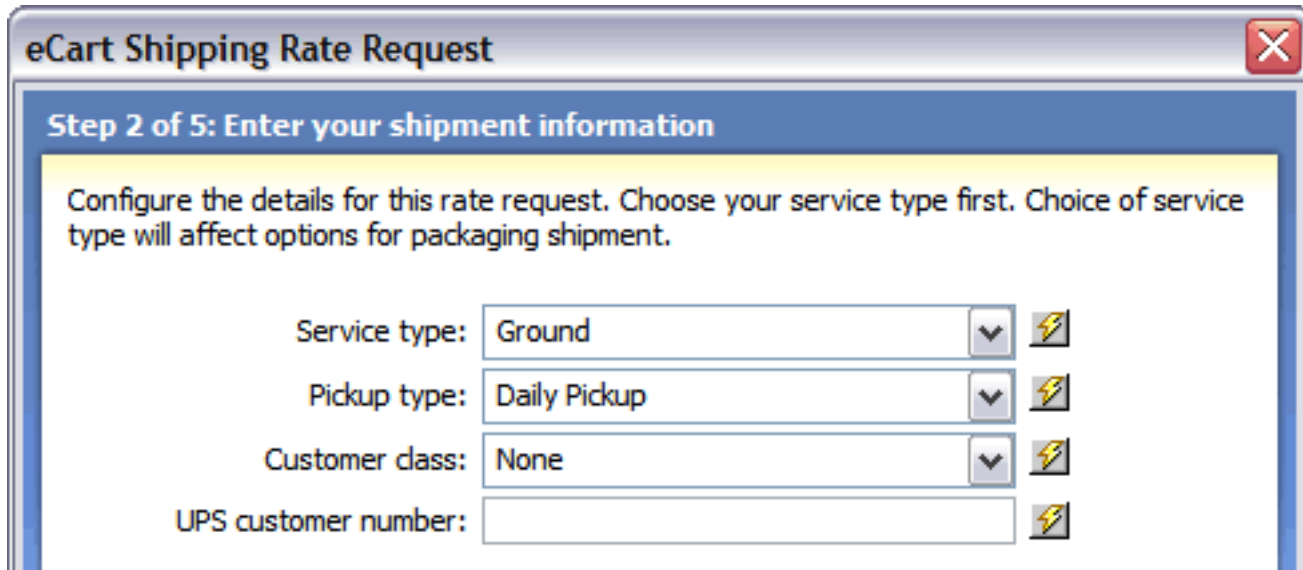


The following describes each of the configuration settings available and how they affect the shipping configuration. When you have made your selections, click the Next button to proceed to the following step:

- **User ID:** Enter the UPS user ID assigned to your account.
- **Password:** Enter the password for your account.
- **XML access key:** Enter the XML access key for your account.
- **Cart:** This is the name of the eCart Object specific to your shopping cart implementation associated with this request. This field is only updateable if you have multiple shopping cart implementations applied to the page the rate request is being applied to.
- **Trigger:** Select the trigger that will call the shipping service. The following triggers are available:
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the query to the shipping service. This is typically used when a form on a different page contains information needed by the shipping service.
 - **Before Page Load:** The query to the service is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the query to the shipping service. This is used typically when applying the server behavior to the same page that contains a form with the necessary shipping information.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
- **Success redirect URL:** The path to the page you want displayed if the shipping service returns a shipping quote successfully. Typically this field is left blank except for testing purposes.
- **Failure redirect URL:** The path to the page you want displayed if the shipping service returns an error. Typically this field is left blank except for testing purposes or to inform customers of a communications problem gathering the shipping quote.

Step 2 – Enter your shipment information

In step 2 of the wizard, you're asked to specify the types of services you'd like. Each of the choices allow you to chose either static values or dynamic values.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button in the top right corner. The main heading is "Step 2 of 5: Enter your shipment information". Below this, a yellow highlighted box contains the instruction: "Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment." There are four input fields, each with a dropdown arrow and a lightning bolt icon to its right, indicating dynamic values:

- Service type: Ground
- Pickup type: Daily Pickup
- Customer class: None
- UPS customer number: (empty text box)

The following describes each of the configuration settings available and how they affect the shipping service options. When you have made your selections, click the Next button to proceed to the following step:

- **Service type:** Selects the specific kind of service desired. To allow the customer to choose the option, insert the UPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
 - **Pickup type:** Determines the type of pickup desired. Options are Daily Pickup, Customer Counter, One Time Pickup, On Call Air Pickup, Suggested Retail (UPS Store), Letter Center and Air Service Center; typically these are set to one type by the shipper.
 - **Customer class:** Sets the customer class. Options are None, Wholesale, Occasional Pickup and Retail. Again, these are typically set to one value by the shipper.
 - **UPS customer number:** Sets the customer number of the shipper; typically this is a single value.
-

Step 3 - Select your shipping origin

For UPS to determine the proper rate quote, you must indicate the shipping location. If the online store ships from different locations for different item, the address values may be assigned to dynamic data.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 3 of 5: Select your shipping origin". Below the heading is a yellow background area with the following text: "Enter the location from which the order will be shipped. Postal code is the only required field for shipments within the United States. Countries marked with an asterisk require a postal code." Below this text are four input fields, each with a lightning bolt icon to its right:

- City: [Text input field]
- State: [Dropdown menu showing "Not Applicable"]
- Postal code: [Text input field]
- Country: [Dropdown menu showing "United States*"]

The following describes each of the shipping origin settings available. When you have made your selections, click the Next button to proceed to the following step:

- **City:** Enter or choose the city where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
 - **State:** Enter or choose the state or province where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
 - **Postal Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
 - **Country:** Enter or choose the country where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
-

Step 4 – Select your shipping destination

UPS requires the shipping destination to determine the correct rate quote. Typically, the required values are assigned to form elements filled out at run-time by the shopper.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 4 of 5: Select your shipping destination". Below the heading is a yellow highlighted text area that reads: "Enter the location to which the order will be shipped. Postal code is the only required field for deliveries within the United States. Countries marked with an asterisk require a postal code." Below this text are four form fields, each with a lightning bolt icon to its right:

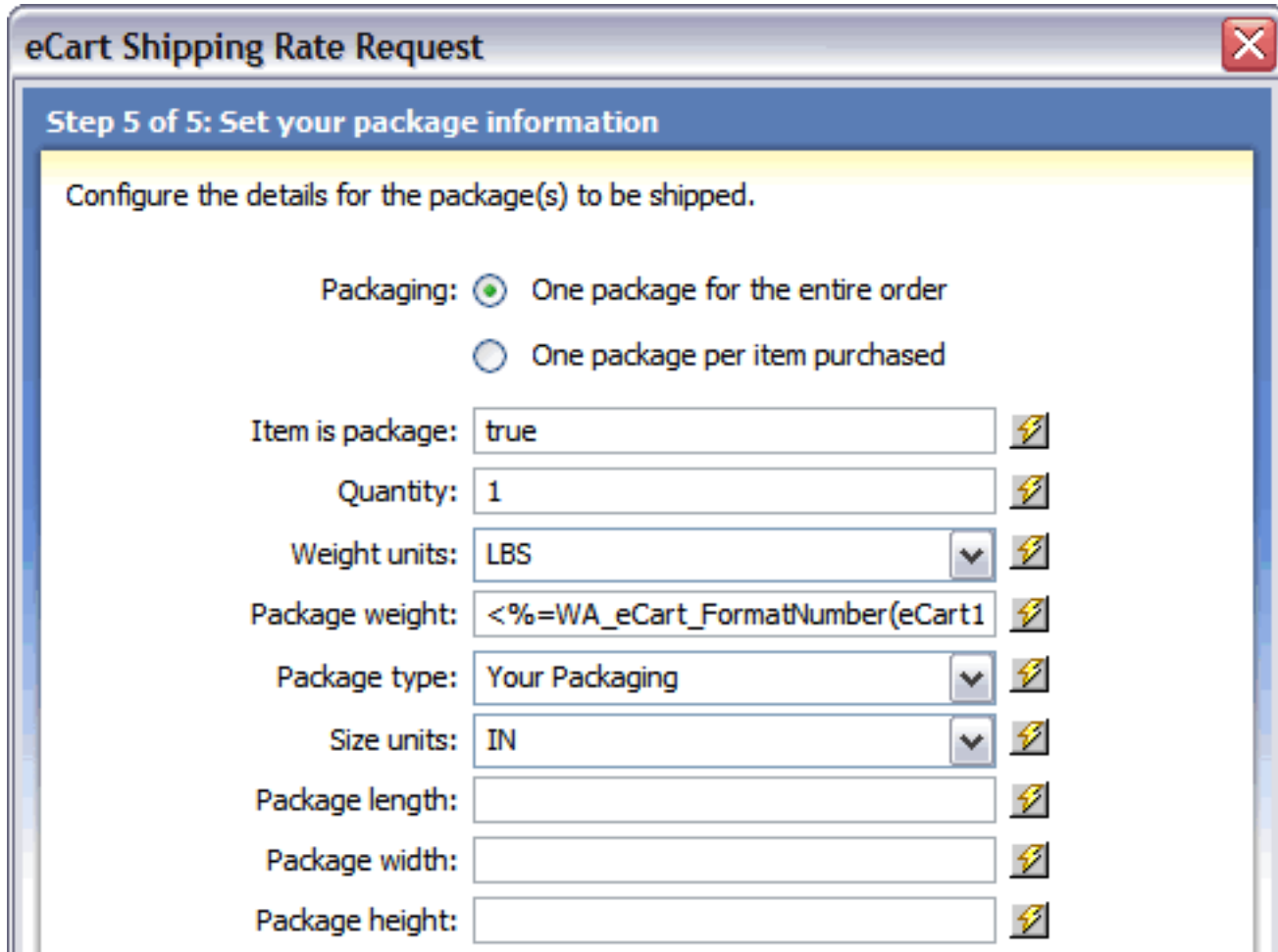
- City:** A text input field.
- State:** A dropdown menu currently showing "Not Applicable".
- Postal code:** A text input field.
- Country:** A dropdown menu currently showing "United States*".

The following describes each of the shipping destination settings available. When you have made your selections, click the Next button to proceed to the following step:

- **City:** Enter or choose the city where your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned to a field in the shipping information form.
 - **State:** Enter or choose the state or province where your packages are shipped to. Insert the provided UPS Shipping Destination State/Province snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
 - **Postal Code:** Enter the postal code that your packages are shipped to. Insert the provided UPS Shipping Destination Postal Code snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
 - **Country:** Enter or choose the country where your packages are shipped to. Insert the provided UPS Shipping Destination Country snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element.
-

Step 5 – Enter your shipment information

In step 5 of the wizard, you're asked to specify the details of the packages to be shipped. You can choose between setting up one package for the entire order or a separate package for every item. Every field on this step of the wizard is required.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading is "Step 5 of 5: Set your package information". Below this, a yellow highlighted area contains the instruction: "Configure the details for the package(s) to be shipped." The form includes the following fields:

- Packaging:** Two radio buttons. The first is selected and labeled "One package for the entire order". The second is labeled "One package per item purchased".
- Item is package:** A text input field containing "true" and a lightning bolt icon.
- Quantity:** A text input field containing "1" and a lightning bolt icon.
- Weight units:** A dropdown menu showing "LBS" and a lightning bolt icon.
- Package weight:** A text input field containing "<%=WA_eCart_FormatNumber(eCart1" and a lightning bolt icon.
- Package type:** A dropdown menu showing "Your Packaging" and a lightning bolt icon.
- Size units:** A dropdown menu showing "IN" and a lightning bolt icon.
- Package length:** An empty text input field and a lightning bolt icon.
- Package width:** An empty text input field and a lightning bolt icon.
- Package height:** An empty text input field and a lightning bolt icon.

The following describes each of the package settings available. When you have made your selections, click the Finish button to complete the wizard:

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is package:** Determines whether the item in the cart is to be shipped via UPS or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used. Choose the Your Packaging option if you use non-UPS packaging; to use UPS packaging choose one of the following: UPS Express Envelope, UPS Express Box, UPS Pak, UPS Tube, UPS 10Kg Box, or UPS 25Kg Box. If you previously selected UPS Ground as your service, only the Your Packaging, UPS 10Kg Box and UPS 25Kg Box options are available.
- **Size units:** Declares the type of size units to be used in calculating the total size of the shipment. Standard values are either IN (inches) or CM (centimeters).
- **Package length:** Sets the length of the package to be shipped.
- **Package width:** Sets the width of the package to be shipped.
- **Package height:** Sets the height of the package to be shipped.

Configuring eCart USPS Shipping Rate Request

The USPS shipping service offered in eCart returns shipping quotes in real-time from the USPS Web service. The Checkout wizard is the most direct method for configuring the rate request server behaviors for your shopping cart and applying them to the page. However, these server behaviors can also be updated directly after completing the Checkout wizard, and can also be applied individually to a page specific to your needs.

Applying this server behavior updates the eCart Object automatically to apply the shipping rate retrieved to the cart total. See [Discounts, Charges, Taxes and Shipping](#) for more information.

To add this service to your shopping cart, you'll need to supply a range of both general information (such as your USPS user ID and password) and cart specific information (including which form elements the shipping destination details should come from). For these reasons, it's important to set up your USPS account as well as the database schema and page forms prior to incorporating the shipping service.

The USPS shipping service may be added to the site through any page that includes a shopping cart; typically the service is added before the shopping cart display is inserted into the page, but after any required forms have been included in the site. A wizard type interface guides you through the process of setting up the service. The following section details the steps for configuring the eCart USPS Shipping Rate Request feature.

To access the eCart USPS Shipping Rate Request wizard:

The eCart USPS Shipping feature is implemented as a server behavior. To begin the configuration process, open the Server Behaviors panel and choose Add (+). From the list, choose eCart > Shipping > USPS.

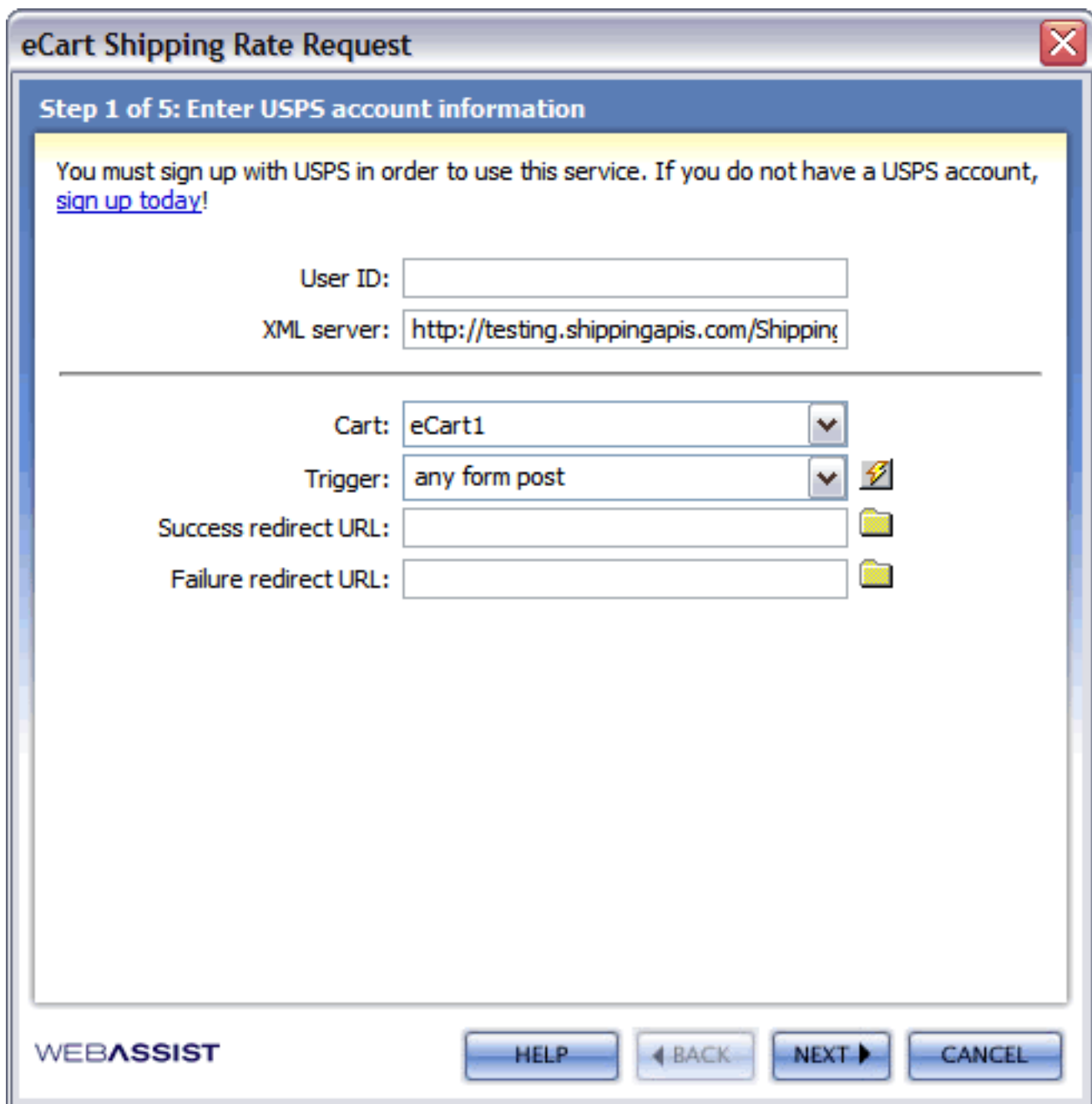
To use the eCart USPS Shipping Rate Request wizard:

1. Open a page to insert the code to query the shipping service and return the shipping charge.

If you are gathering the customer shipping information from a form, apply the eCart USPS Shipping Rate Request server behavior to the page specified as the action in that form, which is often the checkout page. If you are retrieving the shipping information from a database, place the server behavior on a page containing the recordset information prior to the checkout page.

2. From the Server Behavior panel, choose Add (+) and, from the list, select eCart > Shipping > USPS.

The eCart USPS Shipping Rate Request wizard opens:



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main content area has a blue header "Step 1 of 5: Enter USPS account information". Below the header is a yellow warning box with the text: "You must sign up with USPS in order to use this service. If you do not have a USPS account, [sign up today!](#)".

The form contains the following fields:

- User ID:
- XML server:
- Cart: (dropdown menu)
- Trigger: (dropdown menu with a lightning bolt icon)
- Success redirect URL: (with a folder icon)
- Failure redirect URL: (with a folder icon)

At the bottom left is the "WEBASSIST" logo. At the bottom right are four buttons: "HELP", "BACK", "NEXT", and "CANCEL".

3. The wizard gathers the information necessary to communicate with the shipping service through its five steps:

- o [Enter USPS account information](#)
- o [Enter your shipment information](#)
- o [Select your shipping origin](#)
- o [Select your shipping destination](#)
- o [Set your package information](#)

4. Use the Back and Next buttons to navigate through each of these sections.

5. Upon completion of the eCart USPS Shipping Rate Request, click Finish to apply the code to the shopping cart object and the server behavior that queries the shipping provider.

The wizard creates and activates a rule in the Shipping tab of the eCart Object. You can modify the rule by selecting it and choosing Edit.

Step 1 - Enter USPS account information

In the first step of the eCart USPS Shipping Rate Request wizard, you'll need to enter general information about your account as well as about how the shipping service should be initiated and redirected.

eCart Shipping Rate Request

Step 1 of 5: Enter USPS account information

You must sign up with USPS in order to use this service. If you do not have a USPS account, [sign up today!](#)

User ID:

XML server:

Cart:

Trigger:

Success redirect URL:

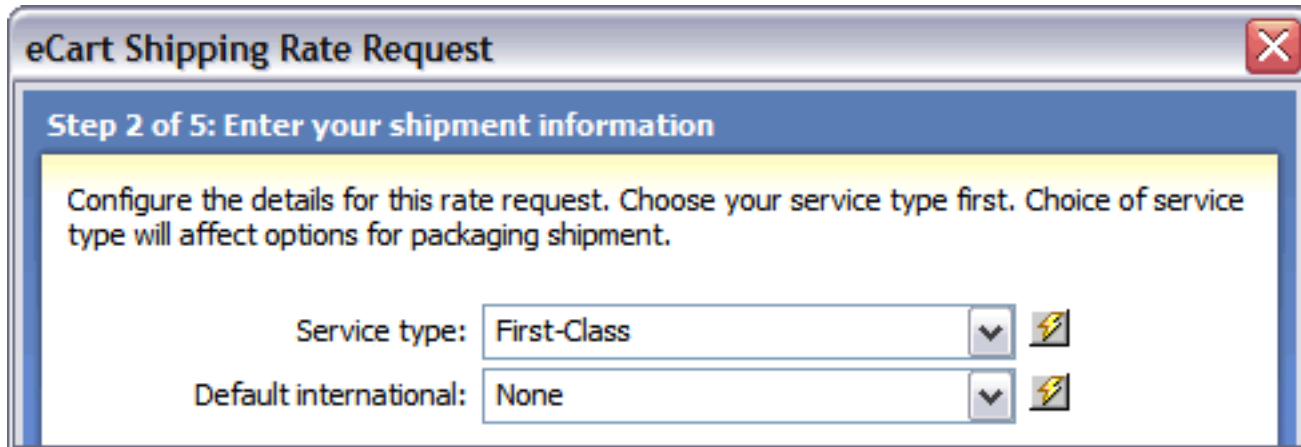
Failure redirect URL:

The following describes each of the configuration settings available and how they affect the shipping configuration. When you have made your selections, click the Next button to proceed to the following step:

- **User ID:** Enter the USPS user ID assigned to your account.
- **XML server :** Enter the full path to the secure XML server for your account (should be available by default).
- **Cart:** This is the name of the eCart Object specific to your shopping cart implementation associated with this request. This field is only updateable if you have multiple shopping cart implementations applied to the page the rate request is being applied to.
- **Trigger:** Select the trigger that will call the shipping service. The following triggers are available:
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the query to the shipping service. This is typically used when a form on a different page contains information needed by the shipping service.
 - **Before Page Load:** The query to the service is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the query to the shipping service. This is used typically when applying the server behavior to the same page that contains a form with the necessary shipping information.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
- **Success redirect URL:** The path to the page you want displayed if the shipping service returns a shipping quote successfully. Typically this field is left blank except for testing purposes.
- **Failure redirect URL:** The path to the page you want displayed if the shipping service returns an error. Typically this field is left blank except for testing purposes or to inform customers of a communications problem gathering the shipping quote.

Step 2 – Enter your shipment information

In step 2 of the wizard, you're asked to specify the types of services you'd like. Each of the choices allow you to chose either static values or dynamic values.



eCart Shipping Rate Request [Close]

Step 2 of 5: Enter your shipment information

Configure the details for this rate request. Choose your service type first. Choice of service type will affect options for packaging shipment.

Service type: [v] [⚡]

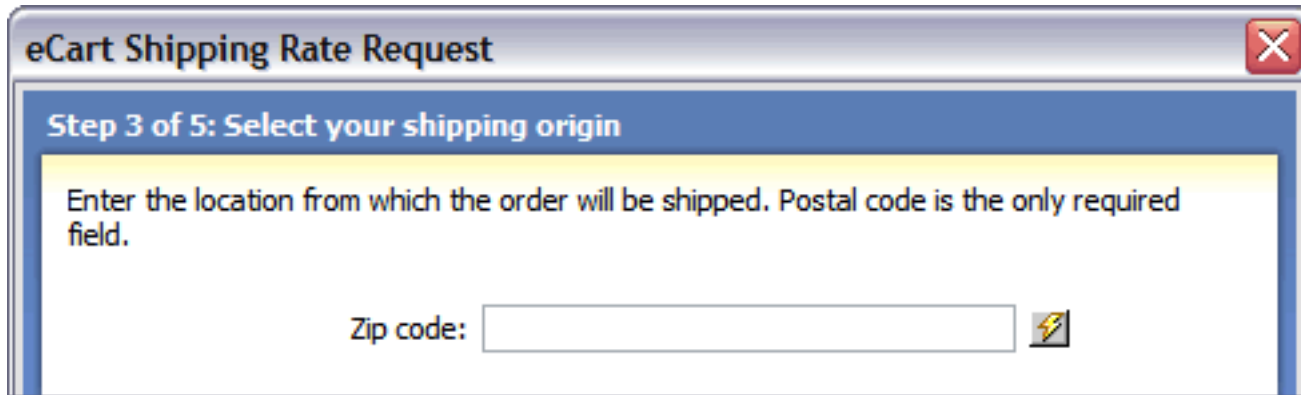
Default international: [v] [⚡]

The following describes each of the configuration settings available and how they affect the shipping service options. When you have made your selections, click the Next button to proceed to the following step:

- **Service type:** Selects the specific kind of domestic service desired. To allow the customer to choose the option, insert the USPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
 - **Default international:** Selects the specific kind of international service desired. To allow the customer to choose the option, insert the USPS Service Type snippet in the shipping form and select that form list's name from the Dynamic Data dialog.
-

Step 3 - Select your shipping origin

For USPS to determine the proper rate quote, you must indicate the shipping location. If the online store ships from different locations for different item, the location value may be assigned to dynamic data.



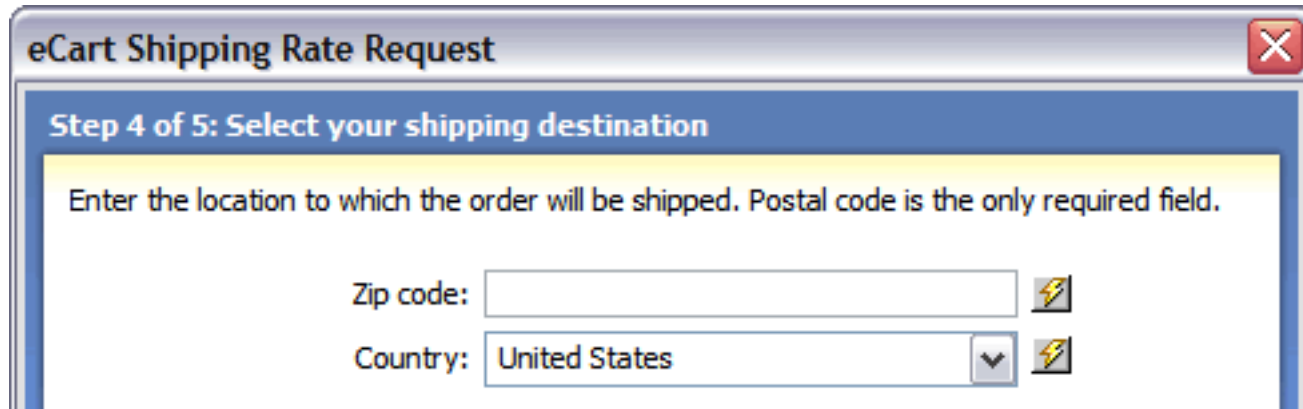
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 3 of 5: Select your shipping origin". Below this heading, a yellow highlighted area contains the text: "Enter the location from which the order will be shipped. Postal code is the only required field." Below this text is a text input field labeled "Zip code:" followed by a lightning bolt icon, which is used to open the Dynamic Data dialog box.

The following describes the shipping origin settings available. When you have made your selection, click the Next button to proceed to the following step:

- **Zip Code:** Enter the postal code that your packages are shipped from. Select the lightning bolt to open the Dynamic Data dialog box if this value is assigned dynamically.
-

Step 4 – Select your shipping destination

USPS requires the shipping destination to determine the correct rate quote. Typically, the required values are assigned to form elements filled out at run-time by the shopper.



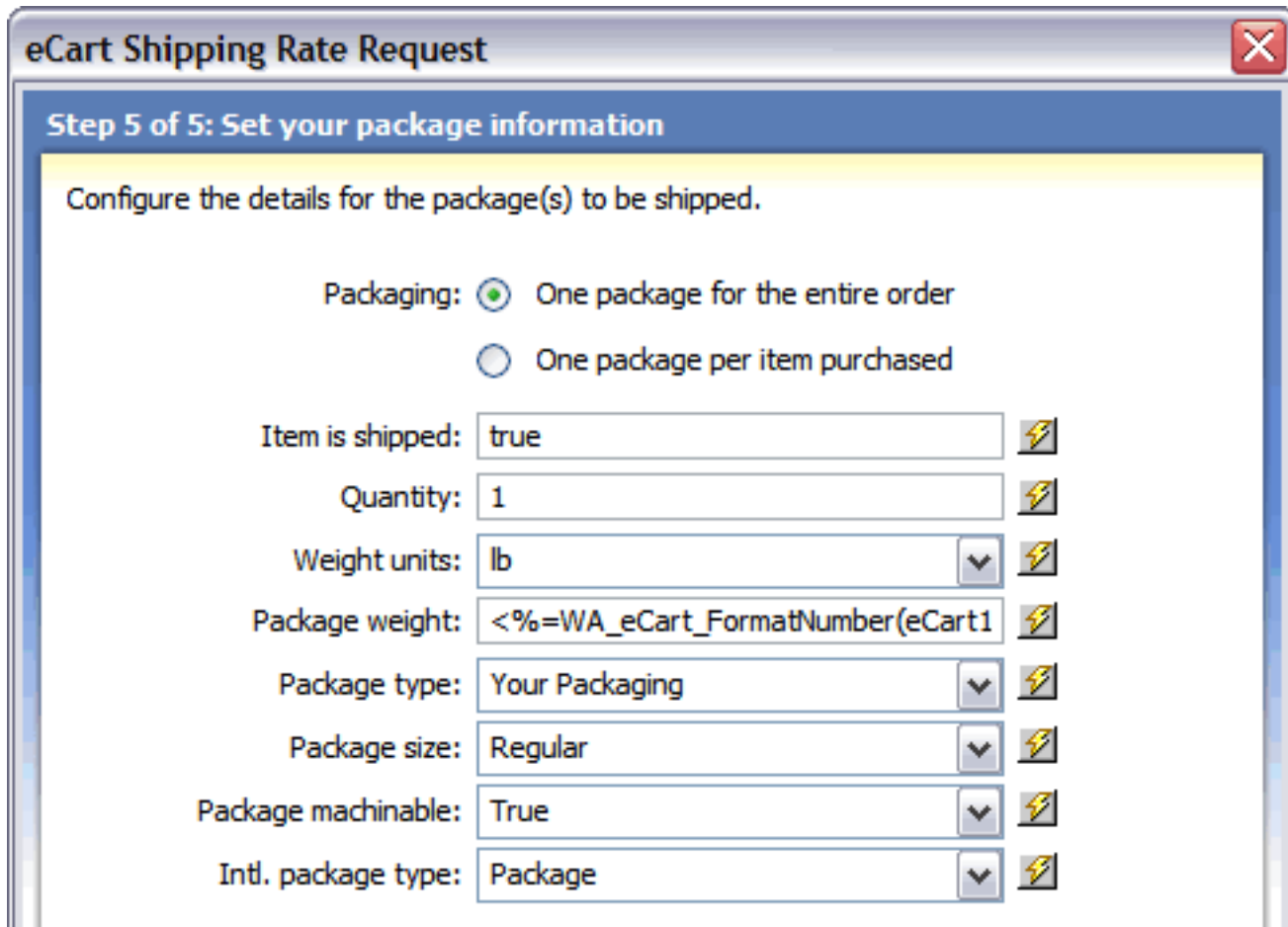
The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button (X) in the top right corner. The main heading inside the dialog is "Step 4 of 5: Select your shipping destination". Below the heading is a yellow highlighted instruction: "Enter the location to which the order will be shipped. Postal code is the only required field." There are two form elements: a text input field labeled "Zip code:" with a lightning bolt icon to its right, and a dropdown menu labeled "Country:" with "United States" selected and a lightning bolt icon to its right.

The following describes each of the shipping destination settings available. When you have made your selections, click the Next button to proceed to the following step:

- **Zip code:** Enter the postal code that your packages are shipped to. Select the lightning bolt to open the Dynamic Data dialog box and assign the field to the shipping zip code form element in your checkout form for the most common usage scenario.
 - **Country:** Enter or choose the country where your packages are shipped to. Insert the provided Shipping Countries snippet in the shipping information form and select the lightning bolt to open the Dynamic Data dialog box and assign the field to the added form element in your checkout form.
-

Step 5 – Enter your shipment information

In step 5 of the wizard, you're asked to specify the details of the packages to be shipped. You can choose between setting up one package for the entire order or a separate package for every item. Every field on this step of the wizard is required.



The screenshot shows a dialog box titled "eCart Shipping Rate Request" with a close button in the top right corner. The main heading is "Step 5 of 5: Set your package information". Below this, it says "Configure the details for the package(s) to be shipped." There are two radio button options for "Packaging": "One package for the entire order" (selected) and "One package per item purchased". Below these are several input fields, each with a lightning bolt icon to its right, indicating they are dynamic data fields:

- Item is shipped: true
- Quantity: 1
- Weight units: lb
- Package weight: <%=WA_eCart_FormatNumber(eCart1
- Package type: Your Packaging
- Package size: Regular
- Package machinable: True
- Intl. package type: Package

The following describes each of the package settings available. When you have made your selections, click the Finish button to complete the wizard:

- **Packaging:** Sets whether all of the items in the cart should be shipped together or separately.
- **Item is shipped:** Determines whether the item in the cart is to be shipped via UPS or not. This setting is particularly useful in an online store which has a mixed of physical and digital goods; to apply this feature, assign the proper value to a column in the product database table and choose that value from the Dynamic Data dialog box.
- **Weight units:** Declares the type of weight units to be used in calculating the total weight of the shipment. Standard values are either LBS (pounds) or KGS (kilograms).
- **Package weight:** Determines the total weight of the package. Typically this field is set to the calculated shopping cart column TotalWeight by selecting the lightning bolt and expanding the shopping cart node in the Dynamic Data dialog box.
- **Package type:** Sets the type of packaging used.
- **Package size:** Sets the size of the package for the rate request. Options are:
 - Regular: package length plus girth is 84 inches or less
 - Large: package length plus girth that measures more than 84 inches (but less than 108 inches) and weighs less than 15 pounds.
 - Oversized: package length plus girth is more than 108 but not more than 130 inches
- **Package machinable:** Applies only to Parcel Post. The size, content, and weight of a Parcel Post package determine whether it is machinable or nonmachinable. For Parcel Post, the value entered must be either "True" or "False." The "True" or "False" value specifically applies to the size and content of the package. A package is deemed nonmachinable if it weighs less than 6 ounces or over 35 pounds.
- **International package type:** If the shipment is international, sets the valid packaging type.

Storing and retrieving shopping cart data

eCart provides several methods for storing and retrieving your shopping cart data. These methods can be used, among other things, to:

- Store basic information about an order into a database before submitting it for processing.
- Save complete order details after an order has been approved.
- Establishing a recurring order so customers can easily re-order previously purchased goods items.
- Maintain a wish list and retrieve it upon demand.
- Save a cart value in a session variable for use in the application.

Five server behaviors included in eCart control this functionality:

- [Store Cart Summary in Database](#)
- [Store Cart Details in Database](#)
- [Get Contents from Recordset](#)
- [Get Contents from Form](#)
- [Set Session Value](#)

Storing cart information in common usage scenarios is implemented when using the [Checkout wizard](#) to develop your checkout process. You can use the Store Cart Summary and Store Cart Details server behaviors to edit the configuration applied by the wizard, or to implement this functionality outside of the wizard specific to your requirements.

Store Cart Summary in Database

To keep a record of the current order, use the eCart Store Cart Summary in Database server behavior. This server behavior inserts a new record in a database table with basic information about the order.

The server behavior may be triggered by the press of any form button on the page — in general or a specific button — or other actions including the loading of the page or any form posting to the current page; you can even execute the server behavior if a particular session variable or other dynamic value is defined.

After the record has been inserted, the user may be re-directed to another page.

Databases typically use two tables to handle orders, a general Orders table and an OrderDetails table. The two tables are linked by key field, usually OrderID; each record in the OrderDetails table includes a column called something like DetailsOrderID that corresponds to the OrderID column in the Orders table.

This makes it possible to retrieve a summary of all the orders placed by a particular customer by querying just the Orders table or to get a listing of all the products on a particular order by querying the OrderDetails page.

You can also employ a database view (also called a query of a query) that combines information from the two tables.

The Store Cart Summary in Database server behavior adds records to the more general Orders table. Frequently this server behavior is used together with the Store Cart Details in Database server behavior to save all the cart information at the same time.

To apply the Store Cart Summary in Database server behavior:

- From the Server Behaviors panel, choose Add (+) and select eCart > Database Manipulation > Store Cart Summary in Database server behavior.

- Select the action you want to initiate the server behavior from the Trigger list. The options are:
 - Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the record insertion.
 - Before Page Load:** The record insert is triggered when the current page that the server behavior is applied to is loaded.
 - On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the record insertion.
 - eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.

- Select the desired connection from the Connection list.

If necessary, you can create a new connection by clicking the Define button.

- Choose a table to insert the shopping cart information into from the Table list.

The table chosen should have, at a minimum, columns for an order ID (the unique key field) and order total.

Other useful columns include ones for an order reference ID, a user ID, the shipping amount and the tax on the order.

- Select the key field for the table from the Order ID list.

- In the Store As field, enter the name of the session variable you want to store the unique ID in.

Pick a name that is easy to identify and re-use when referencing a particular order, such as OrderNumber.

The session variable defined here will automatically appear in the Bindings panel and be available from Dynamic Data dialogs.

- Bind values to the table columns listed in the columns area. Either static or dynamic values can be used.

To bind a static value, enter it in the Value field for a selected column.

To bind a dynamic value, choose the lightning bolt to open the Dynamic Data dialog box and select an item from a shopping cart entry; click OK to close the Dynamic Data dialog.

Choose a data type from the Type list to confirm your selection.

Dynamic values can also be entered by hand in the Value field.

In the example figure, the following values are assigned:

- OrderID - Left blank and not updated. OrderID is an autonumber field.
 - OrderReferenceID - Choose [Cart ID] from the shopping cart dynamic data entry. The code appropriate to your server model is written to include the SessionID, a unique value.
 - OrderUserID - In this example, the user has logged in and his unique user ID has been assigned to a session variable, UserID. The code for writing this session appropriate to the server model has been entered manually into the Value field and the binding updated.
 - OrderShipping - Shipping is retrieved from the eCart object data binding for the shipping value in the cart.
 - OrderTax - Tax is retrieved from the eCart object data binding for the tax value in the cart.
 - OrderTotal - From the shopping cart entry of the Dynamic Data dialog, the [GrandTotal] item is selected and the appropriate code is written.
- If you want another page to be displayed after the record insertion is complete, enter the path to that file in the Redirect field or click the folder icon and choose your file from the Select File dialog box.
 - Click OK when you're done.

You can modify the server behavior by double-clicking its entry in the Server Behaviors panel.

Store Cart Details in Database

To keep a record of all the items included in a particular order, add the eCart server behavior Store Cart Details in Database to a page. This server behavior inserts a new record for each item in the shopping cart.

The server behavior may be triggered by the press of any form button on the page — in general or a specific button — or other actions including the loading of the page or any form posting to the current page; you can even execute the server behavior if a particular session variable or other dynamic value is defined.

After the record has been inserted, the user may be re-directed to another page.

Databases typically use two tables to handle orders, a general Orders table and an OrderDetails table. The two tables are linked by key field, usually OrderID; each record in the OrderDetails table includes a column called something like DetailsOrderID that corresponds to the OrderID column in the Orders table.

This makes it possible to retrieve a summary of all the orders placed by a particular customer by querying just the Orders table or to get a listing of all the products on a particular order by querying the OrderDetails page.

You can also employ a database view (also called a query of a query) that combines information from the two tables.

The Store Cart Details in Database server behavior adds records to the OrderDetails table. Frequently this server behavior is used together with the Store Cart Summary in Database server behavior to save all the cart information at the same time.

To apply the Store Cart Details in Database server behavior:

1. From the Server Behaviors panel, choose Add (+) and select eCart > Database Manipulation > Store Cart Details in Database server behavior.

2. Select the action you want to initiate the server behavior from the Trigger list. The options are:
 - o **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the record insertion.
 - o **Before Page Load:** The record insert is triggered when the current page that the server behavior is applied to is loaded.
 - o **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the record insertion.
 - o **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - o **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - o **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - o **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - o **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - o **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
3. Select the desired connection from the Connection list.
If necessary, you can create a new connection by clicking the Define button.
4. Choose a table to insert the shopping cart information into from the Table list.
The table chosen should have, at a minimum, columns for a details order ID (the unique key field), an order ID, an item ID, and quantity. Other useful columns include ones for a product name, price and selected options.
5. Select the key field for the table from the Order ID list.

6. Bind values to the table columns listed in the columns area. Either static or dynamic values can be used.
To bind a static value, enter it in the Value field for a selected column.
To bind a dynamic value, choose the lightning bolt to open the Dynamic Data dialog box and select an item from a shopping cart entry; click OK to close the Dynamic Data dialog.
Choose a data type from the Type list to confirm your selection.
Dynamic values can also be entered by hand in the Value field.

- In the example figure, the following values are assigned:
- o DetailsID - Left blank and not updated. DetailsID is an autonumber field.
 - o DetailOrderID - Expand the Session entry in the Dynamic Data dialog and select OrderNumber. The OrderNumber session variable is defined in the Save Cart Summary in Database Server behavior.
 - o DetailProductID - The ID column from the shopping cart dynamic data entry is selected.
 - o DetailProductName - The Name column from the shopping cart dynamic data entry is selected.
 - o DetailQuantity - The Quantity column from the shopping cart dynamic data entry is selected.
 - o DetailPrice - The Price column from the shopping cart dynamic data entry is selected.
 - o DetailOption1, DetailOption2 - No options are specified for this example, so these entries are left blank. For an entry to be left blank, the associated column in the database table must be able to accept zero length (null) values.
7. If you want another page to be displayed after the record insertion is complete, enter the path to that file in the Redirect field or click the folder icon and choose your file from the Select File dialog box.
 8. Click OK when you're done.

You can modify the server behavior by double-clicking its entry in the Server Behaviors panel.

Get Contents from Recordset

Use the Get Contents From Recordset Server Behavior when you want to populate a shopping cart object from a recordset specified on the page. This server behavior is useful for displaying previously stored orders or wish lists.

The following are the most common usages for this server behavior:

- Retrieving a shopping cart instance - retrieve a stored shopping cart state for a returning user
- Retrieving purchase information - retrieve an order record which contains specifics about information purchased using the shopping cart.

Database and Recordset structure

To properly use this server behavior with database derived product lists, a database view that combines the stored eCart information table with the product catalog table should be used to create the recordset.

This is necessary because information may be contained in multiple tables, such as the Orders and OrderDetails table; referencing a single table for the recordset will not gather all the needed information to fill the shopping cart.

Use a Left Outer Join on the Product ID as the link when defining the recordset. Your SQL for the recordset should look something like this:

```
SELECT DetailProductID, DetailProductName, DetailQuantity, DetailPrice, ProdShortDesc, ProdWeight
FROM OrderDetails LEFT OUTER JOIN Products ON OrderDetails.DetailProductID = Products.ProdID
```

Filter the recordset to the appropriate record to be retrieved. It is easiest to find a record by filtering to its unique key. Finding a shopping cart record for a specific user would be retrieved by filtering for the User ID.

The following describes a minimal structure for a table used to store the state of a shopping cart (see [Store Cart Summary in Database](#) and [Store Cart Details in Database](#) for info on creating and populating this table).

- CartID - Used as a primary key only. No eCart information stored here.
- UserID - Associates each stored record with the proper user. This value does not come from the eCart.
- ProductID - Identifies the product in the eCart.
- CartQuantity - Records the amount of each product in the eCart
- CartIndex - When bound with the [Present Iteration] column of the eCart, preserves the order of the eCart contents.

To apply the Get Contents from Recordset server behavior:

1. Open the page to which the server behavior is to be applied.
2. In the Server Behaviors panel group, click the Plus (+) button and select eCart > Add to Cart > Get Contents from Recordset. The following dialog box will open:

Cart column	Value
ID	0
Name	
Description	
Weight	0
Quantity	0
Price	0

3. Select the eCart Object on the page which corresponds to the shopping cart whose contents are to be retrieved from the database. If only one eCart object is available, this field will be disabled.
4. Select when the shopping cart contents are to be retrieved from the recordset. The options are:
 - **If Selected Cart is Empty:** If the shopping cart selected from the Cart list is empty, the server behavior is triggered.
 - **Any Form Post:** Any form posted to the current page that the server behavior is applied to will trigger the data retrieval.
 - **Before Page Load:** The data retrieval is triggered when the current page that the server behavior is applied to is loaded.
 - **On Form Submit:** Any form submitted on the same page that the server behavior is applied to will trigger the data retrieval.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains no items.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Data:** Server-side code can be specified in the dynamic data interface (accessed through the lightning bolt) to create a trigger specific to your needs if one in the list does not meet your requirements.
5. Specify the recordset that the shopping cart is to be retrieved from.

The next step is to specify where the data for each column in the shopping cart is to be retrieved from. This is accomplished by binding each column in the cart to either the recordset column corresponding to its content or a specified value.

6. Bind values to the table columns listed in the Cart columns area.

To bind a dynamic value, choose the lightning bolt to open the Dynamic Data dialog box and select an item from the desired recordset entry; click OK to close the Dynamic Data dialog.

Static and dynamic values can also be entered by hand in the Value field.

For columns that do not apply, enter a 0 value.

Cart column	Value
ID	<%=Orders.Fields.Item("DetailProductID")...
Name	<%=Orders.Fields.Item("DetailProductNam...
Description	<%=Orders.Fields.Item("ProdShortDesc")...
Weight	<%=Orders.Fields.Item("ProdWeight").Val...
Quantity	<%=Orders.Fields.Item("DetailQuantity")...
Price	<%=Orders.Fields.Item("DetailPrice").Valu...

7. Click OK to apply the server behavior to the page.

The Get Contents from Recordset server behavior automatically runs when the page is loaded. You can modify the server behavior by double-clicking its entry in the Server Behaviors panel.

Get Contents from Form

The Get Contents From Form server behavior takes the data passed back from the merchant gateway and uses it to re-populate the shopping cart.

Currently, only PayPal is supported directly in eCart for re-population of shopping cart contents in this manner. PayPal's Instant Payment Notification (IPN) service posts not only the high level information for the transaction, but the line item details contained within the shopping cart. When the Get Contents From Form server behavior is run on the page, it correlates the line item details with the appropriate columns in the shopping cart.

Since the information for each cart column is available on the return post, the contents of the shopping cart can be re-created once that data has returned.

A custom option is available to configure support for gateways not directly supported in eCart.

The configure button allows you to correlate information returned to columns within the shopping cart specific to the type of POST returned by IPN. This data can be used for a shopping cart display for an order receipt, maintaining the shopping cart contents if a transaction fails, or even inserting shopping cart contents into the database.

Re-population of the shopping cart is triggered by a specified event on the page.

To apply the Get Contents From Form server behavior:

1. Open the page that will receive the POST from your merchant gateway.
2. In the Server Behaviors panel, click the plus button and select eCart > Add to Cart > Get Contents From Form

The Get Contents From Form dialog opens:

Get Contents From Form

Trigger: Valid Paypal Transaction

Cart: eCart1

Payment gateway: PayPal IPN Cart

Description: Creates server side code to capture returned data from a PayPal transaction including authorization status as well as credit card holder and shipping information. Auction information, if any, is also returned.

WEBASSIST

HELP OK CANCEL

3. Select the trigger on the page that will run the server-side code for this server behavior.
4. Select the shopping cart name that correlates to the eCart Object specific to your shopping cart implementation.
5. Select the IPN configuration specific to your needs from the list available.
6. If necessary, click Configure to modify the given IPN configuration specific to your needs.

The following shows the default configuration for PayPal available for re-populating the cart.

Configure Transaction Details

Cart Column	Value
ID	<%=Request.Form("item_number"+(displayInde...
Name	<%=Request.Form("item_name"+(displayIndex...
Description	
Weight	
Quantity	<%=Request.Form("quantity"+(displayIndex+1...
Price	<%=parseFloat(Request.Form("mc_gross_" + di...

Value: <%=Request.Form("item_number"+(displayIndex

Update

OK Cancel Help

Specialized server-side code is provided for the ID, Name, Quantity, and Price columns. This code takes the line item details information, extracts values specific to line items, and correlates them to the given columns.

7. Click OK.

The server behavior is applied to the page.

Set Session Value

Many Web applications that require values to persist over multiple pages make extensive use of session variables.

eCart works with either session variables or cookies; many developers prefer to use session variables to avoid storing sensitive information on the client's machine.

To facilitate the use of session variables, eCart includes the Set Session Value server behavior.

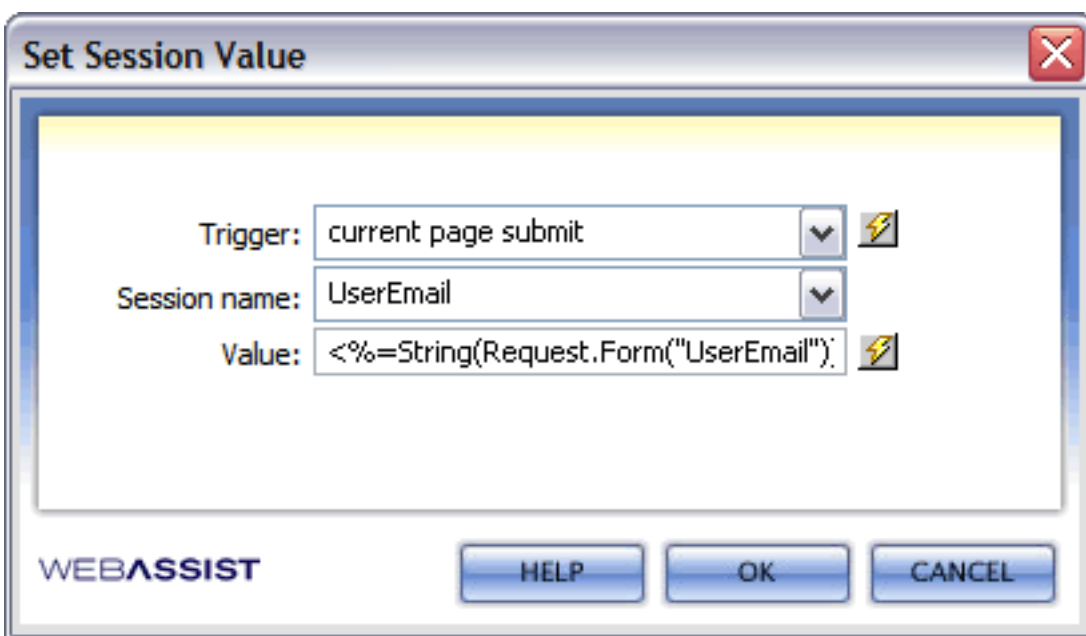
Note: Although not commonly known, session variables require the user enable cookies. When a session is first initiated, the session ID is stored on the client machine in a cookie.

The Set Session Value server behavior sets any existing or newly defined session variable to a dynamic or static value. Like the other data related server behaviors covered in this section, the server behavior may be triggered by a number of events.

The triggering actions include the press of any form button on the page — in general or a specific button — or other actions including the loading of the page or any form posting to the current page; you can even execute the server behavior if a particular session variable or other dynamic value is defined or if the referenced session variable is not defined.

To set a session variable value:

1. From the Server Behaviors panel, choose Add (+) and select eCart > General > Set Session Value server behavior.



2. Select the action you want to initiate the server behavior from the Trigger list. The options are:
 - **If Session Variable Is Not Defined:** Defines the session variable listed in the Session Name field and sets the value.
 - **Any Form Post:** Any form posted to the current page sets the session value.
 - **Before Page Load:** The specified session variable is set to the requested value when the current page is loaded but before it is rendered.
 - **Current Form Submit:** Any form submitted on the current page activates the server behavior.
 - **eCart [cart name] is not empty:** The server behavior is triggered if the specified shopping cart applied to the page contains at least one item.
 - **eCart [cart name] is empty:** The server behavior is triggered if the specified shopping cart applied to the page contains no items.
 - **Button [button name] pressed:** The server behavior is triggered if the specified button on the page is clicked.
 - **Recordset [recordset name] is not empty:** The server behavior is triggered if the specified recordset applied to the page contains at least one item.
 - **Recordset [recordset name] is empty:** The server behavior is triggered if the specified recordset applied to the page shopping cart contains no items.
 - **Dynamic Value Defined:** Click the associated lightning bolt and choose any dynamic value. If the selected value has been defined, the session variable entered is set to the requested value.
3. Select the session variable you want to affect from the Session Name list; if the session variable is not available enter its name in the field.
4. Enter the desired value in the Value field. To apply a dynamic value, click the lightning bolt icon and select the value from the dynamic data dialog box; click OK to close that dialog when you're done.
5. Click OK to close the Set Session Value dialog box and apply the server behavior.

You can modify the server behavior by double-clicking its entry in the Server Behaviors panel.

Prior version support

Some features available in versions of eCart prior to the 3.0 release are no longer available in the 3.0 and greater, having been replaced by improved features.

Existing applications of these features are recognized as server behaviors in eCart 3.0+ , making removal possible through the server behaviors panel, but insertion of these components is no longer supported.

The following is a list of features from eCart 1.x that have been replaced. Each links to the new feature that has replaced and improved upon its function:

- **PayPal Quick Checkout**
Replaced by advanced checkout configuration available in the Insert Checkout Form feature, which directly supports PayPal configuration , including IPN.
- **WorldPay Quick Checkout**
Replaced by advanced checkout configuration available in the Insert Checkout Form feature, which directly supports WorldPay configuration.
- **Store Contents in Recordset server behavior**
Replaced by two new server behaviors, [Store Cart Summary in Database](#), and [Store Cart Details in Database](#). These two server behaviors work in conjunction to insert shopping cart contents into the database without using a recordset, improving the data storage methodology used by eCart.
- **eCart Looper**
Replaced by the [Repeat Cart Region server behavior](#)

Upgrading eCart 1.x features to 3.0+

Users who are upgrading to eCart 3.0 from 1.x only need to upgrade the eCart Object to the current version to begin taking advantage of the features available within eCart 3.0. This is accomplished by:

1. Installing the 3.0+ release
2. Open a page developed with 1.x within your site
3. In the Server Behaviors panel, double click the existing application of the eCart Object
4. Click OK.

This will update the eCart Object definition file for your entire site. You can then use the new version to edit existing features developed using 1.x, as well as implement new functionality available in the extension.

Run-time structure

The following is an overview of the structure of the components involved in implementing a shopping cart into your site.

This section details the following structural properties which need to be taken into consideration when configuring your shopping cart:

- [Site structure](#) - Discusses how shopping cart functionality fits within the scope of your site. Discusses shopping cart components available and their relative location within your pages.
 - [Database structure](#) - Describes the basic components needed to use a database in conjunction with your shopping cart.
-

Site structure

eCart Object definition files and the global definition file are stored in the **WA_eCart** folder off of the site root.

WebAssist default button images are stored in a further sub-directory **WA_eCart/Images** off of the site root. Likewise, CSS files used in conjunction with the shopping cart display are stored in the WA_eCart/CSS folder.

Note: If images are moved from the images directory, Dreamweaver should maintain the image paths to the new directory.

There are two include files necessary for each eCart integration. These files contain the code that is the heart of the eCart object. The first is WA_eCart_Definition. This file contains code universal to all eCart objects in the site and does not require editing of any kind. A different version is written for each server model. For example, the definition file for ASP-JavaScript is named WA_eCart_Definition_JS.asp.

Each eCart Object defined for the site also has it's own specific include file. These include files follow the convention: *ObjectName_Server model designation* (JS: JavaScript, VB: VBScript, CF: ColdFusion, and PHP: PHP).

Example:

An ASP JavaScript eCart Object, WA_eCart1, would have an include file named:

WA_eCart1_JS.asp.

This is the object definition file and it contains all of the information specific to each defined eCart Object including but not limited to the following:

- [Cookie persistence](#) configuration
- [Cart column](#) and [calculation](#) definitions
- [Merchandising Rule](#) information

Removing eCart definition files from this directory effectively removes them from the site. This means that the files will no longer appear in the list of available [eCart™ Objects](#) in the [eCart floater](#).

Database structure

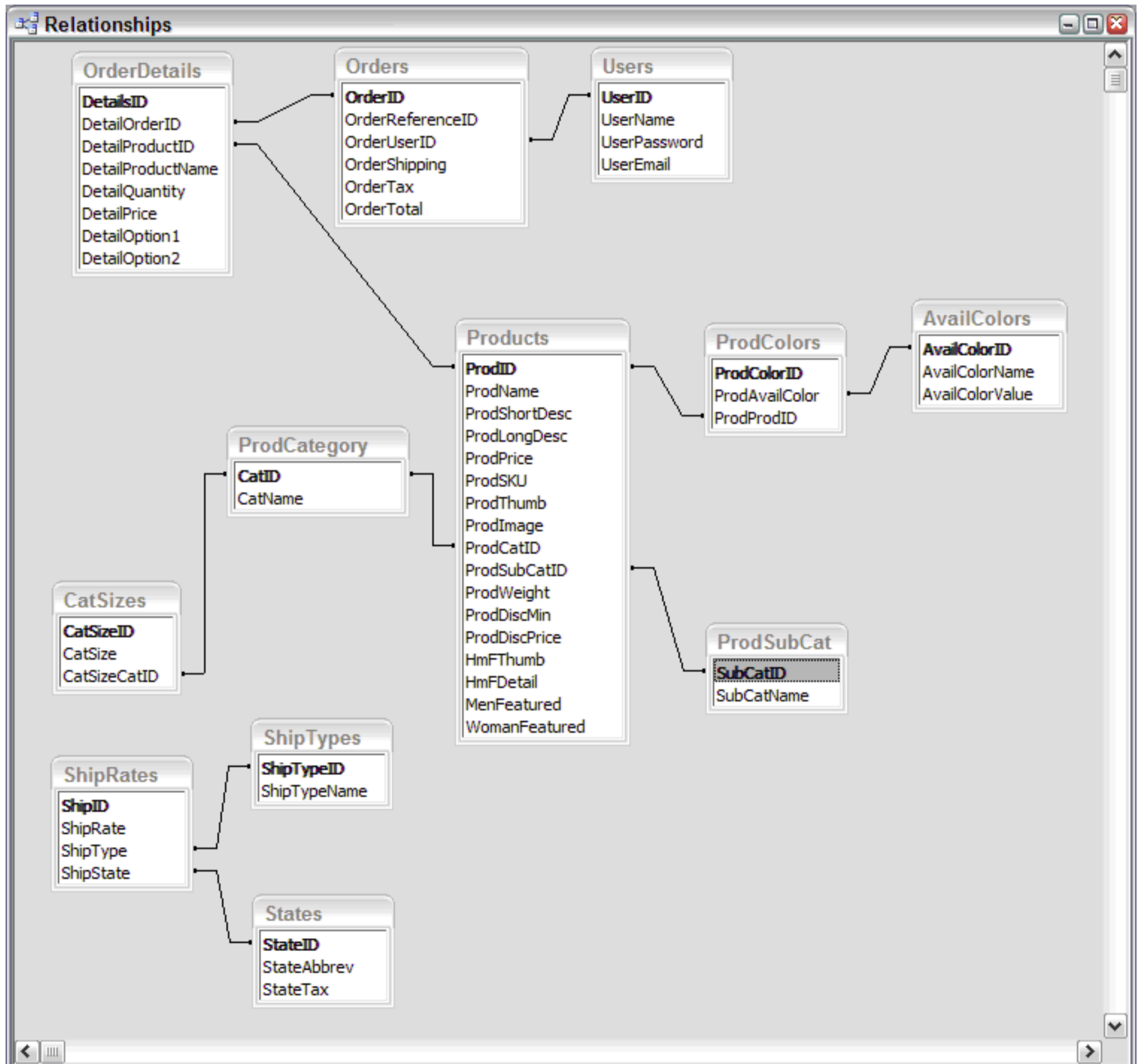
eCart is built to be functional over a variety of different database types. The structure for your database depends on the type of information you wish to store for use with your shopping cart application.

For your convenience, the following outlines a common database structure used in the sample store referenced in the Tutorials:

The Blue Sky Footwear online store is powered by a database with 12 tables:

- Orders - The order summary table
- OrderDetails - Holds a record for each item on each record
- Users - Registered users
- Products - Contains product information for building catalog pages
- ProdCategory - Divides products into two basic categories, mens and womens
- ProdSubCat - Lists all the different styles for shoes available
- CatSizes - Details sizes for various products organized by category (men or women)
- CatSizes - Details sizes for various products organized by category (men or women)
- ProdColors - Shows which colors are available for specific products
- AvailColors - Contains a list of colors to use in the ProdColors table
- ShipRates - Holds the various shipping rates used
- ShipTypes - Lists the shipping options, such as overnight, standard, etc.
- States - Lists the states where tax must be charged and their state tax rate

The following diagram shows the relationships between the various tables; bold column names represent key fields.



Release Notes

The following items detail issues to be aware of when using eCart 2.1:

- 1. Uninstalling eCart through the Extension Manager:** Please note that uninstalling takes a significant amount of time. Please be patient and allow 15-20 minutes depending on the resources for your computer.
 - 2. Display Manager/Checkout Wizard error: The Application WebtoPNG has quit unexpectedly**
This issue occurs on the Macintosh and is related to issues with the preview engine displaying a selected configuration. Restarting the Display Manager or Checkout Wizard should address the issue.
 - 3. Security Warning when opening sample MS Access database**
If attempting to open the sample database in Access 2003, the following Security Warning is alerted:
"This file may not be safe if it contains code that was intended to harm your computer. Do you want to open this file or cancel the operation?"
This is a standard warning and you should not be concerned when opening the provided sample.
 - 4. Using Live Data to preview pages**
Pages that have eCart object include files referenced on the page will not work with Dreamweaver Live Data. This is the case for ASP-JavaScript and ASP-VBScript server models. WebAssist has identified the problem and is working on a fix - but the fix is not included in the release version of eCart 2.1. If you would like a beta version of eCart that includes this fix please contact sales@webassist.com.
 - 5. Dreamweaver Error: - Scripts/ssClasses.js file - nodeToOffsets: Argument number 1 is invalid**
This error occurs when using eCart functionality and switching between pages. If you have applied code to a page using an eCart feature and then switch to a different document to edit, and if the first edit you make to the new document is to attempt to add eCart functionality, this error may arise. The workaround is to simply modify the new page by adding some text prior to attempting to implement any eCart functionality for the feature to be reset to work appropriately.
 - 6. Design View error for shopping cart displays**
If you have used the Display Manager to insert a shopping cart display numerous times, you may encounter issues with the rendering of a shopping cart display in Design View. Buttons within the display may appear as broken image references. This is due to memory limitations within Dreamweaver that prevent Design View from rendering the page correctly, but does not reflect issues with the code placed on the page. If you have this issue, save your page and close and restart Dreamweaver. When you re-open the page upon restarting, you should see it render correctly.
 - 7. User Authentication in PHP sample store**
PHP pages that include any of the Server Behaviors for User Authentication and eCart will generate a run-time error by default. To fix this problem - move the User Authentication code chunks above the eCart code chunks. This will remove the run-time error.
 - 8. PHP runtime error: Warning: session_start(): Cannot send session cookie**
This error arises when you have placed multiple eCart objects on a page, and then removed one of them. When the object is removed, the include file reference is removed from the source code, but leaves a blank line, which causes this error. To fix this problem, ensure that any blank lines in the location of the removed include file are removed as well.
 - 9. ColdFusion runtime error: Cannot lock session scope**
This error arises when your ColdFusion site is not configured to use session variables. ColdFusion sites that use eCart have the following two requirements:
 - o a file named *Application.cfm* must be included at the root of your site and contain the following code:

```
<cfapplication name="eCart" sessionmanagement="Yes">
```
 - o Your ColdFusion Server must have session variables enabled. Log in to your ColdFusion Server Administrator and under *Server Settings*, click the *Memory Variables* link to navigate to that configuration section. Make sure that **Enable Session Variables** is checked, and click the *Submit Changes* button to update your settings. If you do not administrate your server or have it hosted elsewhere, please contact your hosting provider to confirm these settings.
 - o **11. PHP runtime error when Shipping bindings are applied to a page.**
This error arises when you add a Shipping binding to a PHP page where the session has not been set. For this to work correctly, the following code must be placed in the head of the page to set the session:

```
<?php if (!session_id()) session_start(); ?>
```
 - 10. ColdFusion data binding issues for currency**
ColdFusion does not properly format currency with the raw output for data-binding. The run-time code will display a currency as \$195.0000. To fix this problem select Format from the Bindings panel for ProdPrice and set it to Dollar Format. This will format the value correctly and add a dollar sign to the data-binding value. You will need to remove the dollar sign from the code snippet that you place on the page for the eCart tutorial.
 - 11. Coldfusion: Upgrading shopping cart displays from 1.x**
The looper feature within a shopping cart display from version 1.x does not inspect correctly within the 2.0 version of eCart and is represented as a <cfif> statement within the display. For this reason, you cannot remove the looper feature from an older display using the Server Behavior panel, and would need to manually remove this from your source code. For this reason, we recommend removing and creating a new shopping cart display using the new Display Manager if you intend to make modifications to your shopping cart display.
 - 12. ColdFusion: Issues using dynamic lists with eCart**
ColdFusion has problems including dynamically populated option lists in eCart; to be used effectively, option lists must be hand-coded. In eCommerce Recipes, WebAssist has provided a series of code snippets that can be inserted onto the page. ColdFusion developers should use these snippets as a starting point when including option lists in their own pages.
 - 13. Authorize.Net checkout for PHP requires that mhash is compiled for your server**
Authorize.Net requires that *mhash* support is installed on your PHP server to properly encode transaction information for processing on their gateway. *Mhash* is a free (under GNU Lesser GPL) library which provides a uniform interface to a large number of hash algorithms. Contact your server administrator or ISP directly to determine if your server supports *mhash*.

The following location contains more information regarding *mhash* and PHP.
<http://us4.php.net/mhash>

Contact your server administrator or ISP directly to determine if your server supports *mhash*.
 - 14. Authorize.Net checkout is not supported for ColdFusion**
Authorize.net uses *mhash* to encode transactions submitted to their server, and the necessary algorithm they provide is currently not supported for ColdFusion servers.
 - 15. PayPal - issues with using the Line Items checkout configuration**
PayPal supports passing line items for checkout, but does not support passing negative line items, such as discounts calculated by eCart. Consequently, the provided checkout configuration for line items contains specialized code that passes charges as an additional line item in the shopping cart.

You may create a customized PayPal checkout configuration that passes charges to PayPal through their API's provided shipping variable if you wish, but any other charge rules passed to PayPal will be passed as line items. Passing the total of all charges in the shipping variable is potentially confusing to your customers, as all charges will be labeled as shipping charges when handled by PayPal's checkout form on their server.

In addition, you may use the Get Transaction Details server behavior to retrieve line item information from PayPal. If you pass charges as line items to PayPal and are using IPN, charges will be returned as line items within the shopping cart. This may present an issue if you try to re-populate the shopping cart with these line items, as your merchandising rules are still active for the eCart Object used in your pre-checkout configuration. This will cause the charges to be added as an item in the cart, as well as a charge applied from the rule configuration, effectively doubling your charges.

To get around this issue, you should create a new eCart Object specific to the page the Get Transaction Details server behavior is applied to, or any page that works with the shopping cart contents passed back through IPN. This configuration should be identical to your original configuration, except no rules should be applied to it, as those charges are now handled as line items in the shopping cart. This will prevent the charges from being applied to the cart twice.
 - 16. WorldPay - issues with using the Line Items checkout configuration**
WorldPay supports passing line items using their Select Checkout gateway option, but does not support passing negative line items, such as discounts calculated by eCart. Consequently, the provided checkout configuration for line items contains specialized code that passes charges as an additional line item in the shopping cart.

Currently, the WorldPay configuration provided for the Get Transaction Details server behavior is not configured to retrieve line item details in the callback available from the WorldPay gateway. Support for this will be added in an upcoming release, but advanced developers can attempt to add the appropriate configuration information to the existing WorldPay configuration available.
 - 17. WorldPay - Get Transaction Details is not supported for Coldfusion**
Support will be added in a future release.
-