**The Shopping Cart Section**

Your shopping cart section of the storefront displays the shopping cart information for the customer. This provides a real-time update for all of the items the customer has selected for purchase. The customer knows at any time what items are in the shopping cart just by glancing at the shopping cart section, which appears on all Web pages in the storefront.

The key to the shopping cart is the PHP *session cookie*. Each time a customer accesses your Web site, he or she creates a *session*. The Apache server assigns each session a unique session ID that it tracks on the server. PHP allows you to store data within the session, called a *session cookie*. When you use the *session\_start()*function at the beginning of your Web session, PHP allows you to access data stored in the session cookies. The session cookie stores multiple data values as an array variable.

Session cookies only last for the duration of the browser session. By default, when a customer closes the browser window, the session cookie disappears.

**Storing Arrays in Session Cookies**

So far, you've seen that the session cookie is an associative array where you can store multiple values by defining a key and a value:

$\_SESSION['user'] = $userid;

You could use this technique to store the products the customer selects for purchase, but things would get tricky. You could use the product description as the associative key, but that could get messy, depending on the products sold in the store. A better solution is to use the product ID for the product selected as the key and store the quantity purchased as the value:

$\_SESSION[1] = 10;  
$\_SESSION[6] = 5;

This would work, but it would make life more difficult if you needed to store additional information, such as the customer name and address. If you added additional keys for customer information, you wouldn't be able to tell which cookies were products and which were customer information.

The simple solution to this challenge is to use *multidimensional arrays*. A multidimensional array is nothing more than an array variable where each element in the array contains an array variable. If it sounds confusing, it really isn't.

The multidimensional array allows you to create a session cookie value that contains an array of values. This way, you can create a single session cookie, called a *cart*. It contains an array consisting of the selected product IDs as keys and the quantities as values. Using this method, you can also create an additional session cookie, called *user*, to track the user ID of the customer session, and you can easily tell which session cookie contains the shopping cart data.

To create an array within a session cookie, just use the array() function and start assigning keys and values. It'll look something like this:

$\_SESSION['cart'] = array();  
$\_SESSION['cart'][1] = 10;  
$\_SESSION['cart'][6] = 5;

Notice that the multidimensional array adds the second array keys onto the session cookie array key (cart).

You'll use a couple of functions when working with the arrays. When you want to count how many items are in the shopping cart, use the following code:

$items = count($\_SESSION['cart']);

This returns the number of keys in the multidimensional array within the *cart* array. Just like a regular array, if you want to iterate through all of the product keys, you can use the *foreach()* function:

foreach( $\_SESSION['cart'] as $prodid => $quantity)

{

echo "You purchased $quantity of product ID: $prodid

}

To add more of a product that's already in the cart, you increase the quantity value:

$\_SESSION['cart'][1] = $\_SESSION['cart'][1] + 3;

The same obviously applies if you want to remove some of the product quantity from the shopping cart. If you want to remove a product completely from the shopping cart, you can use the unset() function:

unset($\_SESSION['cart'][1]);

Armed with this new knowledge of session cookies, you can create the cart.inc.php program. This program will create the shopping cart variable if it doesn't exist and then display the current products stored in the shopping cart. Just follow these steps:

1. Create the file *cart.inc.php* in the store folder.
2. Open the file and enter the following code:

<?php

echo "<h2>Your shopping cart:</h2>\n";

if (!isset($\_SESSION['cart']))

{

$\_SESSION['cart'] = array();

echo "is empty\n";

} else

{

$items = count($\_SESSION['cart']);

if ($items == 0)

{

echo "is empty\n";

} else

{

$total = 0;

echo "<table width=\"100%\" cellpadding=\"1\" border=\"1\">\n";

echo "<tr><td>Product</td><td>Quantity</td><td>Total</td></tr>\n";

foreach($\_SESSION['cart'] as $prodid => $quantity)

{

$query = "SELECT description, price FROM products WHERE prodid = $prodid";

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$description = $row['description'];

$price = $row['price'];

$subtotal = $price \* $quantity;

$total += $subtotal;

printf("<tr><td>%s</td><td>%s</td><td>$%.2lf</td></tr>\n", $description, $quantity, $subtotal);

}

printf("<tr><td colspan=\"2\">Total</td><td>$%.2lf</td></tr>\n", $total);

echo "</table>\n";

}

}

?>

1. Save the file and exit the editor.

The first section uses the standard *isset()* function to determine if the shopping cart already exists. If it doesn't (the first time the customer connects to the Web page), it creates it in the session cookie. If the session cookie exists, the code uses the count() function to see if there are any products in the shopping cart. If so, it's just a matter of using the *foreach()* function to iterate through each value and displaying its information.

The cart.inc.php code makes use of the *printf()* function to ensure that the price values stay in the proper format.

We covered a lot of ground in the project today. Let's go to Chapter 5 and review all you did.