**Adding Products**

Now that your store has a category to hold products, it's time to create some products. The code you'll use to add a new product to the store also uses a standard HTML form. However, there are a couple of twists to this. Let's create the code file first, and then we'll take a closer look at it so I can show you what I mean.

1. Create a file called *newproduct.inc.php* in the *admin* folder of your store application.
2. Enter the following code into the file:

<?php

if (!isset($\_SESSION['store\_admin']))

{

echo "<h2>Sorry, you have not logged into the system</h2>\n";

echo "<a href=\"admin.php\">Please login</a>\n";

} else

{

$userid = $\_SESSION['store\_admin'];

echo "<form enctype=\"multipart/form-data\" action=\"admin.php\" method=\"post\">\n";

echo "<h2>Enter the new product</h2><br>\n";

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

echo "<tr><td>Category</td>\n";

echo "<td><select name=\"cat\">\n";

$query="SELECT catid,name from categories";

$result=mysql\_query($query);

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

{

$catid = $row['catid'];

$name = $row['name'];

echo "<option value=\"$catid\">$name</option>\n";

}

echo "</select></td></tr>\n";

echo "<tr><td>Description</td><td><input type=\"text\" size=\"40\" name=\"description\"></td></tr>\n";

echo "<tr><td>Price</td><td><input type=\"text\" size=\"10\" name=\"price\"></td></tr>\n";

echo "<tr><td>Quantity in stock</td><td><input type=\"text\" size=\"10\" name=\"quantity\"></td</tr>\n";

echo "<input type=\"hidden\" name=\"MAX\_FILE\_SIZE\" value=\"1024000\">\n";

echo "<tr><td>Picture</td><td><input type=\"file\" name=\"picture\"></td></tr>\n";

echo "</table>\n";

echo "<input type=\"submit\" value=\"Submit\">\n";

echo "<input type=\"hidden\" name=\"content\" value=\"addproduct\">\n";

echo "</form>\n";

}

?>

1. Save the file and exit the editor.

You should recognize most of this code. Just like when you added a category, you need to check if the session cookie is set, so you must use PHP for the entire file. The HTML form uses a table to lay out each data label and data entry item for the manager to enter new data values. To create the HTML form, you need to use two features you might not be familiar with yet.

The first feature is how you allow the manager to select which category the product belongs. The code uses a standard HTML *select* form item to provide a list of the available categories. The select item displays a drop-down box containing a list of items. The manager can then select one item from the list, which is what the form returns to the action file.

After declaring the <select> tag, each item is added using the <option> HTML tag. Just use the standard mysql\_fetch\_array() function to extract all of the category data records and create the individual <option> tags. This allows you to push as many items into the list as you need to. When it's done, you need to close out the select section using the </select> tag.

To get the values for the drop-down box, you must extract both the individual category names and catid values from the categories table and then populate the drop-down box object with that information. The category name appears in the drop-down box. When the manager selects an item, the catid value goes to the receiving program (this is the value that's stored in the product table anyway).

The second feature you may not be familiar with is the *file* input type.

echo "<input type=\"hidden\" name=\"MAX\_FILE\_SIZE\" value=\"1024000\">\n";

echo "<tr><td>Picture</td><td><input type=\"file\" name=\"picture\"></td></tr>\n";

This neat little feature in HTML shows a textbox where the user can either enter a filename or click a *Browse* button. When a Web site visitor clicks the button, it produces the graphical file system explorer for the PC and allows the user to browse and select the file from his or her file system. When the visitor selects a file, the filename automatically appears in the textbox.

The file input type data entry

The *MAX\_FILE\_SIZE* hidden value allows you to set a maximum size for files the manager can upload. When the manager finishes filling out the new product form and clicks the Submit button, the form action sends the data to the *addproducts.inc.php* file. The file input type uploads the designated file to the Web server.

Now you need to extract the file and process the image, along with the rest of the data, to store in the products table.

**The addproducts Code**

Let's create the addproduct.inc.php file first and then walk through what it does.

1. Create a file called addproduct.inc.php in the admin folder in your store application area.
2. Enter the following code into the file:

<?php

$catid=$\_POST['cat'];

$description=$\_POST['description'];

$price=$\_POST['price'];

$quantity=$\_POST['quantity'];

if (get\_magic\_quotes\_gpc())

{

$catid = stripslashes($catid);

$description = stripslashes($description);

$price = stripslashes($price);

$quantity = stripslashes($quantity);

}

$catid = mysql\_real\_escape\_string($catid);

$description = mysql\_real\_escape\_string($description);

$price = mysql\_real\_escape\_string($price);

$quantity = mysql\_real\_escape\_string($quantity);

$thumbnail = getThumb($\_FILES['picture']);

$thumbnail = mysql\_real\_escape\_string($thumbnail);

$query = "INSERT INTO products (catid, description, picture, price, quantity) " .

" VALUES ('$catid','$description','$thumbnail', '$price', '$quantity')";

$result = mysql\_query($query) or die('Unable to add product');

if ($result)

echo "<h2>New product added</h2>\n";

else

echo "<h2>Problem adding new product</h2>\n";

?>

The start of the code extracts most of the form fields using the standard PHP $\_POST[] array variables, then it uses our new friends get\_magic\_quotes\_gpc(), stripslashes(), and mysql\_real\_escape\_string() to ensure the data is in the proper format for the SQL INSERT statement.

Next, you need to handle the uploaded image file. Follow me to Chapter 4, and you'll see how to do that.