OSCommerce Basic Modification Guide

By Michael Sasek

OSCdox

Mod Guide Chapter 1 General Concepts

The way osCommerce is constructed looks a bit complicated at first, but once you learn its conventions, you can efficiently modify your cart quite extensively. First, if you look at index.php in a text editor, you will see that the code is divided into sections. These sections are defined like this:

```
<!-- header //-->
<?php require(DIR_WS_INCLUDES . 'header.php'); ?>
<!-- header_eof //-->
```

The code

```
<!—sometext//-->
```

defines the start of a specific section, and the code

```
<!—sometext_eof//-->
```

defines the end of that section. This is important for those of us that do not read php or html code very well.

Example (from index.php):

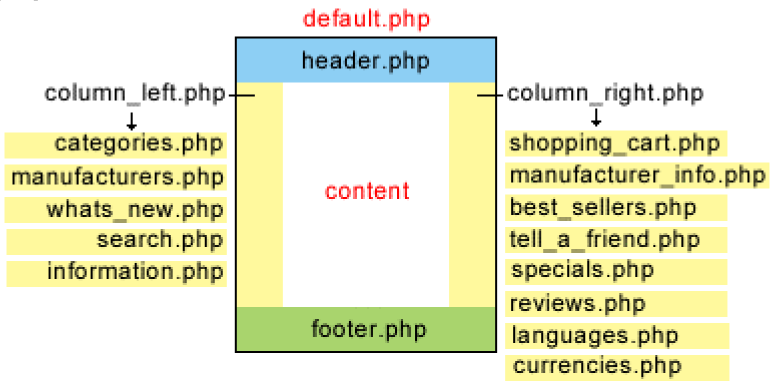
```
<body marginwidth="0" marginheight="0">
<!-- header //-->
<?php require(DIR_WS_INCLUDES . 'header.php'); ?>
<!-- header_eof //-->
<!-- body //-->
<table border="0" width="100%" cellspacing="3" cellpadding="3">
 <tr>
   <td width=" <?php echo BOX_WIDTH; ?>" valign="top">
        <table border="0" width=" <?php echo BOX_WIDTH; ?>" cellspacing="0" cellpadding="2">
<!-- left_navigation //-->
<?php require(DIR_WS_INCLUDES . 'column_left.php'); ?>
<!-- left_navigation_eof //-->
```

Above, php code is separated from the html with **<?php** at the beginning of the code, and **?>** at the end. Knowing this, you can safely edit the html without breaking any of the core PHP code.

In the above example, you can see the conventions in action. First, you see a standard html **<body>** tag followed by the section divider **<!—header//-->**. The **<?php** and **?>** tags offset the php command/code that is between them, in this case, telling the server to require header.php before parsing any more of the file default.php. And finally, the closing of the section using **<!—header_eof//-->** . Note how this is repeated in the subsequent sections of code, body and left navigation. Understanding the above convention is critical to editing the look of osCommerce, and it is used in every file in osCommerce.

Mod Guide Chapter 2 File Schema

As a would-be Oscommerce designer, you need to become familiar with the files that actually set how osCommerce looks. The way osCommerce dynamically generates pages allows a consistent editing process. You modify the same code in a similar location in all the necessary files. These are the files you will need to become familiar with in order to customize the look of your shopping cart.
This handy graphic shows you what files make up all the included parts in the **index.php** :

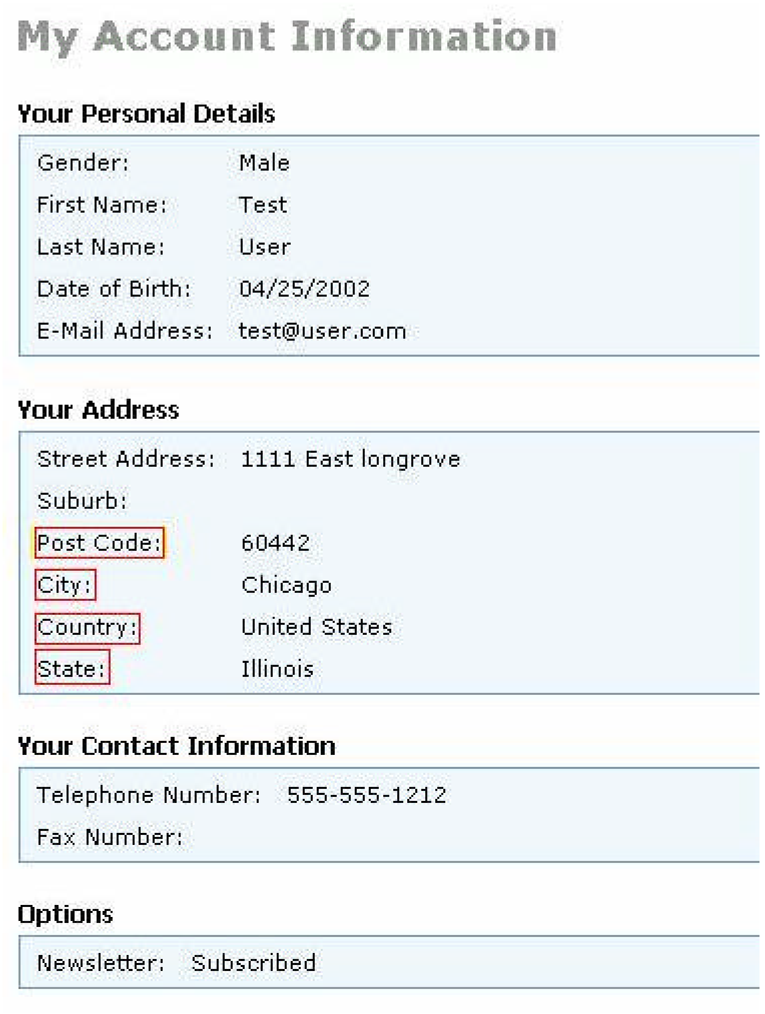


Mod Guide Chapter 3 Practice Exercises

These exercises are more designed to teach you the basics of osCommerce while allowing you to make meaningful modifications to the software.

Exercise 1 Learning to Edit osCommerce

Changing Field order in account_edit.php
The code between the <?php ?> tags, including the actual tags, can be treated as discrete code modules, and can be moved around within html tags and to some extent, within the script itself. A simple example of this is changing the order of fields in the file account_edit.php. See the screen captures and code snippets below:



My Account Information

Your Personal Details

Gender:	Male
First Name:	Test
Last Name:	User
Date of Birth:	04/25/2002
E-Mail Address:	test@user.com

Your Address

Street Address:	1111 East longrove
Suburb:	
Post Code:	60442
City:	Chicago
Country:	United States
State:	Illinois

Your Contact Information

Telephone Number:	555-555-1212
Fax Number:	

Options

Newsletter:	Subscribed

The project is to change the order of the items highlighted by the red boxes in the above image. The default order is rather haphazard, and not at all intuitive. I am going to put them in a more user friendly order, i.e. City, State, Post Code, and Country.

To do this, I need to open **account_edit.php**, and look for the section of the file that relates to the above output. I know this will be in the **<!—body_text//-->** section by convention, so now I have to find it!

Ok, I can’t find it! But I did find this bit of code:

```
require(DIR_WS_MODULES . 'account_details.php')
```

This points us in the right direction. Whenever you see php code that starts with **require**, look for the directory name and filename in the parentheses, in this case, **DIR_WS_MODULES** and **account_details.php**. This tells me I have to look in the

/modules directory, as specified in the file configure.php, to find the file account_details.php. Great!

After opening **account_details.php** in a text editor, look for anything that will put you in the correct vicinity. In this case, look for a variable or text that refers to city or state, these are the items that will be moved. I often use the *Find* command in my text editor (CTRL-F), so I don't have to look too hard. Using this method, zero in on the code blocks that build the form output for city and state.

```
 <tr>
      <td class="main">&nbsp;<?php echo ENTRY_CITY; ?></td>
           <td class="main">&nbsp;
<?php
  if ($is_read_only) {
    echo $account['entry_city'];
  } elseif ($error) {
    if ($entry_city_error) {
      echo tep_draw_input_field('city') . '&nbsp;' . ENTRY_CITY_ERROR;
    } else {
      echo $HTTP_POST_VARS['city'] . tep_draw_hidden_field('city');
    }
  } else {
    echo tep_draw_input_field('city', $account['entry_city']) . '&nbsp;' . ENTRY_CITY_TEXT;
 }
?> </td>
         </tr>
```

The PHP is in italics. The HTML is not. Notice how this creates a nice modular block of code. This block actually controls the output for where the City field will be displayed.

Exercise 2 Setting the Width of osCommerce

This section illustrates how to change the width of your shopping cart as it displays in a browser. This is pretty easy to do. Open **index.php** in Dreamweaver or a good text editor. In Dreamweaver, you will see the code view and the design view, which is very handy for changing the table structure on the fly. You will be able to see the changes in a wireframe type view instantly. Using a text editor, you have to understand the code a little more clearly, but either will get the job done.

You are looking for the html that controls the attributes of the master table.

Change this:

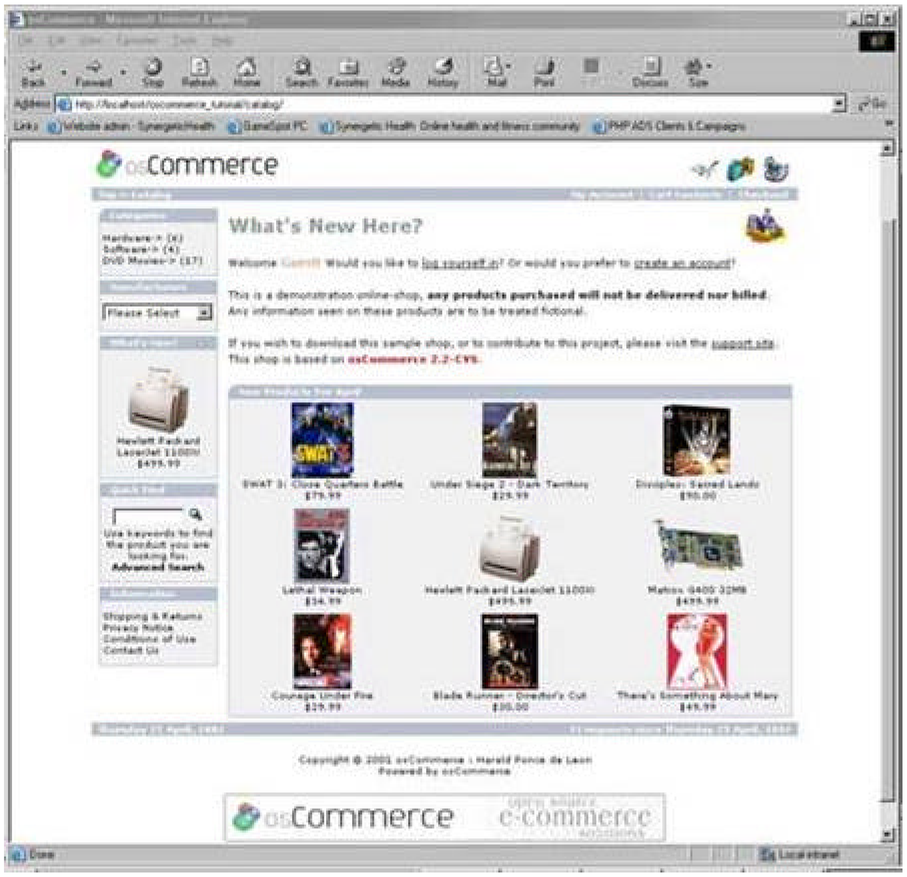
```
 <!-- body //-->
  <table border="0" width="100%" cellspacing="3" cellpadding="3">
  <tr>
    <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table border="0"
 width="<?php echo BOX_WIDTH; ?>" cellspacing="0" cellpadding="2">
```

To this:

```
 <!-- body //-->
 <table border="0" width="750" cellspacing="1" cellpadding="1" align=”center”>
  <tr>
    <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table border="0"
 width="<?php echo BOX_WIDTH; ?>" cellspacing="0" cellpadding="2">
```

The above change will set the width of your cart to 750 pixels and center the table on the page. It will also reduce the padding and spacing to 1 pixel. Step one finished.

Next you must change the same settings in **header.php** and **footer.php** in your **/catalog/includes** directory. See the screenshot that follows.



What's New Here?

Welcome Guest! Would you like to log yourself in? Or would you prefer to create an account?

This is a demonstration online-shop, **any products purchased will not be delivered nor billed**. Any information seen on these products are to be treated fictional.

If you wish to download this sample shop, or to contribute to this project, please visit the support site. This shop is based on osCommerce 2.2-CVS.

Hardware-> (6)
Software-> (4)
DVD Movies-> (17)
Please Select
Hewlett Packard LaserJet 1100Xi $499.99
Use keywords to find the product you are looking for.
Advanced Search
Shipping & Returns
Privacy Notice
Conditions of Use
Contact Us
SWAT 3: Close Quarters Battle $79.99
Under Siege 2 - Dark Territory $29.99
Disciples: Sacred Lands $90.00
Lethal Weapon $34.99
Hewlett Packard LaserJet 1100Xi $499.99
Matrox G400 32MB $499.99
Courage Under Fire $29.99
Blade Runner - Director's Cut $30.00
There's Something About Mary $49.99

Exercise 3 Boxes: Removing Boxes

Removing Boxes

The next example will show how to remove the right boxes from displaying in the shopping cart, and how to move existing boxes from the right side to the left. It will also demonstrate how to disable boxes that are not needed. Knowing how to do this will allow you to quickly change the look of the cart.

First, to stop a given box from displaying, you need to edit the php in one of two files, **column_left.php** or **column_right.php**, both found in the **/catalog/includes/** directory.

To stop the **reviews** box from displaying, simply remove the following line from **column_right.php**:

```
 require(DIR_WS_BOXES . 'reviews.php');
```

To stop the **languages** box and the **currencies** box from displaying on the right side of the page, remove the following from **column_right.php**. To move them to the left side, just cut and paste the code into **column_left.php** where you want it to show up:

```
  if (substr(basename($PHP_SELF), 0, 8) != 'checkout') {
    include(DIR_WS_BOXES . 'languages.php');
    include(DIR_WS_BOXES . 'currencies.php');
  }
```

To change the width of the actual side columns, find and edit the following line in **application_top.php**. Change 125 to whatever width you want to have in pixels:

```
 define('BOX_WIDTH', 125);
```

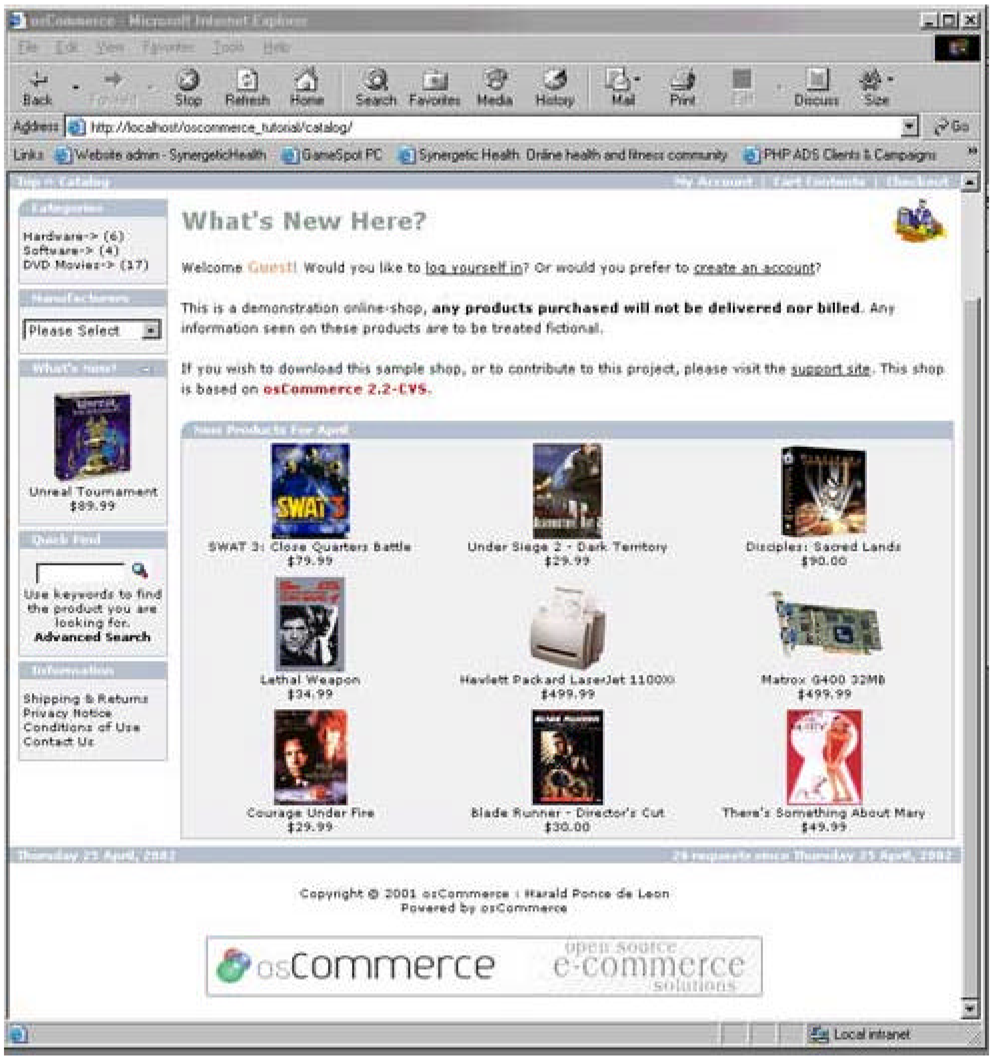
Now, to completely remove the right side column, we need to first edit the file **index.php**. Look for the section labeled

```
 <!-- body_text_eof //-->.
```

Directly below this tag, you will need to delete or comment out the following code:

```
   <td width="<?php echo BOX_WIDTH; ?>" valign="top"><table border="0"
 width="<?php echo BOX_WIDTH; ?>" cellspacing="0" cellpadding="2">
 <!-- right_navigation //-->
 <?php require(DIR_WS_INCLUDES . 'column_right.php'); ?>
 <!-- right_navigation_eof //-->
    </table></td>
```

Save **index.php** and visit your catalog in your browser. The right column is gone. See the following screenshot:


Back
Stop
Refresh
Home
Search
Favorites
Media
History
Mail
Print
Discuss
Size
Address
Links
Website admin - SynergeticHealth
GameSpot PC
Synergetic Health Online health and fitness community
PHP ADS Clients & Campaigns
Go
Local intranet

http://localhost/oscommerce_tutorial/catalog/

Top » Catalog | My Account | Cart Contents | Checkout

Categories
Hardware-> (6)
Software-> (4)
DVD Movies-> (17)

Manufacturers
Please Select

What's New?
Unreal Tournament
$89.99

Quick Find
Use keywords to find the product you are looking for.
Advanced Search

Information
Shipping & Returns
Privacy Notice
Conditions of Use
Contact Us

What's New Here?

Welcome Guest! Would you like to log yourself in? Or would you prefer to create an account?

This is a demonstration online-shop, **any products purchased will not be delivered nor billed**. Any information seen on these products are to be treated fictional.

If you wish to download this sample shop, or to contribute to this project, please visit the support site. This shop is based on **osCommerce 2.2-CVS**.

New Products For April

SWAT 3: Close Quarters Battle $79.99	Under Siege 2 - Dark Territory $29.99	Disciples: Sacred Lands $90.00
Lethal Weapon $34.99	Hewlett Packard LaserJet 1100Xi $499.99	Matrox G400 32MB $499.99
Courage Under Fire $29.99	Blade Runner - Director's Cut $30.00	There's Something About Mary $49.99

Thursday 25 April, 2002 | 28 requests since Thursday 25 April, 2002

Copyright © 2001 osCommerce : Harald Ponce de Leon
Powered by osCommerce

osCommerce open source e-commerce solutions

We are not finished yet. In order for the right column to be removed throughout the entire site, you have to remove the code in question from **EVERY** php file that displays a page. Search every file in your **/catalog** directory for

```
<!-- body_text_eof //-->
```

and comment out or delete the same code block we deleted from **index.php**. To make this easier, use a good multi-file text editor that will allow you to search all the files at once, and edit them quickly.

Exercise 4 Boxes: Adding New Boxes

Adding New Boxes, Links and Pages

We will now look into adding our own custom boxes, as well as adding/changing links, pages and graphics to boxes. In addition, we will learn how to add items to the columns, outside of the actual boxes. All of these modifications are fairly easy to perform, but they are not entirely intuitive (understatement!).

Making a New Box

Let’s jump right in. The files involved are: **/catalog/includes** :

1. column_left.php
2. column_right.php

/catalog/includes/boxes : **ALL** files in this directory

Open **/catalog/includes/boxes/information.php** in a text editor and save it as **/catalog/includes/boxes/test.php**.

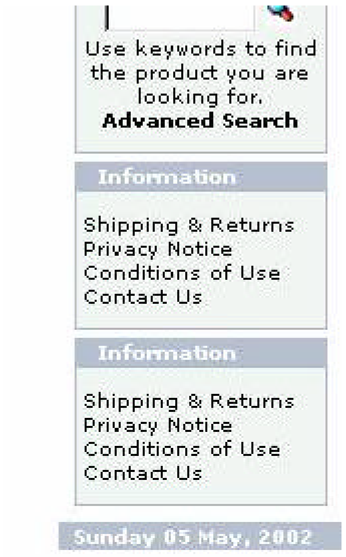
Then in column_left.php, add this line:

```
require(DIR_WS_BOXES . 'test.php');
```

directly below this line:

```
require(DIR_WS_BOXES . 'information.php');
```

Save **column_left.php** to your server, and reload the main catalog page in your browser. You will now see two **information** boxes on the left. The second one we just added with one line of code. That is the easy part.



Changing Links and Text : Customizing your new box and adding new pages

The next step is to customize that box, and to do it, we need to modify a few more files. I want to change the title bar of our new box, as well as make links to new, custom pages that I will also create. This process is a bit more clunky than it should be, but we will have to make due. Here we go!
For this example, I will be creating four links to pages called **testpage1.php, testpage2.php testpage3.php, and testpage4.php** in the new **information** block we created in the previous step.
I am using the original **shipping.php** files as my base template. Use this example to familiarize yourself with the procedure. The process is the same for other blocks, you just need to identify the proper files to copy and modify. Confused? Good. Read on…

Open the following files in WordPad or some other text editor that will not modify code without you telling it to, and will allow you to search and replace:
/catalog/includes/application_top.php
/catalog/includes/languages/english.php
/catalog/includes/languages/english/shipping.php
/catalog/shipping.php

/catalog/includes/boxes/test.php

In the file **/catalog/includes/application_top.php**, find the section marked **define filenames used in the project**. In this section, copy any one of the file definitions, and paste it to a new line, just after the one you copied. Now you need to modify the newly pasted line to point to **testpage1** See the example below:

Copy the first file definition listed:

```
define('FILENAME_ACCOUNT', 'account.php');
```

Then paste this on a new line immediately following it, four times. Create four new define statements as follows:

```
define('FILENAME_TESTPAGE1', 'testpage1.php');
define('FILENAME_TESTPAGE2', 'testpage2.php');
define('FILENAME_TESTPAGE3', 'testpage3.php');
define('FILENAME_TESTPAGE4', 'testpage4.php');
```

Now, save **/catalog/includes/application_top.php**. This is the step that creates the filename definitions so that osCommerce can build links.

Next, in the file **/catalog/includes/languages/english.php**, find the section marked **information box text**. Copy the entire section and paste it below the original section.

Change the section to look like this:

```
// information box text in includes/boxes/test.php
define('BOX_HEADING_TEST', 'Test Box');
define('BOX_TEST_LINK1', 'Test Link 1');
define('BOX_TEST_LINK2', 'Test Link 2');
define('BOX_TEST_LINK3', 'Test Link 3');
define('BOX_TEST_LINK4', 'Test Link 4');
```

Save **/catalog/includes/languages/english.php**. This step creates the link text that will go into each new link you create.

In the file **/catalog/includes/languages/english/shipping.php** edit the following:

```
define('NAVBAR_TITLE', 'Shipping & Returns');
define('HEADING_TITLE', 'Shipping & Returns');
define('TEXT_INFORMATION', 'Enter your shipping info here');
```

To look like this:

```
define('NAVBAR_TITLE', 'Test Page 1');
define('HEADING_TITLE', 'Test Page 1');
define('TEXT_INFORMATION', 'This is an added sample page');
```

Save as **/catalog/includes/languages/english/testpage1.php**

Repeat the above steps three more times, creating testpage2, testpage3, and testpage4. This is the step that actually creates the text that will be on each of your new pages, and in the process, creates four new files.

In the file: **/catalog/shipping.php** using the replace feature of you text editor

Replace this **FILENAME_SHIPPING**

With this **FILENAME_TESTPAGE1**

Save As **/catalog/testpage1.php**

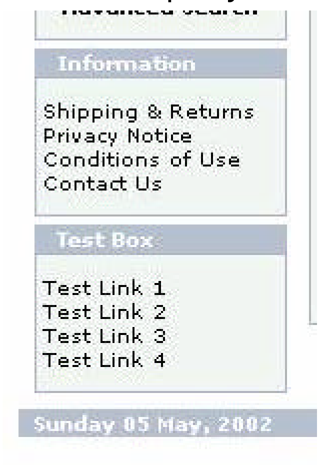
Repeat this three more times, changing **FILENAME_TESTPAGE1** to FILENAME_TESTPAGE2, FILENAME_TESTPAGE3 and FILENAME_TESTPAGE4 and saving as testpage2.php, testpage3.php and testpage4.php. This step creates the actual pages that will be loaded by the links.

Finally, edit the file **/catalog/includes/boxes/test.php** to look like this:

```
<?php
 $info_box_contents = array();
 $info_box_contents[] = array('align' => 'left',
                              'text'  => BOX_HEADING_TEST
                             );
 new infoBoxHeading($info_box_contents, false, false);
 $info_box_contents = array();
 $info_box_contents[] = array('align' => 'left',
                              'text'  => '<a href="' . tep_href_link
(FILENAME_TESTPAGE1, '', 'NONSSL') . '">' . BOX_TEST_LINK1 . '</a><br>' .
                                         '<a href="' . tep_href_link
(FILENAME_TESTPAGE2, '', 'NONSSL') . '">' . BOX_TEST_LINK2 . '</a><br>' .
                                         '<a href="' . tep_href_link
(FILENAME_TESTPAGE3, '', 'NONSSL') . '">' . BOX_TEST_LINK3 . '</a><br>' .
                                         '<a href="' . tep_href_link
```

(FILENAME_TESTPAGE4, '', 'NONSSL') . '">' . BOX_TEST_LINK4 . '</a>'

This changes the text that is output in the browser. You are finished editing files at this point. Make sure you upload the files to the proper directories, as some of them have the same filenames. View your catalog in your browser and the new links should show up in your new block! See the example below.



Exercise 5 Boxes: Changing the Look and Style

Changing the Look and Style

We have learned how to add and remove boxes, move them from side to side, and add and remove links and pages. Now we will delve into actually making them look different. This is where we get into modifying the font, graphics, colors and styles of the boxes.

The key files in these modifications are: **catalog/includes/classes/boxes.php** - This controls the actual building of the boxes. **catalog/stylesheet.css** – This is where you change or add styles to affect the boxes. **catalog/includes/boxes/** - all PHP files here are the actual boxes.

The next example will demonstrate what code you need to edit in boxes.php and stylesheet.css in order to remove or change the corner graphics, change the color, and add a top and bottom border as a box separator.

In **stylesheet.css**, create the style **infoBoxHeading** and make it have a top and bottom border with a width of 2px.

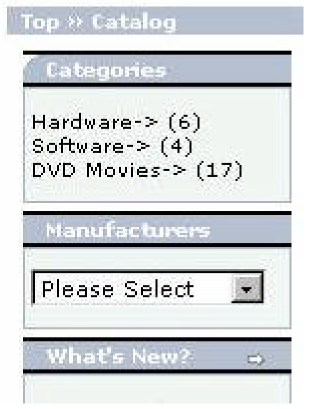
In **/catalog/includes/classes/boxes.php** find the following code (about line 97-100):

```
class infoBoxHeading extends tableBox {
   function infoBoxHeading($contents, $left_corner = false, $right_corner = false,
$right_arrow = false) {
     $this->table_cellpadding = '0';
```

And add this line below it:

```
$this->table_parameters = 'class="infoBoxHeading"';
```

This will create a class tag in the generated html code when the blocks are drawn which will change the heading to have the top and bottom border. See screenshot:



Next, to clean up the box headings, we want to remove the rounded and square corner graphics from the boxes completely. There are 2 ways to do this. One, remove the code that creates this or change the graphics to an transparent .gif. I find the easiest way is to load the transparent gif. To do this, simply find this code in boxes.php :

```
class infoBoxHeading extends tableBox {function infoBoxHeading($contents,
$left_corner = true, $right_corner = true, $right_arrow = false) {$this-
>table_cellpadding = '0';
```

```
      if ($left_corner) {
        $left_corner = tep_image(DIR_WS_IMAGES . 'infobox/corner_left.gif');
      } else {
        $left_corner = tep_image(DIR_WS_IMAGES . 'infobox/corner_right_left.gif');
      }
      if ($right_arrow) {
        $right_arrow = '<a href="' . $right_arrow . '">' . tep_image
 (DIR_WS_IMAGES . 'infobox/arrow_right.gif', ICON_ARROW_RIGHT) . '</a>';
      } else {
        $right_arrow = '';
      }
      if ($right_corner) {
        $right_corner = $right_arrow . tep_image
 (DIR_WS_IMAGES . 'infobox/corner_right.gif');
      } else {
        $right_corner = $right_arrow . tep_draw_separator
 ('pixel_trans.gif', '11', '14');
      }
```

and replace the highlighted paths with the filename ‘pixel_trans.gif’ It should look like the code that follows:

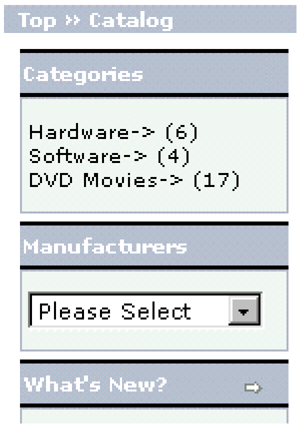
```
 class infoBoxHeading extends tableBox {
    function infoBoxHeading($contents, $left_corner = false, $right_corner = false,  $right_arrow =
false) {
      $this->table_cellpadding = '0';
      $this->table_parameters = 'class="infoBoxHeading"';
      if ($left_corner) {
        $left_corner = tep_image(DIR_WS_IMAGES . 'pixel_trans.gif');
      } else {
        $left_corner = tep_image(DIR_WS_IMAGES . 'pixel_trans.gif');
      }
      if ($right_arrow) {
        $right_arrow = '<a href="' . $right_arrow . '">' . tep_image
 (DIR_WS_IMAGES . 'infobox/arrow_right.gif', ICON_ARROW_RIGHT) . '</a>';
      } else {
        $right_arrow = '';
      }
      if ($right_corner) {
        $right_corner = $right_arrow . tep_image
 (DIR_WS_IMAGES . 'pixel_trans.gif');
      } else {
        $right_corner = $right_arrow . tep_draw_separator
 ('pixel_trans.gif', '11', '14');
      }
```

This removes the corner images completely. You could also specify your own images just as easily.

One final change is to increase the height of the box heading. This makes the headings on the boxes wider. To do this, find the following code in boxes.php :

```
$info_box_contents = array();
    $info_box_contents[] = array(array('align' => 'left', 'params' => 'height="14"
class="infoBoxHeading"', 'text' => $left_corner),
                                 array('align' => 'left', 'params' => 'width="100%"
height="14" class="infoBoxHeading"', 'text' => '<b>' . $contents[0]
'text' . '</b>'),
                           array('align' => 'left', 'params' => 'height="14"
class="infoBoxHeading"', 'text' => $right_corner));
    $this->tableBox($info_box_contents, true);
```

and change the height tag to whatever number you want. Higher numbers = wider headings. I am changing them to 20 for this example. See the screenshot below:



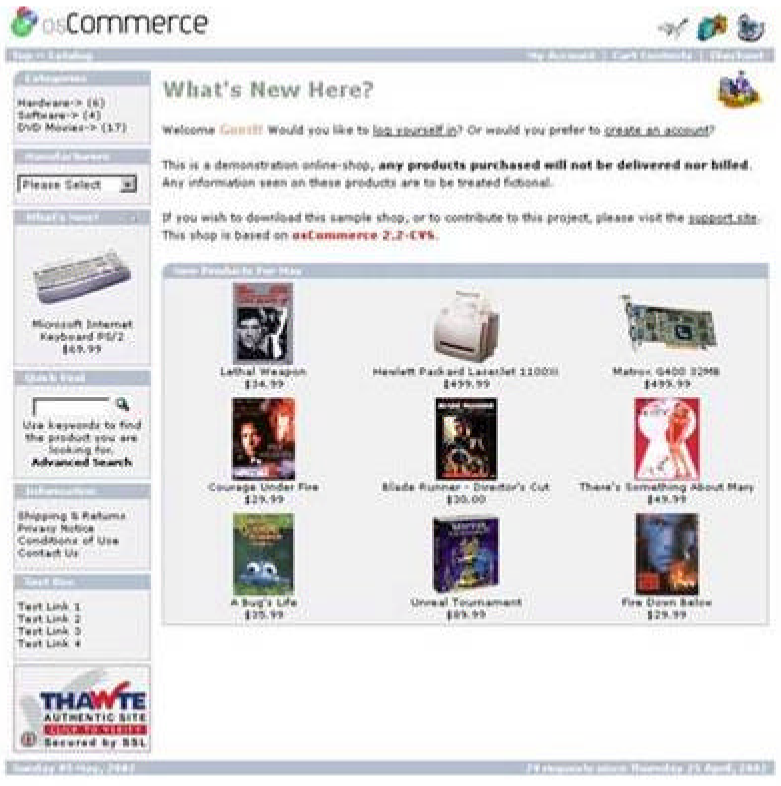
Exercise 6 Boxes: Adding Links outside of Boxes

Adding Links outside of Boxes

You may want to add links or html to the left or right column without enclosing it in a box, such as button advertising, logos or whatever else you can think up. This is another pretty easy edit, and only requires that you add a little bit of code to either **column_left.php** or **column_right.php** . Below is a sample of the code to add after the closing php command **?>**

```
 <tr>
   <td><img src="http://www.link-to-sampleimage.com"></td>
 </tr>
```

See the screenshot of all changes so far. Note the Thawte image at the bottom of the left column. It is outside the boxes all by itself:



What's New Here?

Welcome Guest! Would you like to log yourself in? Or would you prefer to create an account?

This is a demonstration online-shop, **any products purchased will not be delivered nor billed**. Any information seen on these products are to be treated fictional.

If you wish to download this sample shop, or to contribute to this project, please visit the support site. This shop is based on osCommerce 2.2-CVS.