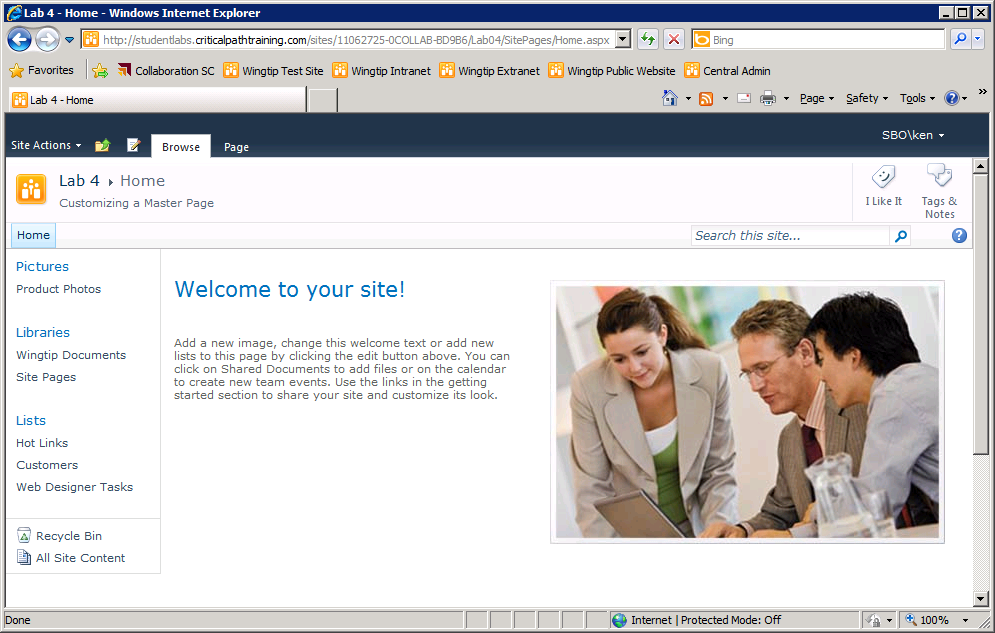
## Customizing Master Pages in SharePoint 2010

**Lab Time**: 45 minutes

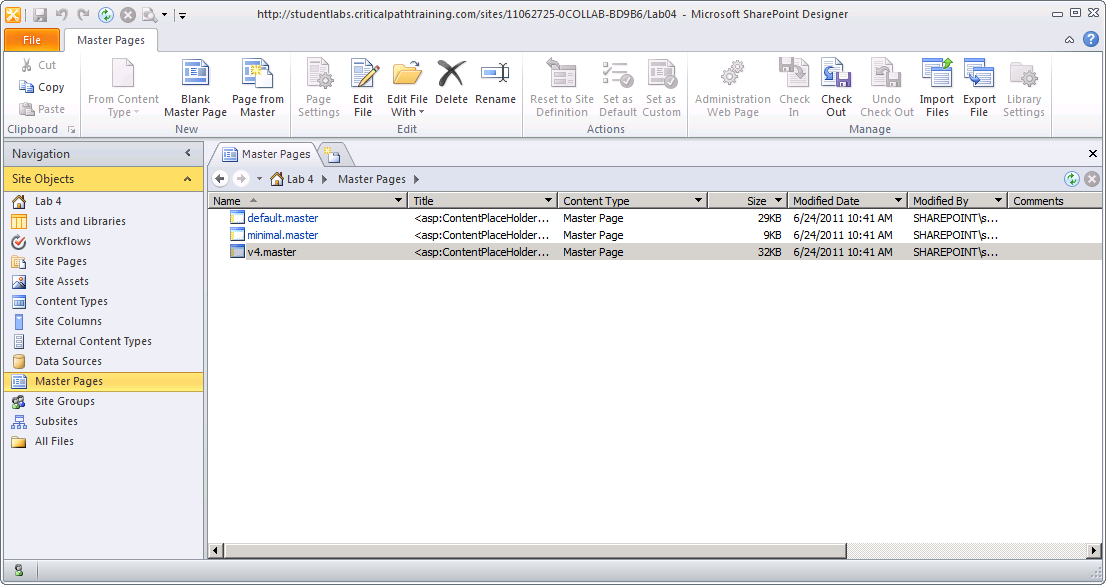
### Exercise 1: Customizing v4.master with a custom footer

In this exercise you will change the master page for the Lab 4 site. The objective is for you to gain experience customizing master pages in SharePoint Designer.

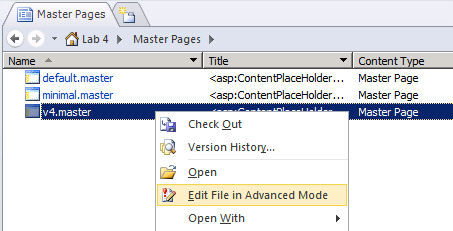
1. Using the browser, navigate to the Collaboration Site Collection located at **http://[[COLLAB-SITE]]**. Next, navigate to the child site for **Lab 4** by using the global navigation menu of the top-level site. You should be able to verify that this is a standard SharePoint site that contains a few lists and document libraries.



1. Open the **Lab 4** site in SharePoint Designer at **http://[[COLLAB-SITE]]/Lab04**. Next, click the **Master Pages** link in the **Site Objects** section and verify that there are three master pages in the master page gallery for the site, including the one you will customize in this lab named **v4.master**.



1. Right-click **v4.master** and then click the **Edit File in Advanced Mode** menu command to open this master page for editing.



1. Make sure you are editing **v4.master** using Code View. Scroll down to around line 624 and find the tag for the **DeveloperDashboard** control.

<SharePoint:DeveloperDashboard runat="server"/>

1. Immediately after the **DeveloperDashboard** control, add the following div element to create a custom footer for the page:

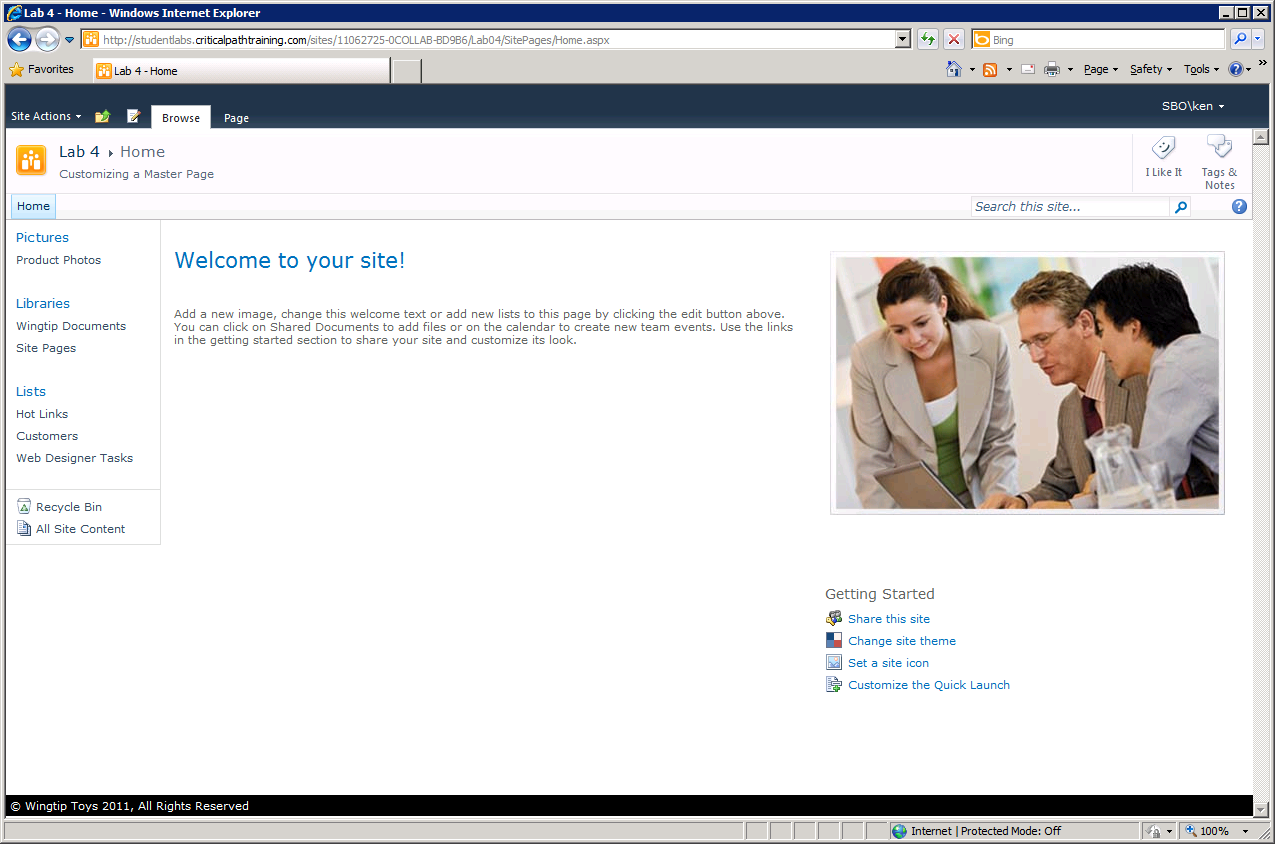
<SharePoint:DeveloperDashboard runat="server"/>

<div style="background-color: black; color:white; padding: 4px; clear: both;" >

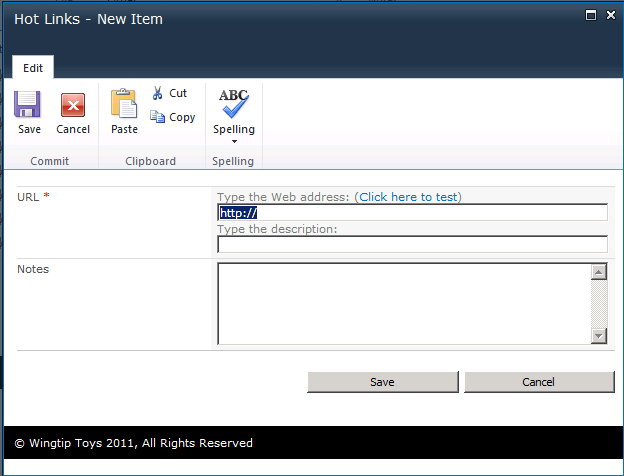
&copy; Wingtip Toys 2011, All Rights Reserved

</div>

1. Now test your work. Save your changes to **v4.master** and then refresh the site's home page in the browser. You should see your custom footer at the bottom of the page. You should also see the footer as you navigate to other pages in the **Lab 4** site.



1. Now it's time to see how your custom footer effect dialogs such as the ones SharePoint 2010 uses to add new items. In the browser, click the link in the Quick launch bar with the caption of **Hot Links** to navigate to the default view for that list. Once you have navigated to the **Hot Links** list, click on the link with the caption **Add new link** which will display a **New Item** dialog with your custom footer at the bottom.



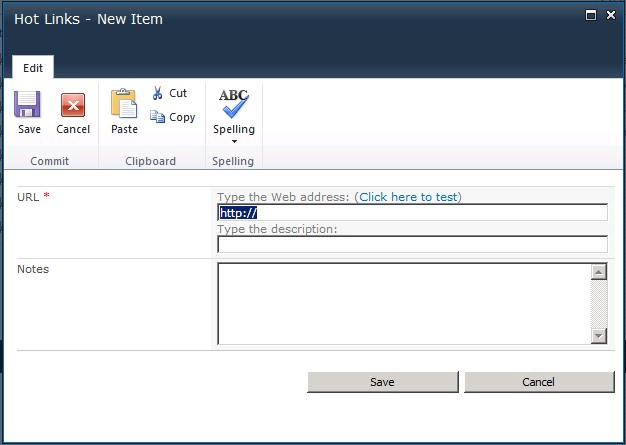
1. From the previous step, you can observe that elements added to a master page are also shown on dialogs by default. In this step you will make a change to the div element you created for the custom footer so that it does not appear in dialogs. In SharePoint Designer, return to Code View of v4.master and locate the div you added for your custom footer. Add the class **s4-notdlg** so that this div element is hidden in dialogs. Save your changes to **v4.master**.

<div style=”...” class=”s4-notdlg” >

&copy; Copyright 2011 Wingtip Toys

</div>

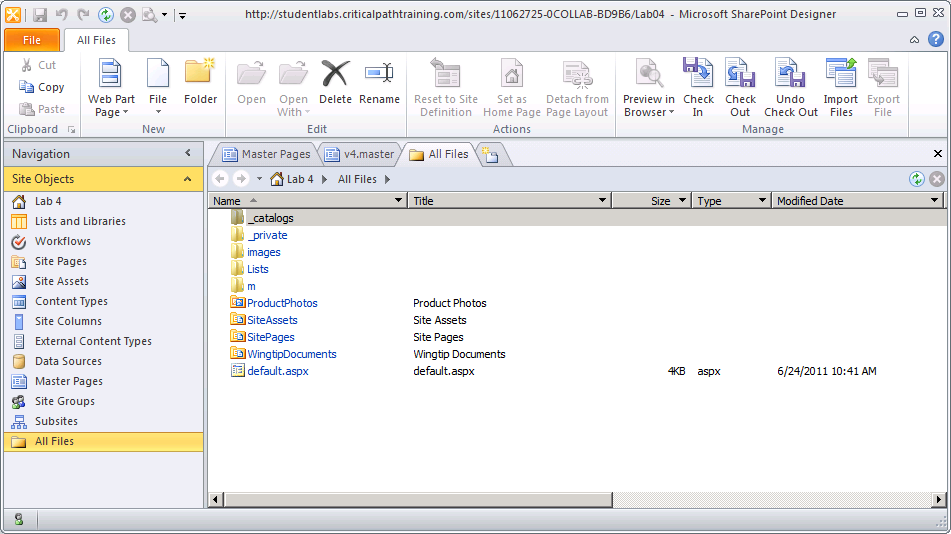
1. Now test your work. Go back to the browser and return to the **Hot Links** list. Click the link with the caption of Add new link to display the New Item dialog. You should be able to verify that your custom footer is now hidden in dialogs.



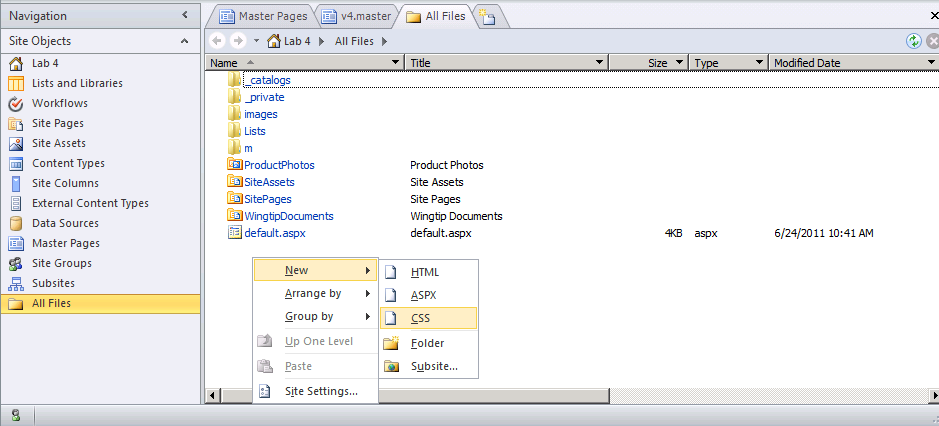
### Exercise 2: Linking A Master Page to a custom CSS file

In this exercise you will move your CSS rules out into a separate CSS file.

1. In the SharePoint Designer, click on the **All Files** link in **Site Objects** section. In the **All Files** view in the right-hand side of the screen you should be able to see all the files and folders that exist at the root of the **Lab 4** site.



1. In this step you will add a custom CSS file to the root of the site. Right-click in the white space of the **All Files** view and select the **New > CSS** menu command to create a new CSS file.



1. Rename the new CSS file to **custom\_styles.css**.



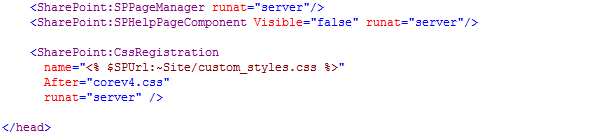
1. Now you will return to **v4.master** and remove the inline styles you added to the div element for the custom footer. The objective is to move the CSS styles for the custom footer into **custom\_styles.css**. When removing the styles from the div element for the custom footer, add an **id** of **custom\_footer** so that you can create a selector in the CSS file to identify that element. Once you have modified the div element for the custom footer, it should now look like this:

<div id="custom\_footer" class="s4-notdlg" >

&copy; Wingtip Toys Inc 2011, All Rights Reserved.

</div>

1. Now you must add a link into the bottom of the **head** section of **v4.master** to reference **custom\_styles.css**. Accomplish this by adding a **CssRegistration** control that is parameterized like the control shown in the following example.



1. Save your work by saving your changes to **v4.master**.
2. Open the new CSS file named **custom\_styles.css** in Advanced File Mode and add the following CSS rule:

#custom\_footer{

background-color: black;

color:white;

padding: 4px;

clear: both;

}

1. Save your changes to **custom\_styles.css**. You should now be able to return to the browser and refresh the page to verify that the custom footer still looks the way it should.

Note that in the majority of SharePoint branding scenarios you should prefer to add custom CSS files inside the **Style Library** in the top level site and link to it from that location. However, in this lab you have created a custom CSS file at the root of a child site to keep all your work in one location, making things easier when you are learning how to work with master pages and CSS files at the same time.

### Exercise 3: Customizing a Master Page with a fixed width

In this exercise you will add CSS rules to **custom\_styles.css** and modify **v4.master** to support a fixed-width page layout. You will also play with the SharePoint scrolling system by disabling scrolling in the standard **#s4-workspace** div element and enabling scrolling at the page body level.

1. Add the following CSS rules to **custom\_styles.css**. Save your work when you are done.

body.v4master {

background-color: orange;

overflow:auto;

}

#s4-ribbonrow, #s4-workspace {

width: 960px;

max-width: 960px;

margin: 0 auto;

padding: 0px;

}

body #s4-workspace {

background-color: transparent;

overflow:hidden !important;

}

#s4-bodyContainer{

background-color:white;

}

#MSO\_ContentTable {

min-height: 420px;

}

1. Return to **v4.master** in Code View and place your cursor at the bottom of the head section right below the **CssRegistration** control you added earlier. Add a script section with an empty implementation of the JavaScript function named **FixRibbonAndWorkspaceDimensions**. This will effectively stop SharePoint from trying to dynamically resize the page each time it is refreshed. Also add the functions **ProcessImn** and **ProcessImnMarkers,** This will effectively stop SharePoint from trying to create ActiveX controls in the browser which result in annoying (and scary) messages to the user.

<script type="text/javascript">

function FixRibbonAndWorkspaceDimensions(){}

function ProcessImn(){}

function ProcessImnMarkers(){}

</script>

1. In **v4.master**, scroll down to the body tag and locate the **scroll** attribute with a value of **no**.

<body scroll="no" onload="..."

1. Remove the **scroll** attribute and save your changes to **v4.master**. Now test your work. You should be able to refresh pages in the browser and see that the site now has pages with a fixed-width layout.

