Lab 05: Creating Custom Page Layouts

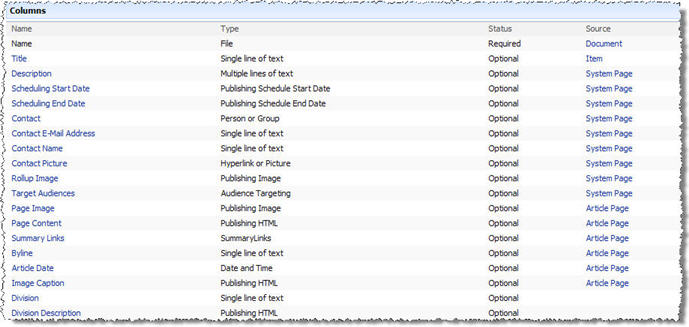
**Lab Time:** 60 minutes

**Lab Overview:** In this lab you will practice creating a new content type and site columns based off the Publishing Page content type. Then you will create a page layout using SharePoint Designer based off this content type. After using the browser and SharePoint Designer to create a new content type and page layout, you will then do the same thing except only using Visual Studio and Features.

Exercise 1: Creating new site columns & a content type based off the Page content type

In this exercise you will create a new content type using the SharePoint browser interface. The content type you create will contain additional site columns from the OOTB Page content type.

1. First you need to create two new site columns. Open a browser and navigate to **http://wcm.litwareinc.com**. Then select **Site Actions » Site Settings » Modify All Site Settings**.
2. On the **Site Settings** page, select **Site columns** from the **Galleries** section. On the **Site Column Gallery** page, select **Create**.
3. On the **New Site Column: Litware Inc.** page, use the following information to create a new site column and click OK:
   * **Name and Type - Column name:** Division
   * **Name and Type - The type of information in this column is:** Single line of text
   * **Group: New group:** WCM401
4. Repeat the previous step using the following information to create another site column and click OK:
   * **Column name:** Division Description
   * **The type of information in this column is:** Full HTML content with formatting and constraints for publishing
   * **Group: Existing group:** WCM401
5. Using the breadcrumb navigation, select **Site Settings**. On the **Site Settings** page, select **Site content types** from the **Galleries** section. On the **Site Content Type Gallery** page, select Create.
6. On the **New Site Content Type** page, use the following information to create a new content type and click OK:
   * **Name:** Division Article
   * Select parent content type from: Page Layout Content Types
   * Parent Content Type: Article Page
   * Group: New Group: WCM401
7. On the Site Content Type: Division Article page, select Add from existing site columns.
8. On the Add Columns to Site Content Type: Division Article page, select the two fields created in steps 3 & 4 (found in group WCM401) and click OK.
9. You should now see the two columns listed along with all the other site columns inherited by this content type as shown in the following image:

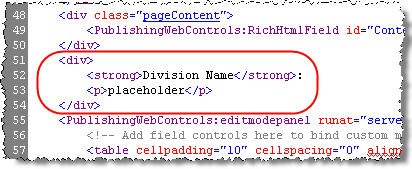


At this point you now have created a few new site columns and a new content type that includes these site columns and inherits from the Page content type.

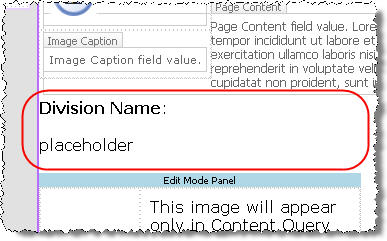
Exercise 2: Creating a new page layout using SharePoint Designer

In this exercise you will create a new page layout associated with the content type created in the previous exercise using SharePoint Designer. Then you will create a new content page based off the page layout and content type.

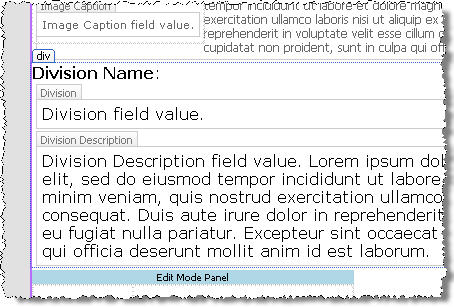
1. Open **SharePoint Designer** and open the site **http://wcm.litwareinc.com**.
2. Now you need to create a new page layout based off the content type created in the previous step. From within SharePoint Designer, select **File » New...**.
3. In the **New** dialog, select the **SharePoint Content** tab, then **SharePoint Publishing** in the left-hand panel and **Page Layout** in the center panel. Use the following information to complete the remainder of the dialog and click OK:
   * **Content Type Group:** WCM401
   * **Content Type Name:** Division Article
   * **URL Name:** DivisionArticleImageLeft.aspx
   * **Title:** Division article with image on left
4. Now you need to add some structure to this page. Cheat a bit by copying the contents of the **ArticleLeft.aspx** page layout and paste them into **DivisionArticleImageLeft.aspx**, replacing everything that was there (do this only within the **Code** view, ~not~ within Design or Split mode).
5. Now you need to add the two fields added to the content type. These will be added just below the main content of the page. Switch to Code mode if you have not already and add the following HTML just after the closing <div> tag for the page content (on or about line #50), as shown by the image below (the circled portion is the new code you should add):



1. Save your changes. Now, switch into **Design** view... you should see an image similar to the following one (the highlighted part is where you added the HTML):



1. The next thing to do is add the columns you added to the content type. In the **Toolbox** tool window in SharePoint Designer (right-hand side of the application), click the C:\Development\Writing\Courses\WCM401\trunk\Student Download\Labs\05_PageLayouts\Figures\Box.jpg icon (found in the upper right corner of the tool window) to give yourself more room to work. Scroll the tool window down to the **SharePoint Controls**... specifically the **Content Fields** within the **SharePoint Controls** group. Notice the two fields you created and added to the content type previously. Drag the two fields into the design surface, placing the **Division** field next to **Division:** in the ASPX file and **Division Description** immediately after **placeholder**. Now remove the placeholder text... you should see something similar to the following image (if yours are arranged slightly differently, it is not a problem... as long as there are no rendering problems you are in good shape):



1. Save all your changes. Now you need to check in and publish the file (same reasons why you had to / didn't have to with the master page in the last lab). Right-click **DivisionArticleImageLeft.aspx** in the **Folder List** tool window and select **Check In**. Then select **Publish a major version** and click **OK**. A dialog will appear asking if you want to view/modify the approval status of the page layout. Click **Yes** which will open a new browser window loading the Master Page Gallery with your page layout near the top of the list. From the ECB menu of the **DivisionArticleImageLeft.aspx** page layout, select Approve/reject. On the **Master Page Gallery: DivisoinArticleImageLeft** page, select **Approved**. This item will become visible to all users.
2. At this point the page layout has been published to the Master Page Gallery and you can now create content pages based off the page layout and content type. Browse to the Press Releases section within the **http://wcm.litwareinc.com** site.
3. From the **Press Releases** section, select **Site Actions » Create Page**. On the **Create Page** page, use the following information to create the new content page and click Create:
   * **Title:** Division Article 1
   * **URL Name:** DivisionArticle1
   * **Page Layout:** (Division Article) Division article with image on left
4. You should now see your page in edit mode. Once you have entered the desired content, select **Submit for Approval** at the top of the page to start the page approval workflow.
5. On the **Start "Parallel Approval": DivisionArticle1** page, click **Start**.
6. The page will then load with the **Division Article 1** page you just created, but not in edit mode. To advance the page through the workflow, select the **Approve** button in the Page Editing Toolbar.
7. On the **Workflow Tasks: Please approve DivisionArticle1** page, optionally enter any comments and click the **Approve** button. The browser will refresh with the published page in display mode.

At this point you have created site columns and a content type using the browser and a page layout using SharePoint Designer. Next you'll see how to do the same thing without the browser or SharePoint Designer.

Exercise 3: Creating new site columns & a content type using Features

In this exercise you will create a Feature that, when activated, will create a few new site columns and content type based from the OOTB content type Page (and including the site columns created).

Since you have now created and customized Visual Studio projects that compile then package the contents of the project into a WSS solution package, as well as creating a project that doesn't compile anything but simply packages the contents into a WSS solution package, we have given you a project to get started with. This project already has the directory structure created, the deployment files added and the changes to the project file to run the automated WSS solution package creation.

1. In Visual Studio, open the **Lab5** solution located in the following directory:

c:\Student\Labs\05\_PageLayouts\Lab\Lab5.sln

1. The first thing you need to do is create the Feature definition file. Create a new XML file named **feature.xml** in the **Lab5** Feature folder in the project, filling it with the following markup:

<?xml version="1.0" encoding="utf-8" ?>

<Feature xmlns="http://schemas.microsoft.com/sharepoint/"

Id="E052E45D-B487-4593-AF2B-D6283E21C43A"

Title="Lab 5 - Working with Provisioned Site Columns, Content Types and Page Layouts"

Hidden="FALSE"

Scope="Site"

Version="1.0.0.0">

<ElementManifests>

<ElementManifest Location="SiteColumns.xml" />

<ElementManifest Location="ContentType.xml" />

</ElementManifests>

</Feature>

1. Next, create the two element manifest files listed in the Feature definition file. Create two XML files in the **Lab5** Feature folder in the project named **SiteColumns.xml** & **ContentType.xml**.
2. Open the **SiteColumns.xml** file and add the following XML markup (you can omit the XML comments, they are here just as a comment to what the XML is doing):

<?xml version="1.0" encoding="utf-8" ?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<!-- new site column off the "site line of text" field type -->

<Field SourceID="http://schemas.microsoft.com/sharepoint/"

ID="{DC172FD7-CC9D-493c-A1EB-231B8AF29F3E}"

Name="Division2"

DisplayName="Division"

Group="WCM401Feature"

Type="Text"

Required="FALSE"

Sealed="FALSE"

Hidden="FALSE" />

<!-- new site column based off the "publishing html" field type -->

<Field SourceID="http://schemas.microsoft.com/sharepoint/"

ID="{8DDDC83E-625C-4d5e-B4EF-0CD8A3A434C2}"

Name="DivisionDescription"

DisplayName="Division Description"

Group="WCM401Feature"

Type="HTML"

Required="FALSE"

Sealed="FALSE"

Hidden="FALSE" />

</Elements>

1. Open the **ContentType.xml** file and add the following XML markup (you might want to read the note after the code first, makes this less painful):

<?xml version="1.0" encoding="utf-8" ?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<!-- this content type inherits from the "article page" content type -->

<ContentType ID="0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900242457EFB8B24247815D688C526CD44D00976A5C165AB04065B1F9AFAE0E4658B5"

Name="Division Article Feature"

Group="WCM401Feature">

<FieldRefs>

<FieldRef ID="{DC172FD7-CC9D-493c-A1EB-231B8AF29F3E}" Name="Division2" />

<FieldRef ID="{8DDDC83E-625C-4d5e-B4EF-0CD8A3A434C2}" Name="DivisionDescription" />

</FieldRefs>

<DocumentTemplate TargetName="/\_layouts/CreatePage.aspx" />

</ContentType>

</Elements>

Notice the very long content type ID. This content type is based off the Article Page content type. Don't try to type this in... copy it from the following file:

c:\Program Files\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\FEATURES\PublishingResources\PublishingContentTypes.xml

On or about **line #80**, you will find the **Article Page** content type (or search for the string **$Resources:cmscore,contenttype\_articlepage\_name;**, the name of the file). Copy the value from the ID field into the content type ID you are creating in this step. Then you need to add some uniqueness to it... do this by appending **00**, then a new GUID (without the brackets or hyphens). So, the new ID would be something like this:

[ContentType ID from Article Page] + 00 + [GUID with no brackets or dashes]

Another thing to notice is the GUIDs for the two XML **<FieldRef>** elements. These must match exactly to the ID's of the two site columns created in the previous step.

1. Next, add the manifest file for the Feature and the necessary entries in the DDF file to package up the required files. Create a new XML file named **manifest.xml** to the **DeploymentFiles** folder in the project and add the following XML markup:

<?xml version="1.0" encoding="utf-8" ?>

<Solution xmlns="http://schemas.microsoft.com/sharepoint/"

SolutionId="BF7F3B37-953D-4732-8E20-9CE5F65B1F2E"

DeploymentServerType="WebFrontEnd"

ResetWebServer="FALSE">

<FeatureManifests>

<FeatureManifest Location="Lab5\feature.xml"/>

</FeatureManifests>

</Solution>

1. Finally, add the following lines to the **BuildSharePointPackage.ddf** file between the comments:

DeploymentFiles\manifest.xml

.Set DestinationDir=Lab5

TEMPLATE\FEATURES\Lab5\Feature.xml

TEMPLATE\FEATURES\Lab5\SiteColumns.xml

TEMPLATE\FEATURES\Lab5\ContentType.xml

At this point you have created a Feature that will create new site columns and a new content type that (1) inherits from the Page content type and (2) contains the site columns created by the feature.

Exercise 4: Creating a new page layout using Features

In this exercise you will update the Feature created in the previous exercise to include creating a page layout with an associated preview image. Once this is done, it will then be time to deploy the solution, activate the feature, and test everything!

1. The first thing you need to do is add the resource files to the Feature. Copy the **DivisionArticleImageLeft.aspx** and **SamplePageLayoutPreview.gif** files into the **Lab5** Feature folder within the project.
2. Next, create a new XML file in the **Lab5** Feature folder named **PageLayout.xml** and add the following XML markup to the file:

<?xml version="1.0" encoding="utf-8" ?>

<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

<!-- add the page layout to the Master Page Gallery -->

<Module Url="\_catalogs/masterpage"

RootWebOnly="TRUE">

<!-- provision the page layout into the Master Page Gallery -->

<File Url="DivisionArticleImageLeft.aspx"

Name="DivisionArticleImageLeftFeature.aspx"

Type="GhostableInLibrary">

<!-- specify the content type associated this page layout is associated with -->

<Property Name="PublishingAssociatedContentType"

Value=";#Division Article Feature;#0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900242457EFB8B24247815D688C526CD44D00976A5C165AB04065B1F9AFAE0E4658B5;#" />

<!-- specify the URL to the preview image, provisioned below -->

<Property Name="PublishingPreviewImage"

Value="~SiteCollection/\_catalogs/masterpage/Preview Images/Litware/SamplePageLayoutPreviewFeature.gif, ~SiteCollection/\_catalogs/masterpage/Preview Images/Litware/SamplePageLayoutPreviewFeature.gif" />

<!-- specify the page layout content type -->

<Property Name="ContentType"

Value="$Resources:cmscore,contenttype\_pagelayout\_name;" />

<!-- specify the title of the page layout -->

<Property Name="Title"

Value="Litware Division Article" />

</File>

</Module>

<!-- add the preview image to the Master Page Gallery -->

<Module Url="\_catalogs/masterpage/Preview Images/Litware"

RootWebOnly="TRUE">

<!-- provision the preview image into the Master Page Gallery -->

<File Url="SamplePageLayoutPreview.gif"

Name="SamplePageLayoutPreviewFeature.gif">

<!-- specify the title of the preview image -->

<Property Name="Title"

Value="SamplePageLayoutPreviewFeature.gif" />

</File>

</Module>

</Elements>

*Notice the page layout properties* ***PublishingAssociatedContentType*** *value looks a bit odd. That is a delimited string of two values separated by ";#". the first value is the name of the content type and the second is the content type ID.*

1. Now you need to add the files to the Feature definition file and DDF file for packaging. Open the **feature.xml** file and add the following XML markup to the **<ElementManifests>** node, after the two existing **<ElementManifest>** nodes:

<ElementManifest Location="PageLayout.xml" />

<ElementFile Location="DivisionArticleImageLeft.aspx" />

<ElementFile Location="SamplePageLayoutPreview.gif" />

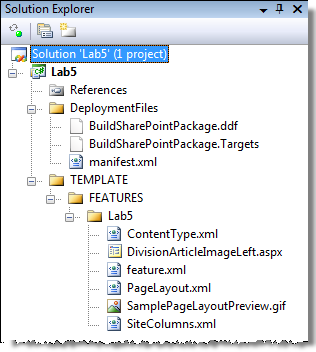
1. Open the **BuildSharePointPackage.ddf** file and add the following lines where the other files from the Feature are added to the package:

TEMPLATE\FEATURES\Lab5\PageLayout.xml

TEMPLATE\FEATURES\Lab5\DivisionArticleImageLeft.aspx

TEMPLATE\FEATURES\Lab5\SamplepageLayoutPreview.gif

1. When you save everything, your project should look like the following image:



1. Now it's time to deploy the WSS solution package.
2. First the WSS solution package must be deployed. Open a command prompt and navigate to the following directory:

c:\Program Files\Common Files\Microsoft Shared\web server extensions\12\BIN

1. Enter the following command into the command line window and hit **Enter**:

stsadm -o addsolution -filename c:\Student\Labs\05\_PageLayouts\Lab\wsp\Debug\Lab5.wsp

1. Launch Central Administration by selecting **Start » All Programs » Microsoft Office Server » SharePoint 3.0 Central Administration**.
2. From the **Central Administration** site, select the **Operations** tab and then select **Solution management** under the **Global Configuration** section.
3. On the **Solution Management** page, click the link on **lab5.wsp**.
4. On the **Solution Properties** page, select **Deploy Solution**.
5. On the **Deploy Solution** page, specify **Now** in the **Deploy When?** section and click OK.
6. Test the Feature by browsing to the **http://wcm.litwareinc.com/** site and select **Site Actions » Site Settings » Modify All Site Settings**.
7. On the **Site Settings** page, select **Site collection features** under the **Site Collection Administration** section.
8. On the **Site Collection Features** page, click **Activate** on the **Lab 5 - Working with Provisioned Site Columns, Content Types and Page Layouts Feature**.
9. With the Feature activated, go check the new site columns created in the **Site Column Gallery** and content type in the **Content Type Gallery**. Then create a new page based off the new page layout in the **Press Releases** section.

At this point you have now created site columns, content types and a page layout using the SharePoint browser interface and SharePoint Designer. You then created a Feature that does the same thing but instead of creating everything in a customized sense within the content database, everything is template!