Lab 9: Using the WSS Provisioning Engine for Publishing

**Lab Time: 45 Minutes**

**Lab Directory: ECM401.SiteProvisioning**

**Lab Overview:**

In this lab you will create a SharePoint Feature that will provision a set of site columns, content types and page layouts for web publishing. The content type will be based on the default Page content type installed by the publishing framework.

# Exercise 1: Creating Site Columns using a SharePoint Feature

1. Start by creating a Class Library project named **ECM401.Publishing** using the **SharePoint Feature** project template. Accept the default name and title, enter a description such as *"This feature creates and deploys custom publishing resources."* , and select **Site** as the scope for the feature. Press **Create** to close the **Feature Wizard** dialog box.

Note: You must select **Site** because you will be provisioning **Field** elements, which are not allowed at **Web** scope.

1. Open the feature.xml file located in the **12\TEMPLATE\FEATURES\ECM401.Publishing** folder. Replace the contents of the **<ElementManifests>** section with the highlighted lines shown below.

<ElementManifests>  
<ElementManifest Location="SiteColumns.xml"/>  
<ElementManifest Location="ContentTypes.xml"/>  
</ElementManifests>

1. Save and close the **feature.xml** file. Next, you will create two element manifest files - one for site columns and another for content types.
2. First, rename the existing **elements.xml** file to **SiteColumns.xml** . Replace the file contents with the following XML code.

<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"  
StaticName="Division2"  
DisplayName="Division"  
Group="ECM401"  
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"  
StaticName="DivisionDescription"  
DisplayName="Division Description"  
Group="ECM401"  
Type="HTML"  
Required="FALSE"  
Sealed="FALSE"  
Hidden="FALSE" />  
</Elements>

1. Now add a second XML file to the same folder named **ContentTypes.xml** and replace its contents with the following XML code.

Note: **YOU WILL \*\*MODIFY\*\* THE CONTENT TYPE ID IN THE NEXT STEP**

<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="ECM401">  
<FieldRefs>  
<FieldRef ID="{DC172FD7-CC9D-493c-A1EB-231B8AF29F3E}" Name="Division" />  
<FieldRef ID="{8DDDC83E-625C-4d5e-B4EF-0CD8A3A434C2}" Name="DivisionDescription" />  
</FieldRefs>  
<DocumentTemplate TargetName="/\_layouts/CreatePage.aspx" />  
</ContentType>  
</Elements>

Note: The content type ID you use here will cause this content type to inherit its metadata from the **Article Page** content type created by the MOSS Publishing framework.To get the correct identifier, perform the following steps:

* Navigate to the **C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\FEATURES\PublishingResources** folder in **Windows Explorer** .
* Copy the **PublishingContentTypes.xml** file to **C:\** .
* Open the **C:\PublishingContentTypes.xml** file in Visual Studio and search for the string "$Resources:cmscore,contenttype\_articlepage\_name;".
* Copy the content type ID into the ID attribute of the content type you are creating.
* Append **"00"** to your new ID.
* Select **Create GUID** from the **Tools** menu in Visual Studio. Copy and paste the new GUID after the "00".
* Remove the curly braces and hyphens from the identifier.

1. You now have a feature that will create new site columns and a new Page content type containing those columns. In the next exercise, you will add a page layout to the feature.

# Exercise 2: Adding a Page Layout

In this exercise you will update the Feature created in the previous exercise to include a custom page layout with an associated preview image. Once this is done, you can deploy the solution, activate the feature, and begin testing.

1. Start by creating the master page template file. Right-click on the **ECM401.Publishing** folder in the Visual Studio **Solution Explorer** and select **Add > New Item...** from the context menu. Create a new text file named "ECM401ArticleLayout.aspx".
2. Replace everything in the file using the "ECM401ArticleLayout" code snippet.

XML Snippet: 'ECM401 Article Layout'

<%@ Page language="C#"   
 Inherits="Microsoft.SharePoint.Publishing.PublishingLayoutPage,Microsoft.SharePoint.Publishing,Version=12.0.0.0,Culture=neutral,PublicKeyToken=71e9bce111e9429c" meta:progid="SharePoint.WebPartPage.Document" %>  
  
<%@ Register Tagprefix="SharePointWebControls" Namespace="Microsoft.SharePoint.WebControls" Assembly="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>   
  
<%@ Register Tagprefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPages" Assembly="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>   
  
<%@ Register Tagprefix="PublishingWebControls" Namespace="Microsoft.SharePoint.Publishing.WebControls" Assembly="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>   
  
<%@ Register Tagprefix="PublishingNavigation" Namespace="Microsoft.SharePoint.Publishing.Navigation" Assembly="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>  
  
<asp:Content ContentPlaceholderID="PlaceHolderAdditionalPageHead" runat="server">  
 <PublishingWebControls:editmodepanel runat="server" id="editmodestyles">  
  
 <!-- Styles for edit mode only-->  
 <SharePointWebControls:CssRegistration name="<% $SPUrl:~sitecollection/Style Library/~language/Core Styles/zz2\_editMode.css %>" runat="server"/>  
  
 </PublishingWebControls:editmodepanel>  
  
 <SharePointWebControls:CssRegistration name="<% $SPUrl:~sitecollection/Style Library/~language/Core Styles/rca.css %>" runat="server"/>  
  
 <SharePointWebControls:FieldValue id="PageStylesField" FieldName="HeaderStyleDefinitions" runat="server"/>  
</asp:Content>  
  
<asp:Content ContentPlaceholderID="PlaceHolderPageTitle" runat="server">  
 <SharePointWebControls:FieldValue id="PageTitle" FieldName="Title" runat="server"/>  
</asp:Content>  
  
<asp:Content ContentPlaceholderID="PlaceHolderPageTitleInTitleArea" runat="server">  
 <SharePointWebControls:TextField runat="server" id="TitleField" FieldName="Title"/>  
</asp:Content>  
  
<asp:Content ContentPlaceHolderId="PlaceHolderTitleBreadcrumb" runat="server">  
 <div class="breadcrumb">  
 <asp:SiteMapPath ID="siteMapPath" Runat="server" SiteMapProvider="CurrentNavSiteMapProviderNoEncode"  
 RenderCurrentNodeAsLink="false" SkipLinkText="" CurrentNodeStyle-CssClass="breadcrumbCurrent" NodeStyle-CssClass="ms-sitemapdirectional"/>  
 </div>  
</asp:Content>  
  
<asp:Content ContentPlaceholderID="PlaceHolderMain" runat="server">  
  
 <SharePointWebControls:CssRegistration name="<% $SPUrl:~sitecollection/Style Library/~language/Core Styles/pageLayouts.css %>" runat="server"/>  
  
 <div style="clear:both">&nbsp;</div>  
  
 <table class="floatLeft" cellspacing=0 cellpadding=0>  
 <tr>  
 <td class="image">  
 <PublishingWebControls:RichImageField id="ImageField" FieldName="PublishingPageImage" runat="server"/>  
 </td>  
 </tr>  
 <tr>  
 <td class="caption">  
 <PublishingWebControls:RichHtmlField id="Caption" FieldName="PublishingImageCaption" AllowTextMarkup="false" AllowTables="false" AllowFonts="false" PreviewValueSize="Small" runat="server"/>  
 </td>  
 </tr>  
 </table>  
  
 <table class="header">  
 <tr>  
 <td class="dateLine">  
 <SharePointWebControls:datetimefield FieldName="ArticleStartDate" runat="server" id="datetimefield3"></SharePointWebControls:datetimefield>  
 </td>  
 <td width=100% class="byLine">  
 <SharePointWebControls:TextField FieldName="ArticleByLine" runat="server"/>  
 </td>  
 </tr>  
 </table>  
  
 <div class="pageContent">  
 <PublishingWebControls:RichHtmlField id="Content" FieldName="PublishingPageContent" runat="server"/>  
 </div>  
 <div style="clear:both"></div>  
 <div>  
 <strong>Division Name</strong>:  
 <SharePointWebControls:TextField FieldName="Division" runat="server" id="TextField1"></SharePointWebControls:TextField>  
 <br /><PublishingWebControls:RichHtmlField FieldName="Division\_x0020\_Description" runat="server" id="RichHtmlField1"></PublishingWebControls:RichHtmlField>  
 </div>  
  
 <PublishingWebControls:editmodepanel runat="server" id="editmodepanel1">  
 <!-- Add field controls here to bind custom metadata viewable and editable in edit mode only.-->  
 <table cellpadding="10" cellspacing="0" align="center" class="editModePanel">  
 <tr>  
 <td>  
 <PublishingWebControls:RichImageField id="ContentQueryImage" FieldName="PublishingRollupImage" AllowHyperLinks="false" runat="server" />  
 </td>  
 <td width="200">  
 <asp:Label text="<%$Resources:cms,Article\_rollup\_image\_text%>" runat="server" />  
 </td>  
 </tr>  
 </table>  
 </PublishingWebControls:editmodepanel>  
</asp:Content>

1. Now copy the image file **ECM401ArticlePageLayout.gif** from the **C:\Student\Resources\Images** folder into the **ECM401.Publishing** folder of the project.
2. Create a new XML file in the same folder and give it the name **PageLayouts.xml** . Replace its contents with the following XML code.

<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="ECM401ArticleLayout.aspx"  
 Name="ECM401ArticleLayoutFeature.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/ECM401ArticlePageLayoutFeature.gif, ~SiteCollection/\_catalogs/masterpage/Preview Images/Litware/ECM401ArticlePageLayoutFeature.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="ECM401ArticlePageLayout.gif"  
 Name="ECM401ArticlePageLayoutFeature.gif">  
 <!-- specify the title of the preview image -->  
 <Property Name="Title"  
 Value="ECM401ArticlePageLayoutFeature.gif" />  
 </File>  
 </Module>  
</Elements>

Note: You need to edit the highlighted attribute of the property named **PublishingAssociatedContentType** so that it matches the content type name and ID you constructed previously. **TAKE CARE TO ENSURE THAT THE ";#" DELIMITER REMAINS IN TACT. IT OCCURS THREE TIMES, AS IN ";#name;#id;#"**

1. Open the **feature.xml** file for editing and insert the following highlighted lines.

<Feature xmlns="http://schemas.microsoft.com/sharepoint/"  
Id="fcafc2f0-2a2d-42b7-bb6f-3a83a8f90819"  
Hidden="FALSE"  
Title="ECM401 - Publishing Feature"  
Description="This feature creates custom publishing resources."  
Scope="Site"  
ReceiverAssembly="ECM401.Publishing, Version=1.0.0.0, Culture=neutral, PublicKeyToken=6dc3927da2abcbba"  
ReceiverClass="ECM401.Publishing.FeatureReceiver"  
Version="1.0.0.0">  
  
<ElementManifests>  
<ElementManifest Location="SiteColumns.xml"/>  
<ElementManifest Location="ContentTypes.xml"/>  
<ElementManifest Location="PageLayouts.xml"/>  
</ElementManifests>  
  
</Feature>

1. Now you are ready to build the project and deploy the solution. Select the **DebugDeploy** configuration and build the project, then navigate to the **Litware Publishing** site to activate the feature. Check the **Site Column Gallery** to ensure that the new site columns were created and then check the **Content Type Gallery** to ensure that the new content type was created. Finally, create a new page using the new page layout.

Note: You have successfully created and deployed custom publishing resources using a SharePoint Feature instead of SharePoint Designer.

**This concludes the lab exercises.**