

# *Microsoft*<sup>®</sup> Virtual Labs

**Commerce Server 2007:  
Customization and Extensibility**

**Microsoft**<sup>®</sup>

# Table of Contents

---

<b>Commerce Server 2007: Customization and Extensibility .....</b>	<b>1</b>
Exercise 1 Extend Catalog Management .....	2
Lab Summary .....	13

# Commerce Server 2007: Customization and Extensibility

---

## Objectives

After completing this lab, you will be better able to:

- Extend Catalog Manager UI
- Extend Catalog Web Service
- Extend Authorization Manger roles

## Scenario

Adventure Works an e-Commerce site has deployed Commerce Server 2007. Although most of the functionality of Commerce Server 2007 meets their needs they require auditing of their Catalog Data and only allow Catalog Administrators to view the audit trail. As a developer for Adventure Works Enterprise, you have been selected to extend their management tools.

## Estimated Time to Complete This Lab

60 Minutes

## Computer used in this Lab




CS2007RTM

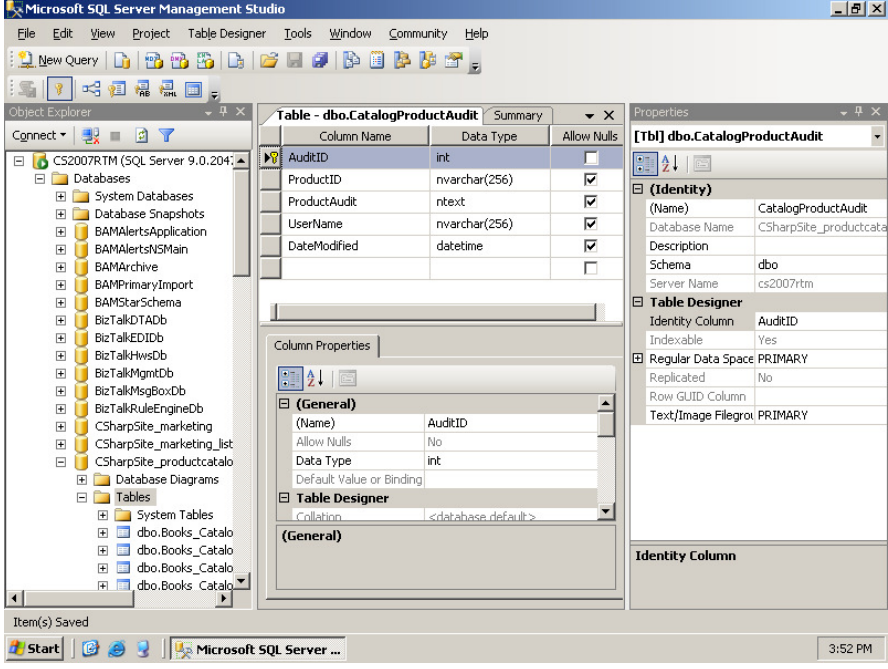
# Exercise 1

## Extend Catalog Management

### Scenario

In this exercise you will create a custom table to log the audit trail for product changes and extend the Catalog Web Service to log and retrieve product audits. You will extend the Authorization Manager to allow Catalog Administrators to view the audit trail for products and modify the Catalog Manager to view the audit logs.

Tasks	Detailed Steps																								
<p><b>Complete the following 6 tasks on:</b></p> <p> <b>CS2007RTM</b></p> <p>1. Extend Product Catalog Database for SQL Server 2005</p>	<p>a. Select <b>Start   All Programs   Microsoft SQL Server   SQL Server Management Studio</b></p> <p>b. When prompted to connect to server select <b>Connect</b>.</p> <p>c. From <b>SQL Server Management Studio</b> console, navigate to <b>Object Explorer</b> and expand the tab for the local server. Expand <b>Databases   CSharpSite_productcatalog   Tables</b>.</p> <p>d. Right click and select <b>New Table</b>.</p> <p>e. From the <b>New Table</b> screen create a table called <b>CatalogProductAudit</b>. Create five columns with the following data:</p> <table border="1" data-bbox="508 978 1409 1226"> <thead> <tr> <th>Column Name</th> <th>Data Type</th> <th>Length</th> <th>Allow Nulls</th> </tr> </thead> <tbody> <tr> <td>AuditID</td> <td>Int</td> <td>4</td> <td>Unselected</td> </tr> <tr> <td>ProductID</td> <td>nvarchar(256)</td> <td>256</td> <td>Selected</td> </tr> <tr> <td>ProductAudit</td> <td>Ntext</td> <td>16</td> <td>Selected</td> </tr> <tr> <td>UserName</td> <td>nvarchar(256)</td> <td>256</td> <td>Selected</td> </tr> <tr> <td>DateModified</td> <td>Datetime</td> <td>8</td> <td>Selected</td> </tr> </tbody> </table> <p>f. In the <b>Table Designer</b> Properties section, specify the Identity Column as <b>AuditID</b>.</p> <p>g. Right click on the <b>AuditID</b> row in the Table Designer Window and click “<b>Set Primary Key</b>”</p> <p>h. After creating all columns select the <b>Save</b> icon to save your changes. Your screen should appear as follows:</p>	Column Name	Data Type	Length	Allow Nulls	AuditID	Int	4	Unselected	ProductID	nvarchar(256)	256	Selected	ProductAudit	Ntext	16	Selected	UserName	nvarchar(256)	256	Selected	DateModified	Datetime	8	Selected
Column Name	Data Type	Length	Allow Nulls																						
AuditID	Int	4	Unselected																						
ProductID	nvarchar(256)	256	Selected																						
ProductAudit	Ntext	16	Selected																						
UserName	nvarchar(256)	256	Selected																						
DateModified	Datetime	8	Selected																						

Tasks	Detailed Steps
	
<p>2. Create Stored Procedures to Query and Update Database.</p>	<ol style="list-style-type: none"> <li>Create a stored procedure to update this table by clicking the <b>New Query</b> button.</li> <li>Select your connection as <b>Windows Authentication</b> and connect to <b>CS2007RTM</b> computer next select <b>Connect</b> button.</li> <li>Next we will create an insert record stored procedure to record audits. From the <b>Tools Window</b> select the <b>CSharpSite_ProductCatalog</b> database then type the following stored procedure then select <b>Execute</b> from the <b>Tools Window</b>:             <pre data-bbox="542 1115 1398 1814"> CREATE PROCEDURE sp_InsertCatalogProductAudit (@ProdID nvarchar(256), @Audit ntext, @UserName nvarchar(256), @Date datetime ) AS INSERT INTO [CSharpSite_productcatalog].[dbo].[CatalogProductAudit] (     [ProductID],     [ProductAudit],     [UserName],     [DateModified] ) VALUES (     @ProdID,     @Audit,     @UserName,     @Date )             </pre> </li> <li>Create a stored procedure to query all audit information about a product. Click the <b>New Query</b> button and type the following stored procedure then select <b>Execute</b></li> </ol>

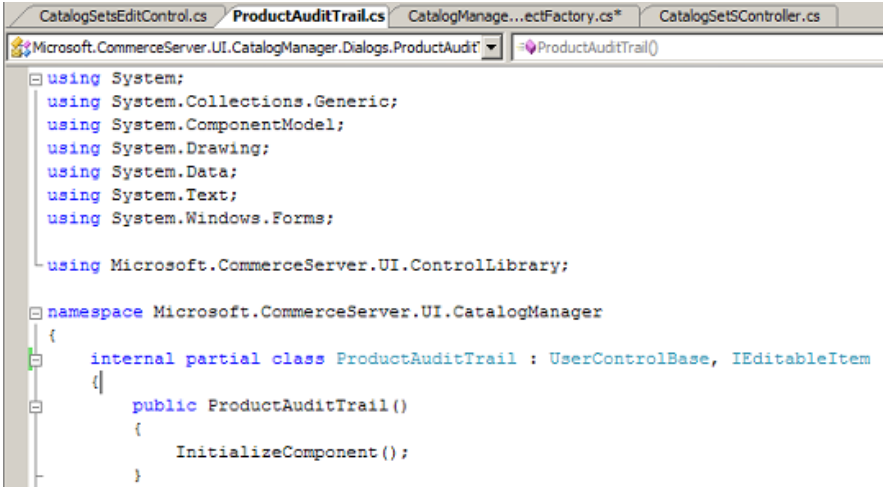
Tasks	Detailed Steps
	<p>from the <b>Tools Window</b>:</p> <pre data-bbox="542 233 1398 516">CREATE PROCEDURE sp_SelectCatalogProductAudit (@ProdID nvarchar(256)) AS SELECT * FROM [CSHarpSite_productcatalog].[dbo].[CatalogProductAudit] WHERE ProductID = @ProdID</pre> <p>e. Set permissions for the above stored procedures so that the Catalog Web Service account can execute them. From the SQL Server Management Studio navigate to <b>Databases   CSharpSite_ProductCatalog   Programmability   Stored Procedures</b> then find <b>sp_SelectCatalogProductAudit</b> then right click and select <b>Properties</b> in the <b>Stored Procedure Properties</b> window select <b>Permissions</b>. Then select <b>Add</b> from the <b>Select User or Roles</b> select the <b>Browse</b> button and check the <b>csCatalogWebSvc</b> role next select <b>OK</b> and from the <b>Select User or Roles</b> select <b>OK</b>. From the <b>Stored Procedure Properties</b> window grant execute permission. Click <b>OK</b>.</p> <p>f. Follow step e above for <b>sp_InsertCatalogProductAudit</b> stored procedure.</p> <p>g. Close the <b>SQL Server Enterprise Manager</b> console if prompted to save any files select <b>NO</b>.</p>
<p>3. Extend Catalog Web Service</p>	<p>a. Select <b>Start   All Programs   Microsoft Visual Studio 2005</b> and select <b>Microsoft Visual Studio 2005</b></p> <p>b. Select <b>File   Open   Web Site...</b></p> <p>c. From the <b>Open Web Site</b> screen navigate to <b>Local IIS</b> and select <b>CatalogWebService</b> and select <b>Open</b>.</p> <p>d. From the <b>Solution Explorer</b> select <b>CatalogWebService</b> then right click and select <b>Add References...</b></p> <p>e. From <b>Add Reference</b> screen click the <b>Browse</b> tab and navigate to <b>C:\Program Files\Common Files\Microsoft Shared\Enterprise Servers\Commerce Server\Assemblies</b> and select <b>Microsoft.CommerceServer.Catalog.dll</b> and select <b>OK</b>.</p> <p>f. Follow <b>Step d</b> then From <b>Add Reference</b> screen click the <b>Browse</b> tab and navigate to <b>C:\Program Files\Microsoft Commerce Server 2007\Assemblies</b> and select <b>Microsoft.CatalogServer.dll</b> then select <b>OK</b>.</p> <p>g. Follow <b>Step d</b> then From <b>Add Reference</b> screen click the <b>Browse</b> tab and navigate to <b>C:\Program Files\Microsoft Commerce Server 2007\Assemblies</b> and select <b>Microsoft.CommerceServer.Interop.Configuration.MSCSCfg.dll</b> then select <b>OK</b>.</p> <p>h. Follow <b>Step d</b> then From <b>Add Reference</b> screen click the <b>.NET</b> tab and select <b>adodb</b> and click <b>OK</b>.</p> <p>i. Rename <b>CatalogWebService.asmx</b> file to <b>CatalogWebService.old.asmx</b>.</p> <p>j. From the <b>Solution Explorer</b> select <b>CatalogWebService</b> then right click and select <b>Add New Item...</b> From the <b>Add New Item</b> screen select <b>Web Service</b> and for file name type "<b>CatalogWebService.asmx</b>".</p> <p>k. By adding the new Web Service Visual Studio will open the <b>CatalogWebService.cs</b> if not then from the <b>Solution Explorer</b> navigate to <b>App_Code</b> and open <b>CatalogWebService.cs</b> file.</p> <p>l. In source code, replace the Namespace <b>http://tempuri.org/</b> with</p>

Tasks	Detailed Steps
	<p><b>http://schemas.microsoft.com/CommerceServer/2006/06/CatalogWebService.</b></p> <p><b>m.</b> Replace the <b>CatalogWebService</b> inherited value <b>System.Web.Services.WebService</b> with <b>Microsoft.CommerceServer.Catalog.WebService.CatalogWebService.</b></p> <p><b>n.</b> From the <b>CatalogWebService</b> constructor call the base method of <b>CalogWebService.</b></p> <p><b>o.</b> Add the following using statements:</p> <pre data-bbox="545 489 1398 611">using Microsoft.CommerceServer.Catalog.XmlData; using Microsoft.CommerceServer.Interop.Configuration; using System.Data.SqlClient; using ADODB;</pre> <p><b>p.</b> Remove the <b>HelloWorld WebMethod.</b></p> <p><i>Note: The CatalogWebService source should appear as follows:</i></p> <pre data-bbox="545 737 1398 1528">using System; using System.Web; using System.Collections; using System.Web.Services; using System.Web.Services.Protocols;  using ADODB; using System.Data.SqlClient; using Microsoft.CommerceServer.Catalog.XmlData; using Microsoft.CommerceServer.Interop.Configuration;  /// &lt;summary&gt; /// Summary description for CatalogWebService /// &lt;/summary&gt; [WebService(Namespace = "http://schemas.microsoft.com/CommerceServer/2006/06/Catal ogWebService")] [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)] public class CatalogWebService : Microsoft.CommerceServer.Catalog.WebService.CatalogWebServ ice {     public CatalogWebService() : base() { } } </pre> <p><b>q.</b> Override the <b>SaveMethod</b> from the base <b>CatalogWebService</b> and add the audit information to the database:</p> <pre data-bbox="545 1644 1398 1892">[WebMethod] public override Product SaveProduct(Product updatedProduct, bool forceUpdate) {     Product prod = base.SaveProduct(updatedProduct, forceUpdate);     SiteConfigFreeThreaded siteCfg = new SiteConfigFreeThreaded(); } </pre>

Tasks	Detailed Steps
	<pre> siteCfg.Initialize("CSharpSite"); Recordset ctlg = (Recordset)siteCfg.Fields["Product Catalog"].Value; string dbCon = ctlg.Fields["connstr_db_Catalog"].Value.ToString(); System.IO.StringWriter strWriter = new System.IO.StringWriter(); prod.Information.Tables[0].WriteXml(strWriter); string prodAudit = strWriter.ToString(); string username = User.Identity.Name; SqlConnection con = new SqlConnection(dbCon.Replace("Provider=SQLOLEDB.1;", string.Empty)); con.Open(); SqlCommand cmd = con.CreateCommand(); cmd.CommandText = "sp_InsertCatalogProductAudit"; cmd.CommandType = System.Data.CommandType.StoredProcedure; cmd.Parameters.Add(new SqlParameter("@ProdID", prod.ProductId)); cmd.Parameters.Add(new SqlParameter("@Audit", prodAudit)); cmd.Parameters.Add(new SqlParameter("@UserName", username)); cmd.Parameters.Add(new SqlParameter("@Date", DateTime.Now)); cmd.ExecuteNonQuery(); ctlg.Close(); con.Close();  return prod; } </pre> <p>r. Create a new Web Method <b>GetProductAudit</b> and retrieve all product changes only if the user is authorized to perform task:</p> <pre> [WebMethod] public System.Data.DataSet GetProductAudit(string ProductId) {     if (this.GetAuthorizedTasks().Tables[0].Select("task='GetProd uctAuditTrail'").Length != 0)     {         //Retrieve connection string from Site Configuration         SiteConfigFreeThreaded siteCfg = new SiteConfigFreeThreaded();         siteCfg.Initialize("CSharpSite");         Recordset ctlg = (Recordset)siteCfg.Fields["Product Catalog"].Value;         string dbCon = ctlg.Fields["connstr_db_Catalog"].Value.ToString();         SqlConnection con = new SqlConnection(dbCon.Replace("Provider=SQLOLEDB.1;", </pre>

Tasks	Detailed Steps
	<pre data-bbox="545 197 1393 856"> string.Empty));     con.Open();      SqlCommand cmd = con.CreateCommand();     cmd.CommandText = "sp_SelectCatalogProductAudit";     cmd.CommandType = System.Data.CommandType.StoredProcedure;     cmd.Parameters.Add(new SqlParameter("@ProdID", ProductId));     SqlDataAdapter da = new SqlDataAdapter(cmd);      System.Data.DataSet ds = new System.Data.DataSet();     da.Fill(ds);     ctlg.Close();     return ds; } else {     return null; } }                 </pre> <p data-bbox="506 865 1422 919">s. Close the <b>CatalogWebService Solution</b> when prompted to save select <b>Yes</b> accept defaults and select <b>Save</b>.</p>
<p data-bbox="188 940 461 999">4. Extend Authorization Manager</p>	<ol style="list-style-type: none"> <li data-bbox="506 940 753 966">a. Select <b>Start   Run</b>.</li> <li data-bbox="506 982 1422 1037">b. From <b>Run</b> screen type “<b>azman.msc</b>” and select <b>Open</b>. If you encounter any errors select <b>OK</b> and continue.</li> <li data-bbox="506 1054 1396 1108">c. From the <b>Authorization Manager</b> console right click <b>Authorization Manager</b> node and select <b>Options</b>.</li> <li data-bbox="506 1125 1156 1150">d. From the <b>Option</b> dialog select <b>Developer</b> and select <b>OK</b>.</li> <li data-bbox="506 1167 1390 1222">e. Select <b>Authorization Manager</b> and right click and select <b>Open Authorization Store</b>.</li> <li data-bbox="506 1239 1338 1327">f. From the <b>Open Authorization Store</b> dialog select <b>Browse</b> and navigate to <b>C:\Inetpub\wwwroot\CatalogWebService</b> and select <b>CatalogAuthorizationStore.xml</b> and then select <b>Open</b>.</li> <li data-bbox="506 1344 1130 1369">g. Select <b>OK</b> from the <b>Open Authorization Store</b> dialog.</li> <li data-bbox="506 1386 1396 1474">h. From the <b>Authorization Manager</b> console navigate to <b>CatalogAuthorizationStore.xml   CatalogAndInventorySystem   Definitions</b> and select <b>Operation Definitions</b>.</li> <li data-bbox="506 1491 1409 1680">i. Right click on <b>Operation Definitions</b> and select <b>New Operation Definition</b> and do the following: <ul style="list-style-type: none"> <li data-bbox="643 1554 1071 1579">• In <b>Name</b> type “<b>GetProductAudit</b>”</li> <li data-bbox="643 1596 1195 1621">• In <b>Description</b> type “<b>Get product audit trail</b>”</li> <li data-bbox="643 1638 1240 1663">• In <b>Operation Number</b> type “<b>1000</b>” and select <b>OK</b>.</li> </ul> </li> <li data-bbox="506 1696 1357 1848">j. Right click on <b>Task Definitions</b> and select <b>New Task Definition</b> and do the following: <ul style="list-style-type: none"> <li data-bbox="643 1764 1130 1789">• In <b>Name</b> type “<b>GetProductAuditTrail</b>”</li> <li data-bbox="643 1806 1195 1831">• In <b>Description</b> type “<b>Get product audit trail</b>”</li> </ul> </li> <li data-bbox="506 1864 1370 1890">k. Select <b>Add</b> button and from the <b>Add Definition</b> dialog select <b>Operation</b> and</li> </ol>

Tasks	Detailed Steps
	<p>select <b>GetProductAudit</b>, then select <b>OK</b> and select <b>OK</b> from <b>New Task Definition</b>.</p> <p><b>l.</b> Select <b>Role Definitions</b> then select <b>Catalog Administrator</b>.</p> <p><b>m.</b> Right click and select <b>Properties</b> select <b>Definitions</b> tab then select <b>Add</b> button from the <b>Add Definitions</b> dialog select <b>Tasks</b> tab then select <b>GetProductAuditTrail</b> then select <b>OK</b> and from <b>Catalog Administrator Definition Properties</b> select <b>OK</b>.</p> <p><b>n.</b> Close the <b>Authorization Manager</b> console.</p>
<p><b>5.</b> Extend Catalog Manager</p>	<p><b>a.</b> Select <b>Start   All Programs   Microsoft Visual Studio 2005</b> and select <b>Microsoft Visual Studio 2005</b></p> <p><b>b.</b> Select <b>File   New   Project</b>.</p> <p><b>c.</b> Under the <b>Project Type</b> Select <b>Visual C#</b>.</p> <p><b>d.</b> From the <b>New Projects Templates</b> pane select <b>Class Library</b></p> <p><b>e.</b> For project <b>Name</b> type “<b>CustomWebServiceProxy</b>” and select <b>OK</b>.</p> <p><b>f.</b> From <b>Solutions Explorer</b> right click on <b>CustomWebServiceProxy</b> project and select <b>Add Web Reference</b>.</p> <p><b>g.</b> From the <b>Add Web Reference</b> screen type “<b>http://localhost/CatalogWebService/CatalogWebService.asmx</b>” for <b>URL</b> then select <b>GO</b>.</p> <p><b>h.</b> When the <b>Web Service</b> is found select <b>Add Reference</b>.</p> <p><b>i.</b> Right click <b>CustomWebServiceProxy</b> and select <b>Properties</b>.</p> <p><b>j.</b> From the <b>CustomWebServiceProxy</b> properties window select signing the check the <b>sign the assembly</b>.</p> <p><b>k.</b> From the <b>Choose a strong name key file</b> select <b>new</b>.</p> <p><b>l.</b> From the <b>Create Strong Name Key</b> dialog type “<b>CustomWebServiceProxy</b>” for the <b>key file name</b> and unselect <b>Protect my key file with a password</b>. Click <b>OK</b></p> <p><b>m.</b> From the <b>Solutions Explorer</b> right click on <b>CustomWebServiceProxy</b> project and select <b>Build</b>.</p> <p><b>n.</b> Close the <b>Visual Studio 2005</b>.</p> <p><b>o.</b> Navigate to <b>C:\Program Files\Microsoft Commerce Server 2006\sdk\Partner SDK</b> and open <b>Commerce Server 2006 Partner SDK.sln</b></p> <p><b>p.</b> From <b>Solutions Explorer</b> right click on <b>CatalogManager</b> and select <b>Add Reference</b>.</p> <p><b>q.</b> From the <b>Add Reference</b> dialog select the <b>Browse</b> tab and navigate to <b>C:\Documents and Settings\Administrator\My Documents\Visual Studio 2005\Projects\CustomWebServiceProxy\CustomWebServiceProxy\bin\Debug</b> and select <b>CustomWebServiceProxy.dll</b> then select <b>OK</b>.</p> <p><b>r.</b> Open the <b>LaunchPad.cs</b> file and at the <b>Private variables</b> region type “<b>private static CustomWebServiceProxy.localhost.CatalogWebService currentCustomCatalogContext = null;</b>”.</p> <p><b>s.</b> In <b>Public Properties</b> region type the following code:</p> <pre data-bbox="542 1724 1398 1906"> public static CustomWebServiceProxy.localhost.CatalogWebService CurrentCustomCatalogContext {     get     { </pre>

Tasks	Detailed Steps
	<pre data-bbox="545 195 1398 506">             if (currentCustomCatalogContext == null)             {                 currentCustomCatalogContext = new                 CustomWebServiceProxy localhost.CatalogWebService();                 currentCustomCatalogContext.Credentials =                 CredentialCache.DefaultCredentials;             }             return currentCustomCatalogContext;         }     }     </pre> <p data-bbox="506 514 1398 573">t. From the <b>Solution Explorer</b> navigate to <b>Dialog</b> folder and right click and select <b>New Item</b>.</p> <p data-bbox="506 583 1398 642">u. From the <b>Add New Item</b> dialog select <b>User Control</b> and for name type <b>“ProductAuditTrail.cs”</b>.</p> <p data-bbox="506 653 1398 682">v. From the <b>ProductAuditTrail</b> design view right click and select <b>View Code</b>.</p> <p data-bbox="506 693 1398 751">w. Type <b>“using Microsoft.CommerceServer.UI.ControlLibrary;”</b> in the using keyword section.</p> <p data-bbox="506 762 1398 821">x. Replace the <b>ProductAuditTrail’s</b> <b>UserContorl</b> inherited object with <b>UserControlBase</b> and <b>IEEditableItem</b>.</p> <p data-bbox="506 831 1398 890">y. Rename the name space to the following <b>“namespace Microsoft.CommerceServer.UI.CatalogManager”</b>.</p> <p data-bbox="506 900 1398 930">z. Make the <b>ProductAuditTrail</b> class <b>internal</b>.</p> <p data-bbox="506 940 1398 970"><i>Note: The control should look like the following:</i></p>  <p data-bbox="506 1518 1398 1577">aa. Implement all the <b>IEEditableItem</b> interface on the <b>ProductAuditTrail</b> class as follows:</p> <pre data-bbox="545 1633 1398 1883"> #region IEEditableItem Members  public void Initialize(CatalogManagerController controller, bool headerVisible) {     this.productAuditName = controller.Object.Name;     dataGridView1.DataSource =     LaunchPad.CurrentCustomCatalogContext.GetProductAudit(pr             </pre>

Tasks	Detailed Steps
	<pre data-bbox="542 195 1393 1014"> productAuditName).Tables[0]; }  public bool Commit(){ return true;}  public void DisplayItem(bool objectChanged){}  public bool ShowItem() {     if(LaunchPad.CurrentCatalogContext.GetAuthorizedTasks(). Tables[0].Select("task='GetProductAuditTrail']").Length != 0)     {         return true;     }     else     {         return false;     } }  public string HelpString{get { return string.Empty; }}  public event ControlValueChanged ControlValueChanged; private string productAuditName; #endregion </pre> <p data-bbox="506 1024 1425 1150"><b>bb.</b> From the <b>Solution Explorer</b> navigate to ProductAuditTrial.cs and expand the plus sign to open the <b>“ProductAuditTrail.Designer.cs”</b>. Next open the file <b>“ProductAuditTrail.Designer.cs”</b> and Rename the name space to the following <b>“namespace Microsoft.CommerceServer.UI.CatalogManager”</b>.</p> <p data-bbox="506 1161 954 1192"><b>cc.</b> Rebuild the <b>CatalogManager</b> project.</p> <p data-bbox="506 1203 784 1234"><i>Note: Ignore build errors</i></p> <p data-bbox="506 1245 1239 1276"><b>dd.</b> Right click on <b>ProductAuditTrail.cs</b> and select <b>View Designer</b>.</p> <p data-bbox="506 1287 1271 1318"><b>ee.</b> From the <b>ProductAuditTrail.cs</b> design view add a <b>DataGridView</b>.</p> <p data-bbox="506 1329 873 1360"><i>Note: Size the control a bit larger</i></p> <p data-bbox="506 1371 1304 1423"><b>ff.</b> From the <b>Solutions Explorer</b> navigate to <b>Controllers</b> folder and open <b>CatalogManagerObjectFactory.cs</b>.</p> <p data-bbox="506 1434 1393 1497"><b>gg.</b> From the <b>CatalogManagerObjectFactory.cs</b> page type <b>“ProductAuditTrail”</b> enum in <b>CatalogManagerItem</b>.</p> <p data-bbox="506 1507 1003 1539"><i>Note: The code should look like the following:</i></p>

Tasks	Detailed Steps
	<div data-bbox="597 193 1393 613" style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <pre> namespace Microsoft.CommerceServer.UI.CatalogManager {     internal enum CatalogManagerItem     {         ProductAuditTrail,         EditProductCatalogLanguages,         EditProductCatalogMainProperties,         EditProductCatalogCustomProperties,         EditProductCatalogDependentCatalogs,         EditProductInventoryCatalogs,         EditVirtualCatalogRules,     } }                     </pre> </div> <p data-bbox="506 625 1393 718"><b>hh.</b> From the Project Explorer navigate to <b>CatalogManager   Wizards</b> and open <b>InventoryCatalogController.cs</b> file and under the <b>OperationMode</b> enum add type the following “<b>ViewProductAuditTrail=2</b>”.</p> <p data-bbox="506 726 1003 756"><i>Note: The code should look like the following:</i></p> <pre data-bbox="620 764 1214 1558"> internal enum OperationMode {     Create = 0,     Edit = 1,     Delete = 2,     Refresh = 3,     ImportInventoryCatalog = 4,     ImportProductCatalog = 5,     ExportInventoryCatalog = 6,     ExportProductCatalog = 7,     CreateBaseCatalog = 8,     CreateVirtualCatalog = 9,     EditBaseCatalog = 10,     EditVirtualCatalog = 11,     CreateCategory = 12,     CreateProduct = 13,     EditCategory = 14,     EditProduct = 15,     AddVirtualCatalogItem = 16,     CreateStaticCatalogSet = 17,     CreateExpressionBasedCatalogSet = 18,     EditStaticCatalogSet = 19,     EditExpressionBasedCatalogSet = 20 ,     CopyProduct = 21,     ViewProductAuditTrail = 22 }                     </pre> <p data-bbox="506 1570 1370 1663"><b>ii.</b> From the Project Explorer navigate to <b>CatalogManager   Dialogs</b> and open <b>CatalogItemEdit.cs</b> file and under <b>CatalogItemEdit_Load</b> method type the following code:</p> <pre data-bbox="545 1713 1370 1898"> if (CatalogItemController.CatalogItemObject.OperationMode == OperationMode.EditProduct) {     pages.Add(CatalogManagerItem.ProductAuditTrail, "Product Audit Trail"); }                     </pre>

Tasks	Detailed Steps
	<p><i>Note: the code should look like the following:</i></p> <pre data-bbox="553 275 1409 426"> if (CatalogItemController.CatalogItemObject.OperationMode == OperationMode.Edit) {     pages.Add(CatalogManagerItem.ProductAuditTrail, "Product Audit Trail"); } Initialize(pages); if ( this.CatalogItemController.CatalogItemObject.OperationMode == OperationMode.CreateProduct    this.CatalogItemController.CatalogItemObject.OperationMode == OperationMode.EditProduct    this.CatalogItemController.CatalogItemObject.OperationMode == OperationMode.CopyProduct ) </pre> <p><b>jj.</b> Add another <b>enum</b> under Operation called <b>productAuditTrail</b>.</p> <p><b>kk.</b> From the <b>CatalogManagerObjectFactory.cs</b> page in <b>CatalogManagerObjectFactory</b> class's create method type the following code:</p> <pre data-bbox="544 583 1398 678"> case CatalogManagerItem.ProductAuditTrail:     objectToCreate = new ProductAuditTrail();     break; </pre> <p><i>Note: The code should look like the following:</i></p> <pre data-bbox="553 764 1409 1026"> internal static IEditableItem Create(CatalogManagerItem itemToC: {     IEditableItem objectToCreate = null;     switch (itemToCreate)     {         case CatalogManagerItem.ProductAuditTrail:             objectToCreate = new ProductAuditTrail();             break;          case CatalogManagerItem.EditProductCatalogLanguages: </pre> <p><b>ll.</b> From the <b>Project Explorer</b> right click and select <b>CatalogManager</b> next select <b>Properties</b>.</p> <p><b>mm.</b> From the <b>Properties</b> window select the <b>Build Event</b> Tab then modify the <b>Post-build command line</b> to the following <b>COPY</b> <b>"\$(ProjectDir)CatalogManagerConfig.xml" "\$(TargetDir)"</b> and close the Properties window.</p> <p><b>nn.</b> Build the solution by select the <b>F5</b> function key.</p>
<p><b>6.</b> Verify Audit changes</p>	<p><b>a.</b> After the solution is built and the <b>Catalog Manager</b> screen is displayed, select <b>Adventure Works Catalog</b>.</p> <p><b>b.</b> In <b>Category and Product Details</b> pane select product <b>AW009-08</b> and right click then select <b>Edit This Product</b>.</p> <p><b>c.</b> Form the Product Edit screen type <b>“Straight specials for auditing”</b> in <b>Description</b> and select <b>Save and Close</b>.</p> <p><b>d.</b> In <b>Category and Product Details</b> pane select product <b>AW009-08</b> and right click then select <b>Edit Product Properties</b>.</p> <p><b>e.</b> From the Product Edit screen select Product Audit Trail the record in previous step that was modified is now recorded.</p> <p><b>f.</b> Close the application and Visual Studio.</p>

## Lab Summary

Commerce Server's Partner SDK accelerates and automates the development efforts and reduces operational\development costs.

In this lab, you have created a custom table to log the audit trail for product changes, you have extended the catalog Web Service to log and retrieve product audits, you have extended the Authorization Manager to allow Catalog Administrators to view the audit trail, and you have modified Catalog Manager to view the audit logs.

- Extend Catalog Manager UI
- Extend Catalog Web Service
- Extend Authorization Manger roles