# Lab 6: Developing SharePoint Workflow Templates with Visual Studio 2008

**Lab Overview:** Now that the management of Litware Inc. has become comfortable with the built in functionality of SharePoint workflows, they have started to request extra features. They have requested a workflow system that archives documents in another document library. When the document library manager chooses to archive a document, they manually start a workflow, which creates an approval task.

To build this workflow, you will need to deal with many different aspects of SharePoint workflow, so you will start simple. In the first part or implementing the **DocumentArchive** project, you will focus on creating the workflow that will simply log a message to the Workflow History list.

## Exercise 0: Setup

### If you did not complete lab 4, you will need to create the **Demo** site collection.

#### Open SharePoint and browse to the Demo site collection

##### The url is **http://litwareinc.com/sites/Demo**.

##### If the site collection does not exist, create it using the **CreateDemo.bat** file in the **C:\Labs\Files folder**.

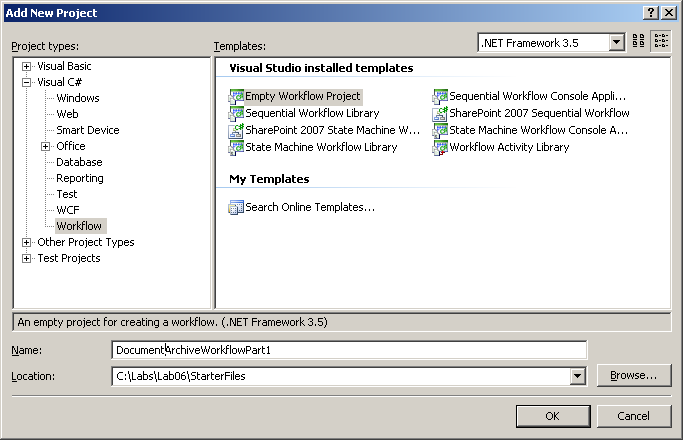
### Open the starter **VS 2008** solution at **\Labs\Lab06\StarterFiles\DocumentArchive Part 1.sln**.

## Exercise 1: Build Hello World Workflow

### Add a new **Empty Workflow Project** named **DocumentArchiveWorkflowPart1**.

#### Right click on the solution in the **Solution Explorer** and select **Add -> New Project**.

#### In the **Add New Project** dialog, select the project type of **Workflow** on the left and a template of **Empty Workflow Project** on the right.



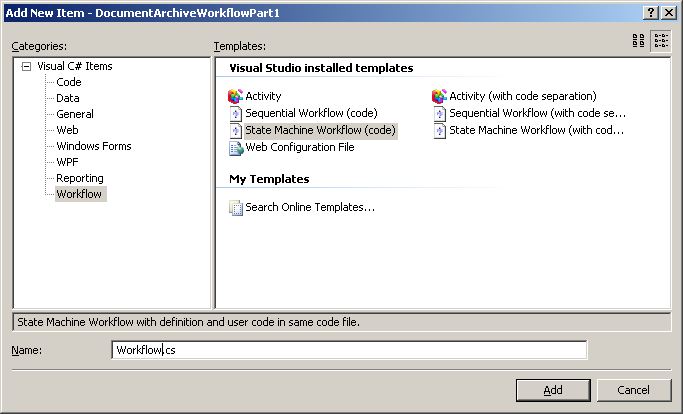
#### Name the project **DocumentArchiveWorkflowPart1** and click **OK**.

### Add a new state machine workflow named **Workflow** to the project.

#### Right click on the **DocumentArchiveWorkflowPart1** project in the solution explorer and select **Add -> New Item…**

#### In the **Add New Item** dialog, select the **Workflow** category on the left and a template of **State Machine Workflow (code)** on the right.

#### Name the new item **Workflow.cs** and click **Add**.

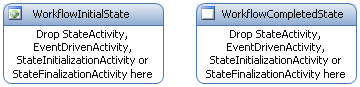


### Add a new state to the workflow named **WorkflowCompletedState**.

#### Right click on the **State Machine** designer canvas and click **Add State**.

#### Set the name of the new state to **WorkflowCompletedState**.

##### In the properties pane, set the **Name** property to **WorkflowCompletedState**.

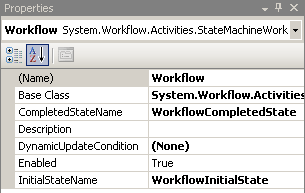


### Set the initial and completed states of the workflow

#### Right click on the **State Machine** designer canvas and click **Properties**.

#### Verify the **InitalStateName** property in the Properties pane is set to **WorkflowInitialState**.

#### Set the **CompletedStateName** property to **WorkflowCompletedState**.



### Add an **OnWorkflowActivated** event driven activity to signal the start of the workflow.

#### Right click on the **WorkflowInitialState** state and select **Add** **EventDriven**.

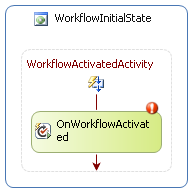
#### Rename the new event driven activity **WorkflowActivatedActivity**.

##### Right click the new event driven activity and click **Properties**.

##### In the properties pane, set the **Name** property to **WorkflowActivatedActivity**.

#### Drag an **OnWorkflowActivated** activity from the toolbox into the **WorklowActivatedActivity** activity and name it **OnWorkflowActivated**.

##### In the properties pane, set the **Name** property to **OnWorkflowActivated**.

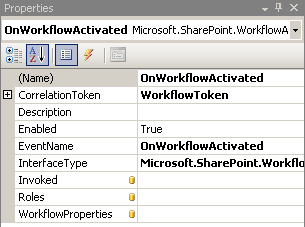


#### Create a new correlation token for the **OnWorkflowActivated** activity. Scope it to the workflow.

##### In the properties pane, set the **CorrelationToken** property to **WorkflowToken**.

##### Expand the **CorrelationToken** property in the property pane.

##### Set the **OwnerActivityName** property to **Workflow**.



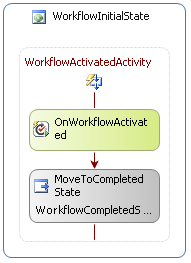
### Transition to the **WorkflowCompletedState** state after the **OnWorkflowActivated** activity executes.

#### Drag a **SetState** activity from the toolbox into the **WorkflowActivatedActivity** immediately following the **OnWorkflowActivated** activity.

##### Make sure you use the **SetState** activity in the **Windows Workflow v3.0** group in the toolbox.

#### Change the new activity’s name to **MoveToCompletedState** and set the target state to **WorkflowCompletedState**.

##### Set the **TargetStateName** property to **WorkflowCompletedState**.



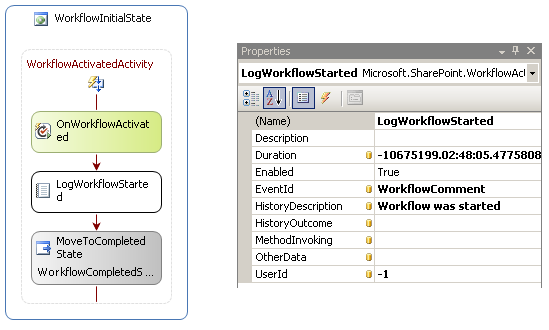
### Log a started message to the workflow’s history to indicate that the workflow ran.

#### Drag a **LogToHistoryListActivity** from the toolbox into the **WorkflowActivatedActivity** between the **OnWorkflowActivated** and **MoveToCompletedState** activities.

#### Change the new activity’s name to **LogWorkflowStarted** and set its **HistoryDescription** to **‘Workflow was started’**.

##### In the properties pane, set the **Name** property to **LogWorkflowStarted**.

##### Set the **HistoryDescription** property to **Workflow was started**.



### Build the project.

#### Right click the project in the **Solution** **Explorer** and click **Build**.

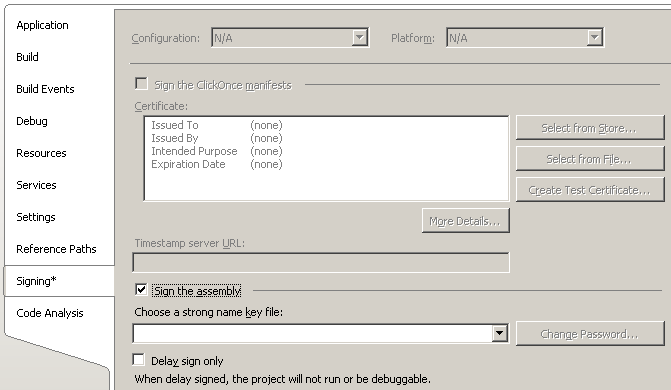
## Exercise 2: Install the workflow as a Feature

### Sign the assembly to allow installation in the GAC.

#### Right click the project in the solution explorer and click **Properties**.

#### Click the **Signing** tab on the left hand side of the project properties window.

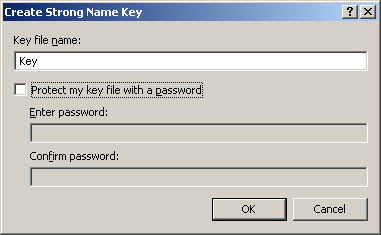
#### Check the **Sign the assembly** checkbox.



#### Choose **<New …>** new from the drop down list box.

#### In the **Create Strong Name Key** dialog box, enter a **Key file name** of **Key**.

#### Uncheck the **Protect my key file with a password** and click **OK**.



#### Close the project properties window.

### Create the folder that will contain the feature files.

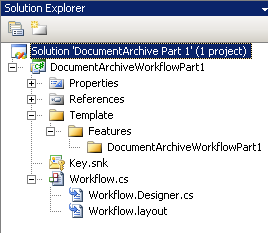
#### Create a folder under the project named **Template**.

##### Right click on the project in the **Solution Explorer** and click **Add -> New Folder**.

##### Right click the folder and click **Rename** to rename the folder to **Template**.

#### Create another folder named under **Template** named **Features**.

#### Create a third folder under **Features** named **DocumentArchiveWorkflowPart1**.



### Create the **feature.xml** in the **DocumentArchiveWorkflowPart1** folder.

#### Create a new file named **Feature.xml** in the **DocumentArchiveWorkflowPart1** folder.

##### Right click the **DocumentArchiveWorkflowPart1** folder and click **Add -> New Item…**.

##### Select the **Data** category on the left and select a template of **XML file**.

##### Name the new file **Feature.xml** and click **Add**.

#### Insert the **Windows SharePoint Service Workflow Feature.xml** snippet.

##### Right click in the editor and select **Insert Snippet…**.

##### In the popup window, select the **Windows SharePoint Service Workflow** option then select **Feature.xml Code**.

#### Replace **GUID** with a new Guid generated using **GuidGen**.

##### Click **Tools -> Create Guid** then copy and paste the registry formatted guid into the snippet’s GUID placeholder.

#### Replace **Title** with **Document Archive – Part 1**.

#### Replace **Description** with **“Contains workflows and custom forms designed to aid in document archiving”**.

<Feature Id="29ac22d1-7d8c-4513-9566-11109ab25983"

Title="Document Archive - Part 1"

Description="Request approval to archive this document in another document library."

Version="12.0.0.0"

Scope="Site"

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

<ElementManifests>

<ElementManifest Location="workflow.xml" />

</ElementManifests>

<Properties>

<Property Key="GloballyAvailable" Value="true" />

</Properties>

</Feature>

#### Save and close the file.

### Create the **workflow.xml** file needed to define the details of the workflow.

#### Create a new file named Workflow.xml in the **DocumentArchiveWorkflowPart1** folder.

#### Insert the **Windows SharePoint Service Workflow Workflow.xml** snippet.

##### Right click in the editor and select **Insert Snippet…**.

##### In the popup window, select the **Windows SharePoint Service Workflow** option.

##### In the second popup window, select **Workflow.xml Code**.

#### Replace **GUID** with a new Guid generated using GuidGen.

##### Click **Tools -> Create Guid**

##### Copy and paste the registry formatted guid into the snippet’s GUID placeholder.

#### Replace **Title** with **Document Archive – Part 1**.

#### Replace **Description** with **“Requests approval to archive this document in another document library”**.

#### Replace **CodeBesideClass** with **DocumentArchiveWorkflowPart1.Workflow**.

#### Replace the **PublicKeyToken** of the **CodeBesideAssembly** with the public key token of the assembly.

##### Open a new instance of the **Visual Studio 2008 Command Prompt** located in **Start -> All Programs -> Microsoft Visual Studio 2008 -> Visual Studio Tools**.

##### **Navigate to C:\Labs\Lab06\StarterFiles\DocumentArchiveWorkflowPart1\bin\Debug**

##### Run **sn.exe –T DocumentArchiveWorkflowPart1.dll** to display the public key token.

#### Remove the **TaskListContentTypeId**, **AssociationUrl**, **InstantiationUrl**, **ModificationUrl** attributes.

##### These will be used later.

#### Remove the content of the **MetaData** element.

##### These will also be used later.

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

<Workflow

Name="Document Archive - Part 1"

Description="Requests approval to archive this document in another document library."

Id="78c8fd78-0b7f-4893-9361-ff005c32b12c"

CodeBesideClass="DocumentArchiveWorkflowPart1.Workflow"

CodeBesideAssembly="DocumentArchiveWorkflowPart1, Version=1.0.0.0, Culture=neutral, PublicKeyToken=15812f954569663f"

StatusUrl="\_layouts/WrkStat.aspx">

<Categories/>

<MetaData>

</MetaData>

</Workflow>

</Elements>

### Add post build steps to aid in developing and deploying the workflow.

#### Right click the project in the **Solution Explorer** and click **Properties**.

#### Click the **Build Events** tab in the project properties window.

#### Enter the following commands into the **Post-build event command line** window.

##### Copy the contents of the **Template** folder to SharePoint’s **Template** folder.

##### Install the workflow assembly into the GAC using **gacutil.exe**.

##### Reset the app domain that may have loaded the workflow assembly.

##### Forcibly install the SharePoint feature using **stsadm.exe**.

xcopy "$(ProjectDir)\TEMPLATE" "C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE" /E /Y

"$(DevEnvDir)\..\..\SDK\v2.0\bin\gacutil.exe" /i "$(TargetPath)" /f

%windir%\system32\cscript.exe c:\windows\system32\iisapp.vbs /a "SharePointDefaultAppPool" /r

"C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\bin\stsadm.exe" -o installfeature -name DocumentArchiveWorkflowPart1 –force

#### Save and close the project properties window.

### Rebuild and deploy the new workflow feature.

#### Right click the project in the solution explorer and click **Rebuild**.

##### If no code changes are made, **Visual Studio 2008** does not re-execute the post build events. If you need to force a redeploy do a rebuild.

#### Look at the **Output** window to determine the post build steps executed successfully.

##### If the **Output** window is not visible, display it using **View -> Output**.

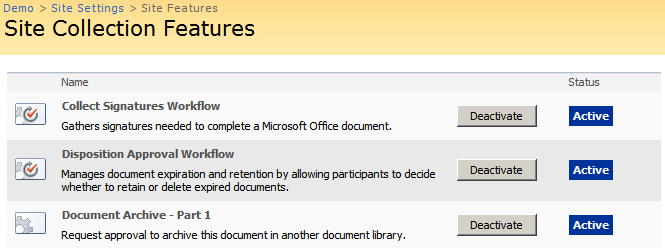
### Activate the new **DocumentArchiveWorkflow** feature.

#### Using **Internet Explorer** navigate to the **Demo** site collection at **http://litwareinc.com/sites/Demo**.

#### Open the features list by clicking **Site Actions -> Site Settings**.

#### On the **Site Settings** page, click **Site collection features** in the **Site** **Collection Administration** section.

#### Click the **Activate** button next to the **Document Archive – Part 1** feature.



### Create an association between the **Shared Documents** document library and the new **Document Archive** **– Part 1** workflow.

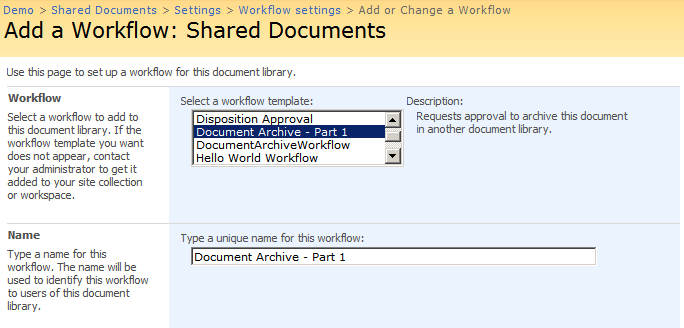
#### Navigate to the **Shared Documents** document library in the **Demos** site.

#### Click **Settings -> Document Library Settings** to load the settings page.

#### Click the **Workflow settings** link in the **Permissions and Management** section.

#### Create a new workflow using the **Document Archive – Part 1** workflow template and a name of **Document Archive - Part 1**.

##### Use the default values for both list and startup options.



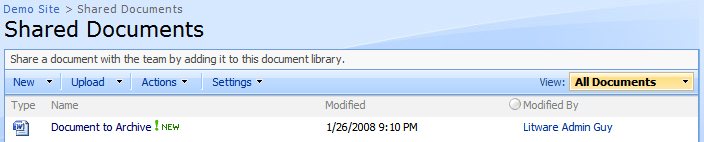
### Add a new document to the **Shared Documents** folder.

#### Navigate to the **Shared Documents** document library in the **Demo** site.

#### Click **New -> New document** to create a new document.

#### Click **OK** to allow Word 2007 to open the file.

#### Enter some data and save the document as **Document To Archive.docx**.



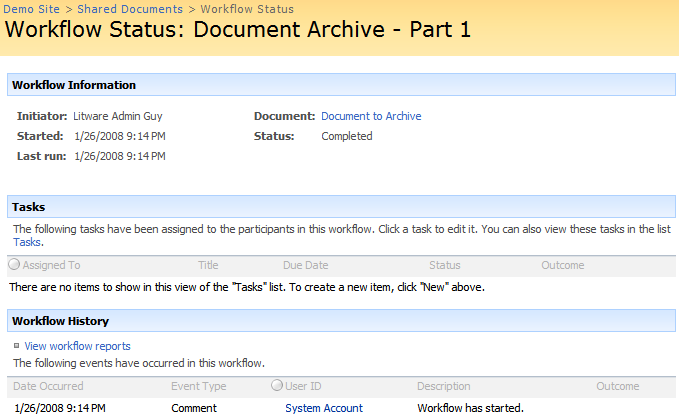
### Run the **Document Archive – Part 1** workflow on the newly created document.

#### Hover over the new document and select **Workflows** from the drop down menu.

#### In the workflows page, click the **Document Archive – Part 1** to start the workflow.

#### In the **Shared Documents** document library, verify the workflow has completed.

#### Click the **Completed** link to view the workflow status and verify the started message was logged to the workflow’s history.



## Challenge: Deploy the workflow as a Solution Package

### Create the folder to store the **Solution Package** definition files.

#### Create a folder under the project named **Solution**.

##### Right click on the project in the **Solution Explorer** and click **Add -> New Folder**.

##### Right click the folder and click **Rename**.

##### Rename the folder **Solution**.

### Create the **Manifest.xml** in the **Solution** folder.

#### Right click the **Solution** folder and click **Add -> New Item…**.

#### Select the **Data** category on the left and select a template of **XML file**.

#### Name the new file **Manifest.xml** and click **Add**.

### Paste the following XML into your manifest file.

#### The **FeatureManifest** element defines the features that are part of this solution.

#### The **Assembly** elements define the target location for any assemblies that are part of the solution. The GAC is the target location in this case.

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

<Solution SolutionId="78c8fd78-0b7f-4893-9361-ff005c32b12c" xmlns="http://schemas.microsoft.com/sharepoint/">

<FeatureManifests>

<FeatureManifest Location="DocumentArchiveWorkflowPart1\Feature.xml" />

</FeatureManifests>

<Assemblies>

<Assembly Location="DocumentArchiveWorkflowPart1.dll" DeploymentTarget="GlobalAssemblyCache"/>

</Assemblies>

</Solution>

### Create the **Package.ddf** in the **Solution** folder. Its purpose is to define the generated package.

#### Right click the **Solution** folder and click **Add -> New Item…**.

#### Select the **General** category on the left and select a template of **Text file**.

#### Name the new file **Package.ddf** and click **Add**.

### Paste the following text into your DDF file to define the files to be packaged.

.OPTION EXPLICIT ; Generate errors

.Set CabinetNameTemplate=DocumentArchiveWorkflowPart1.wsp

.set DiskDirectoryTemplate=CDROM ; All cabinets go in a single directory

.Set CompressionType=MSZIP;\*\* All files are compressed in cabinet files

.Set UniqueFiles="ON"

.Set Cabinet=on

.Set DiskDirectory1=Package

..\..\Solution\Manifest.xml manifest.xml

..\..\Template\Features\DocumentArchiveWorkflowPart1\Feature.xml DocumentArchiveWorkflowPart1\Feature.xml

..\..\Template\Features\DocumentArchiveWorkflowPart1\Workflow.xml DocumentArchiveWorkflowPart1\Workflow.xml

DocumentArchiveWorkflowPart1.dll DocumentArchiveWorkflowPart1.dll

### Add another step to the post build process that creates the solution package.

#### Right click the project in the **Solution Explorer** and click **Properties**.

#### Click the **Build Events** tab in the project properties window.

#### Enter the following commands into the **Post-build event command line** window at the top of the list.

makecab /f "$(ProjectDir)\Solution\Package.ddf"

### Rebuild the project.

#### Right click on the project in the **Solution Explorer** and click **Rebuild**.

#### Verify the rebuild was successful.

### Manually uninstall the previously deployed **Document Archive** feature.

#### Open a command window in the SharePoint **bin** directory.

##### Click **Start -> Run** and enter **cmd**.

##### Navigate to C**:\Program Files\Common Files\Microsoft Shared\web server extensions\12\BIN**.

#### Execute **stsadm.exe** to uninstall the **DocuentArchiveWorkflowPart1** feature.

stsadm –o uninstallfeature –name DocumentArchiveWorkflowPart1 –force

#### Delete the folder at **C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\Template\Features\DocumentarchiveWorkflowPart1**.

### Install and deploy the new solution package.

#### In the same command window as the previous step, execute **stsadm** to add the feature to SharePoint.

stsadm -o addsolution –filename “C:\Labs\Lab06\StarterFiles\DocumentArchiveWorkflowPart1\bin\Debug\Package\DocumentArchiveWorkflowPart1.wsp”

#### Deploy the solution to the farm using the **deploysolution** command in **stsadm**.

stsadm –o deploysolution –name DocumentArchiveWorkflowPart1.wsp -allowgacdeployment -local

### Verify the feature is still available and activated.

#### Using **Internet Explorer** navigate to the **Demo** site collection at **http://litwareinc.com/sites/Demo**.

#### Open the features list by clicking **Site Actions -> Site Settings**.

#### On the **Site Settings** page, click **Site collection features** in the **Site** **Collection Administration** section.

#### Verify the **Document Archive – Part 1** feature is active.

## Challenge: Using the workflow activation properties.

The **OnWorkflowActivated** activity provides a **WorkflowProperties** property. This property contains information about the **SPListItem** that the workflow is attached to. Use the **DisplayName** property in this object to populate the **HistoryOutcome** of the **LogToHistoryList** activity.