# Lab 10: Integrating InfoPath Forms into a SharePoint Workflow Template

**Lab Overview:** Designing workflow forms in ASP.NET is not a task easily delegated to web designers. There is no simple way to provide web designers with a visual design environment; therefore, workflow form design is currently the responsibility of developers. This is not a good scenario for forms that may need minor layout and branding adjustments that don’t change the way they work.

The solution to this problem is to use InfoPath. InfoPath provides a functional user interface that interacts with XML data. This is perfect for interacting with SharePoint Workflow Association, Initiation, Task, and Modification data. In this lab, you’ll be building a simple approval workflow using InfoPath as the user interface.

## 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\Lab10\StarterFiles\InfoPathApprovalWorkflow.sln**.

## Exercise 1: InfoPath Association Forms

### Create a new InfoPath form named **AssocInitForm.xsn** in the project’s **Forms** folder.

#### Open InfoPath by clicking the **Start Button** and selecting **All Programs -> Microsoft Office -> Microsoft Office InfoPath 2007**.

#### In the **Getting Started** window, click the **Design a Form Template** link in the **Design a form** section.

#### In the **Design a Form Template** window, select a **Blank** form template and make sure the **Enable browser-compatible features only** check box is checked and click **OK**.

##### 

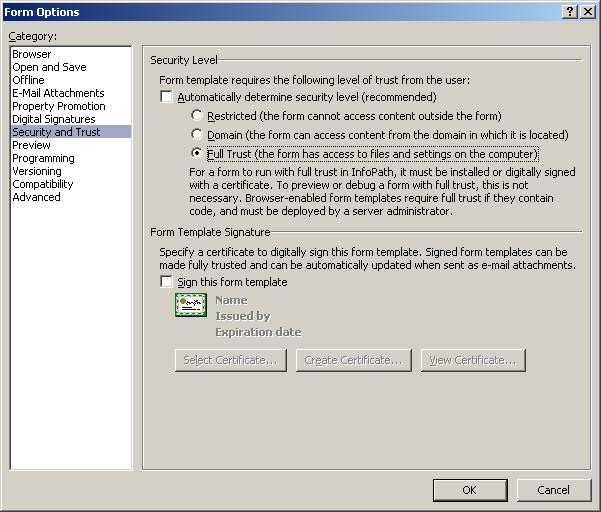
### Update the form’s security options to grant the form full trust when hosted in Forms Services.

#### Click the **Tools -> Form Options** item in the menu bar.

#### In the **Form Options** dialog, select **Security and Trust** in the **Category** list box.

#### Uncheck **Automatically determine security level** and select **Full Trust**.

#### Click **OK** to close the dialog.



### Add a secondary data source that will allow the hosting environment to provide extra data to the form. This is required to use the **Contact Selector** in the next step.

#### In **Visual Studio 2008**, add a new file named **Context.xml** to the **Forms** folder.

##### Right click on the **Forms** folder in the **Solution Explorer** and select **Add -> New Item**.

##### Select the **Data** category in the left side of the screen and then select **Xml** **File** in the **Template** list.

##### Enter a name of **Context.xml** and click **Add**.

#### Enter the following XML into the new **Context.xml** file.

<Context

isStartWorkflow="true"

isRunAtServer="false"

provideAllFields="true"

siteUrl="" />

#### In InfoPath, click **Tools -> Data Connections** in the menu bar.

#### In the **Data Connections** dialog, click the **Add** button.

#### Select the **Create a new connection to** option and then select **Receive data** and click **Next**.

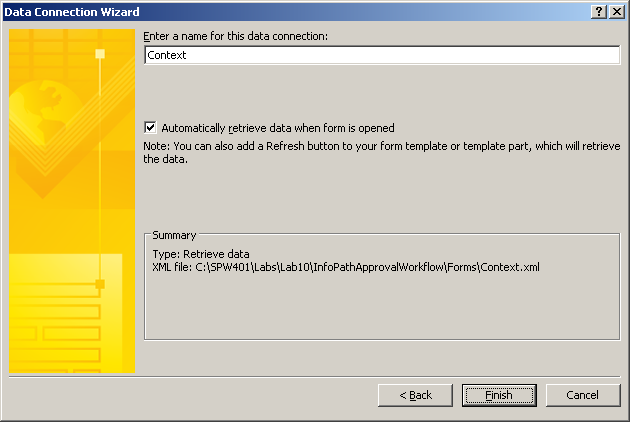
#### On the next page, choose to receive data from an **XML document**.

#### Browse to the **Context.xml** file just created and click **Next**.

#### On the next page, select to include the file as a resource and click **Next**.

#### On the final page, click **Finish** to create the data source.

#### In the **Data Connections** dialog, click **Close** to close the window.



### Add a **Contact Selector** ActiveX control to the controls list for adding to the form later.

#### Locate the **Controls** task pane on the right hand side of the page.

##### If it is not visible, click the drop down menu at the top of the task pane and select **Controls**.

#### Click **Add or Remove Custom Controls** to add the **Contact Selector** control.

#### In the **Add or Remove Custom Controls** dialog, click the **Add** button.

#### In the first page of the wizard, choose **ActiveX Control** and click **Next**.

#### Select **Contact Selector** in the control list and click **Next**.

#### Choose to not include a cab file and click **Next**.

#### Select a binding property of **Value** and click **Next**.

#### In the **Field or group type** drop down list, choose **Field or Group (any data type)** and click **Finish**.

#### On the summary page, click **Close**.

#### Now the **Contact Selector** should be visible in the **Add or Remove Custom Controls** dialog. Close the dialog when you are done.

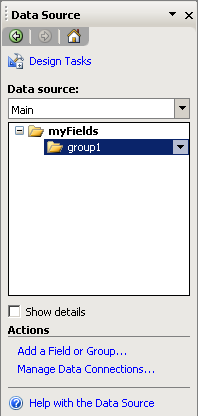
##### 

### Add a **Contact Selector** to the form and bind it to the data source.

#### Drag a **Contact Selector** from the **Controls** task pane onto the design surface.

#### Switch to the **Data Source** view using the drop down menu at the top of the task pane.

##### Notice the **Contact Selector** control has already created a new group named **group1**.



#### Rename the **myFields** and **group1** groups to **AssocInitData** and **Approvers**.

##### Right click **myFields** and select **Properties**.

##### In the **Field or Group Properties** dialog, change the name to **AssocInitData**.

##### Click **OK** to close the dialog.

##### Repeat to change **group1** to **Approvers**.



#### Add a **Person** repeating group as a child of the **Approvers** group.

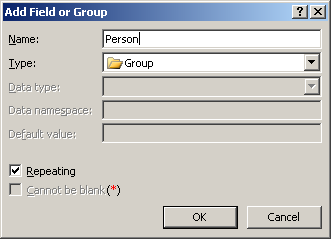
##### Right click the **Approvers** group and select **Add**.

##### In the **Add Field or Group** dialog, enter a name of **Person**.

##### Change the **Type** to **Group**.

##### Check the **Repeating** checkbox.

##### Click **OK** to create the new group.



#### Add three more string fields named **DisplayName**, **AccountId**, **AccountType** as children of the **Person** group.

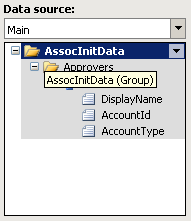
##### Right click the **Person** group and click **Add**.

##### In the **Add Field or Group** dialog, enter a name of **DisplayName**.

##### Verify the **Data** **type** is **string**.

##### Click **OK** to create the new field.

##### Repeat the above steps for **AccountId** and **AccountType**.



### Add a **Text Box** control to the form to be used to enter any instructions to the approver.

#### Add an **Instructions** field to the **AssocInitData** group.

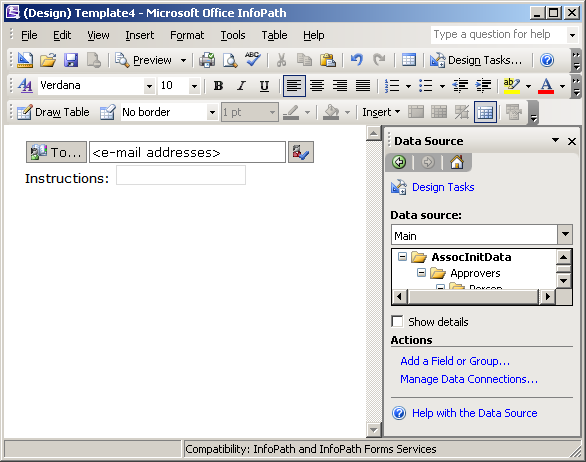
##### In the **Data Source** task pane, right click the **AssocInitData** group and select **Add**.

##### In the **Add Field or Group** dialog, enter a name of **Instructions**

##### Verify the **Data type** is **string**.

##### Click **OK** to create the new field.

#### Drag the **Instructions** field onto the design canvas to create a new text box.



### Add two buttons to the page for the **OK** and **Cancel** operations.

#### Switch to the **Controls** task pane and drag two button controls onto the design canvas.

#### Change the name of one button to **OK**.

##### Right click the button and select **Button Properties**.

##### In the **Button Properties** dialog, change the **Label** value to **OK**.

##### Click **OK** to close the dialog.

#### Change the name of the other button to **Cancel**.

##### Right click the button and select **Button Properties**.

##### In the **Button Properties** dialog, change the **Label** value to **Cancel**.

##### Click **OK** to close the dialog.



### Define rules for the **Cancel** button that will cause it to close the form.

#### Open the **Rules** dialog to manage the rules that will execute when the button is clicked.

##### Right click the cancel button and select **Button Properties**.

##### Click the **Rules** button to open the **Rules** dialog.

#### Add a new rule named **Close Form** that will be executed when the button is clicked.

##### Click the **Add** button to add a new rule.

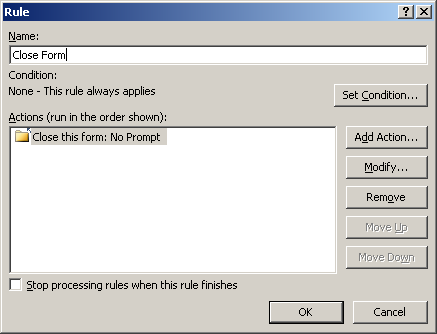
##### Change the **Name** property in the **Rule** dialog to **Close Form**.

#### Add a close form action that will close the current form.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Close** **the form**

##### Click **OK** to close the dialog.



#### Close all of the open dialogs.

### Define rules for the **OK** button that will cause it to submit and then close the form.

#### Open the **Rules** dialog to manage the rules that will execute when the button is clicked.

##### Right click the cancel button and select **Button Properties**.

##### Click the **Rules** button to open the Ru**l**es dialog.

#### Add a new rule named **Submit and** **Close Form** that will be executed when the button is clicked.

##### Click the **Add** button to add a new rule.

##### Change the **Name** property in the **Rule** dialog to **Submit and** **Close Form**.

#### Add a submit form action that will submit the data to Forms Services.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Submit** **using a data connection**.

##### Click the **Add** button to add a new submission data connection.

##### In the **Data Connection Wizard** create a new **Submit** data connectionand click **Next**.

##### Select **To the hosting environment** and click **Next**.

##### Keep the default name of **Submit** and click **Finish**.

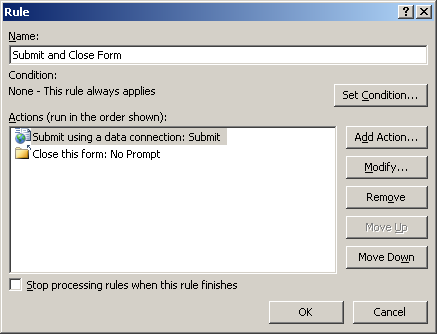
##### Back in the **Action** dialog, verify the data connection **Submit** is selected and click **OK** to close the dialog.

#### Add a close form action that will close the current form.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Close the form**

##### Click **OK** to close the dialog.



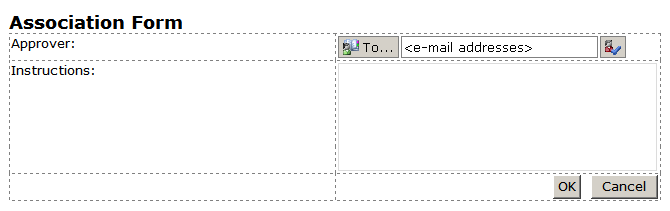
#### Close all of the open dialogs.

### Update the layout of the form to look nicer. Add a title of Associat**ion Form** that is size 14pt and bold and place the controls in a layout table.

#### Using InfoPath like a word processor, add the text **Association Form** to the top of the page. Use a font size of 14pt and bold.

#### Create a layout grid of 2x3 and place the approver and instructions controls in the top two rows. Adjust the height of the Instructions text box so it can contain multiple lines.

#### Place the buttons in the lower right cell of the layout table and right justify them.



### Save the new form template into the **Forms** folder of the Visual Studio Project.

#### Click the **Save** toolbar button.

#### Save the form in **C:\Labs\Lab10\StarterFiles\InfoPathApprovalWorkflow\Forms** with a name of **AssocInitForm.xsn** and close InfoPath.

### Extract the schema file form the new InfoPath form.

#### Locate the **AssocInitForm.xsn** file in the **Windows Explorer** and rename it **AssocInitForm.xsn.cab**.

#### Open the new cab file and copy the **myschema.xsd** file contained within it into the root folder of the **InfoPathApprovalWorkflow** project.

#### Change the name of the **AssocInitForm.xsn.cab** back to **AssocInitForm.xsn**.

### Update the **AssocInitData** class by using the **xsd.exe** tool to regenerate the **AssocInitData** class.

#### In the **InfoPathApprovalWorkflow** folder, rename the **myschema.xsd** to **AssocInitData.xsd**.

#### Open a **Visual Studio 2008 Command Prompt** in the **InfoPathApprovalWorkflow** folder.

##### Click the **Start Button** and click **All Programs -> Microsoft Visual Studio 2008 -> Visual Studio Tools -> Visual Studio 2008 Command Prompt**.

##### In the command prompt, navigate to **C:\Labs\Lab10\StarterFiles\InfoPathApprovalWorkflow**.

#### Execute **xsd.exe** to regenerate the class that will serialize xml to the format defined in the InfoPath schema.

xsd.exe AssocInitData.xsd /c /n:InfoPathApprovalWorkflow

#### Open the generated **AssocInitData.cs** in **Visual Studio 2008**.

##### Comment out the **anyAttrField** field and property if it exists

## Exercise 2: InfoPath Instantiation Form using Views

### Add a new view named **Instantiation View** to the **InfoPath** form

#### Design **AssocInitForm.xsn** in InfoPath.

#### Select the **Views** task pane using the drop down list at the top of the task pages window.

#### Rename the default view to **Association View**.

##### Select the default view and click the **View Properties** button.

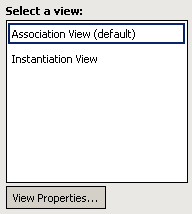
##### Change the **View name** to **Association View** and click **OK**.

#### Add a new view named **Instantiation View** to the form.

##### Click the **Add a New View** link on the **Views** task pane.

##### In the **Add View** dialog, enter the name of **Instantiation View** and click **OK**.

#### Switch back to the **Association View** by clicking it in the list of **Views**.



### Copy all of the controls from the **Association View** to the **Instantiation View** and update them to display an instantiation view

#### Copy and paste the contents of the **Association View** into the **Instantiation** **View**.

##### Select the entire content of the **Association View** in the designer and click **Edit -> Copy**.

##### Switch to the **Instantiation View** using the **Views** task pane.

##### Click **Edit -> Paste** to paste the contents into the new view.

#### Change the header from **Association Form** to **Instantiation Form**.

#### Change the text of the **OK** button to **Start**.

##### Right click the **OK** button and click **Button Properties**.

##### In the **Button Properties** dialog, change the name from **OK** to **Start**.

##### Click **OK** to close the dialog.

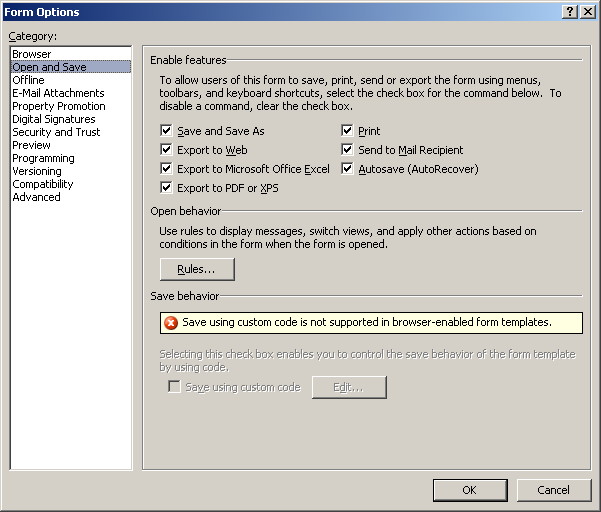


### Add form load rules that will choose the appropriate view based on the **isStartWorkflow** field in the Context data source.

#### Using the **Form Options** dialog, open the **Open** behavior rules for the form.

##### Click **Tools -> Form Options** in the menu bar.

##### Switch to the **Open and Save** category and click the **Rules** button.



#### Add rule to switch to association form if **isStartWorkflow == “false”**.

##### Click the **Add** button to add a new rule.

##### Set the **Name** of the new rule to **Is Not Starting Workflow**.

##### Click the **Set Condition** button to set the conditions under which the actions will execute.

##### On the left hand side drop down list, click **Select a Field or Group**.

##### In the **Select a Field or Group** dialog, choose the **Context** data source and the **isStartWorkflow** field and click **OK**.

##### In the center drop down list, verify **is equal to** is selected.

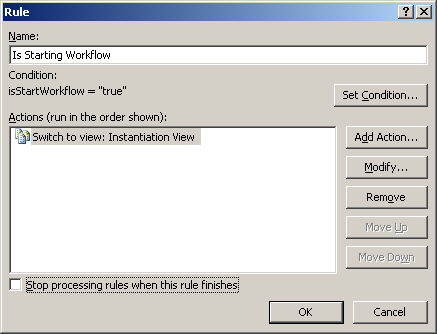
##### In the right drop down list, click **type text**, enter the text **false**, and click **OK**.

##### Click **OK** again to close the condition.

##### Click **Add Action** to add a new switch view action.

##### Select an action of **Switch Views** and a view of **Association View**.

##### Click **OK** to close the dialogs back to the **Rules** dialog.



#### Add rule to switch to instantiation form if **isStartWorkflow** **== “true”**.

##### Click the **Add** button to add a new rule.

##### Set the **Name** of the new rule to **Is Starting Workflow**.

##### Click the **Set Condition** button to set the conditions under which the actions will execute.

##### On the left hand side drop down list, click **Select a Field or Group**.

##### In the **Select a Field or Group** dialog, choose the **Context** data source and the **isStartWorkflow** field and click **OK**.

##### In the center drop down list, verify **is equal to** is selected.

##### In the right drop down list, click **type text** and enter the text **true** and click **OK**.

##### Click **OK** again to close the condition.

##### Click **Add Action** to add a new switch view action.

##### Select an action of **Switch Views** and a view of **Instantiation View**.

##### Click **OK** to close the dialogs back to the designer.

## Exercise 3: InfoPath Task Forms

### Create a new InfoPath form named **TaskForm.xsn**.

#### Open **InfoPath** by clicking the **Start Button** and selecting **All Programs -> Microsoft Office -> Microsoft Office InfoPath 2007**.

#### In the **Getting Started** window, click the **Design a Form Template** link in the **Design a form** section.

#### In the **Design a Form Template** window, select a **Blank form** template and make sure the **Enable browser-compatible features only** check box is checked and click **OK**.

##### 

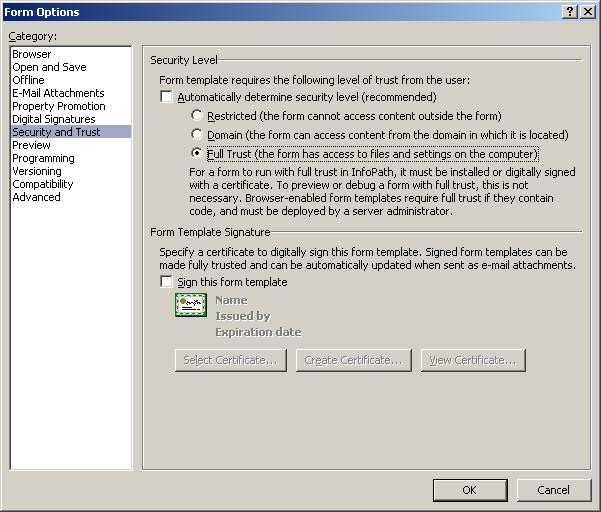
### Update the form’s security options so the form will be granted full trust when hosted in Forms Services.

#### Click the **Tools -> Form Options** item in the menu bar.

#### In the **Form Options** dialog, select **Security and Trust** in the **Category** list box.

#### Uncheck **Automatically determine security level** and select **Full Trust**.

#### Click **OK** to close the dialog.



### Add a secondary data source that will allow the form to access fields in the current task.

#### In **Visual Studio 2008**, add a new file named **ItemMetadata.xml** to the **Forms** folder.

##### Right click on the **Forms** folder in the **Solution Explorer** and select **Add -> New Item**.

##### Select the **Data** category in the left side of the screen and then select **Xml** **File** in the **Template** list.

##### Enter a name of **ItemMetadata.xml** and click **Add**.

#### Enter the following XML into the new **ItemMetadata.xml** file.

<z:row xmlns:z="#RowsetSchema"

ows\_Instructions="" />

#### Add a new data connection using the **ItemMetadata.xml** file.

##### In InfoPath, click **Tools -> Data Connections** in the menu bar.

##### In the **Data Connections** dialog, click the **Add** button.

##### Select the **Create a new connection to** option and then select **Receive data** and click **Next**.

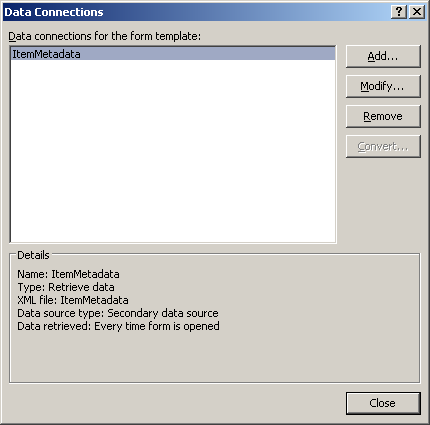
##### On the next page, choose to receive data from an **XML document**.

##### Browse to the **ItemMetadata.xml** file just created and click **Next**.

##### On the next page select to include the file as a resource and click **Next**.

##### On the final page, click **Finish** to create the data source.

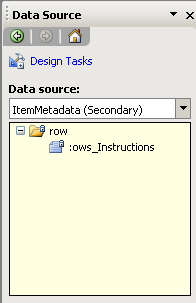
##### In the **Data Connections** dialog, click **Close** to close the window.



#### Using the new data source, add a new control using the **ows\_Instructions** field.

##### In the **Data Source** task pane, use the **Data source** drop down to change to the **ItemMetadata** data source.

##### Drag the **ows\_Instructions** field onto the designer to create an **Instructions** control.



### Define the output data that will be transmitted back to the workflow when the task is submitted.

#### Add a new **Comments** field of type string to the primary data source.

##### In the **Data Source** task pane, right click the **myFields** group and select **Add**.

##### In the **Add Field or Group** dialog, enter a name of **Comments**.

##### Verify the **Data type** is **string**.

##### Click **OK** to create the new field.

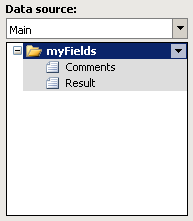
#### Add a new **Result** field of type string to the primary data source.

##### In the **Data Source** task pane, right click the **myFields** group and select **Add**.

##### In the **Add Field or Group** dialog, enter a name of **Result**.

##### Verify the **Data type** is **string**.

##### Click **OK** to create the new field.



#### Drag the **Comments** field onto the design canvas to create a new text box.

### Add three buttons to the page for the **Approve**, **Reject**, and Cancel **operations**.

#### Switch to the **Controls** task pane and drag two button controls onto the design canvas.

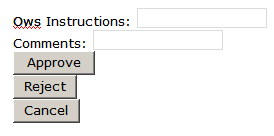
#### Change the name of one button to **Approve**.

##### Right click the button and select **Button Properties**.

##### In the **Button Properties** dialog, change the **Label** value to **Approve**.

##### Click **OK** to close the dialog.

#### Repeat the previous step for the **Reject** and **Cancel** buttons.



### Define rules for the **Cancel** button that will cause it to close the form.

#### Open the **Rules** dialog to manage the rules that will execute when the button is clicked.

##### Right click the **cancel** button and select **Button Properties**.

##### Click the **Rules** button to open the **Rules** dialog.

#### Add a new rule named **Close Form** that will be executed when the button is clicked.

##### Click the **Add** button to add a new rule.

##### Change the **Name** property in the **Rule** dialog to **Close Form**.

#### Add a close form action that will close the current form.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Close** **the form**

##### Click **OK** to close the dialog.

##### Close all of the open dialogs.

### Define rules for the **Approve** button that will cause it to submit and then close the form.

#### Open the **Rules** dialog to manage the rules that will execute when the button is clicked.

##### Right click the cancel button and select **Button Properties**.

##### Click the **Rules** button to open the Ru**l**es dialog.

#### Add a new rule named **Approve and** **Close Form** that will be executed when the button is clicked.

##### Click the **Add** button to add a new rule.

##### Change the **Name** property in the **Rule** dialog to **Approve and Close Form**.

#### Add a **Set a field’s value** action that will set the **Result** field to **Approved**.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Set a field’s value**.

##### Click the button following the **Field** text box and choose the **Result** field.

##### In the **Value** text box, enter **Approved** and click **OK** to create the action.

##### Back in the **Action** dialog, verify the data connection **Submit** is selected and click **OK** to close the dialog.

#### Add a submit form action that will submit the data to Forms Services.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Submit** **using a data connection**.

##### Click the **Add** button to add a new submission data connection.

##### In the **Data Connection Wizard** create a new **Submit** data connectionand click **Next**.

##### Select **To the hosting environment** and click **Next**.

##### Keep the default name of **Submit** and click **Finish**.

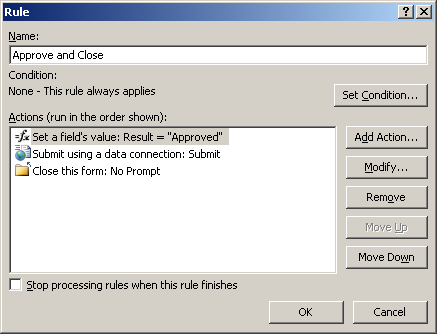
##### Back in the **Action** dialog, verify the data connection **Submit** is selected and click **OK** to close the dialog.

#### Add a close form action that will close the current form.

##### Click **Add Action** to open the **Action** dialog.

##### In the **Action** dialog, select an action of **Close the form**

##### Click **OK** to close the dialog.

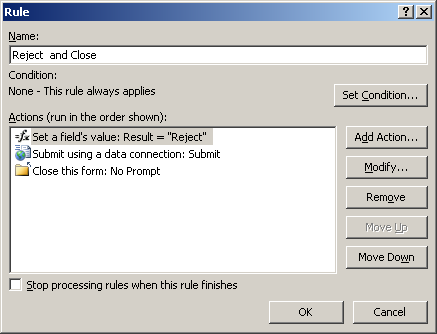


### Repeat the previous step to define the rules for the **Rejected** button.

#### Perform the previous steps with the following changes:

##### Use the rule name of **Reject** **and** **Close Form**.

##### Set the **Result** field value to **Rejected**.



### Update the layout of the form to look nicer. Add a title of **Task Form** that is size 14pt, bold, and place the controls in a layout table.

#### Using InfoPath like a word processor, add the text Asso**ciation Form** to the top of the page. Use a font size of 14pt and bold.

#### Create a layout grid of 2x3 and place the instructions and comments controls in the top two rows. Adjust the height of the text boxes so they can contain multiple lines.

#### Place the buttons in the lower right cell of the layout table and right justify them.



### Save the new form template into the **Forms** folder of the Visual Studio Project.

#### Click the **Save** toolbar button.

#### Save the form in **C:\Labs\Lab10\StarterFiles\InfoPathApprovalWorkflow\Forms** with a name of **TaskForm.xsn**.

### Update the workflow to support providing values to the task form and receiving results back.

#### In **CreateApprovalTask\_Invoking**, add an extended property named Instructions getting its value from the **AssocInitData** property.

##### Right click **Workflow.cs** in the **Solution Explorer** and click **View Code**.

##### Locate the **CreateApprovalTask\_Invoking** and add an Instructions property to the task properties.

##### Get the value of the extended property from the **AssocInitData** property of the workflow.

createApprovalTask.TaskProperties.ExtendedProperties.Add(

"Instructions", InitData.Instructions);

#### In **ApprovalTaskChanged\_Invoked** retrieve the extended property with the Guid id of **{52578fc3-1f01-4f4d-b016-94ccbcf428cf}** representing the **Comments** column.

##### Locate the **ApprovalTaskChanged\_Invoked** method.

##### Add the line to retrieve the comments extended property.

this.ApprovalTaskComments = args.afterProperties.ExtendedProperties[

new Guid("52578fc3-1f01-4f4d-b016-94ccbcf428cf")] as string;

#### In **ApprovalTaskChanged\_Invoked** retrieve the **Result** expended property and convert it into a **TaskResult** enum value.

string result = args.afterProperties.ExtendedProperties["Result"] as string;

this.ApprovalTaskResult = (TaskResult)Enum.Parse(typeof(TaskResult), result);

## Exercise 4: Deploying InfoPath Workflow Forms

### Publish the **AssocInitForm.xsn** form to the **Template\Features\InfoPathApprovalWorkflow\Forms** folder in the project.

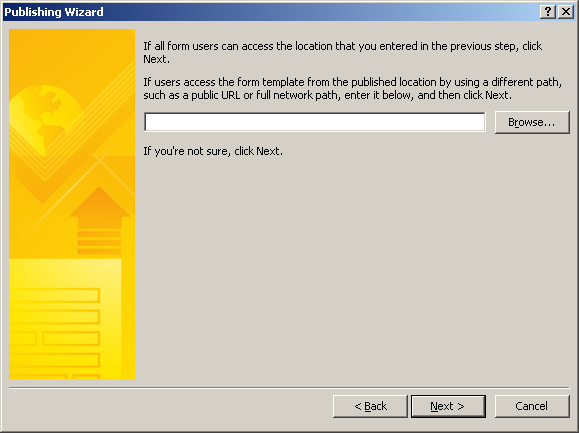
#### Design the **Forms\AssocInitForm.xsn** form template in InfoPath.

#### Publish the form by clicking **File -> Publish** in the menu bar.

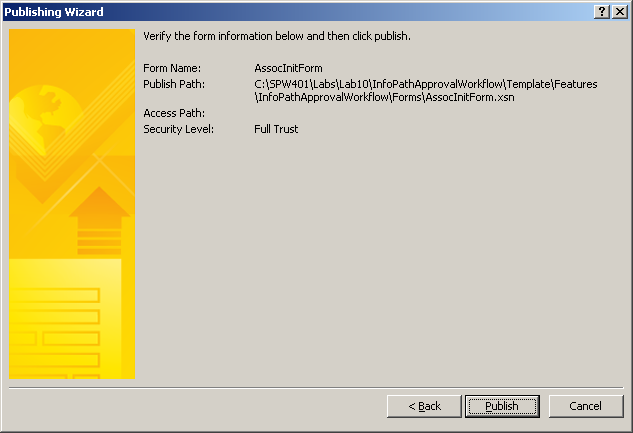
#### In the **Publishing Wizard**, choose to publish to a network location and click **Next**.

#### Publish the document to the **C:\Labs\Lab10\StarterFiles\InfoPathApprovalWorkflow\Template\Features\InfoPathApprovalWorkflow\Forms\AssocInitForm.xsn** location.

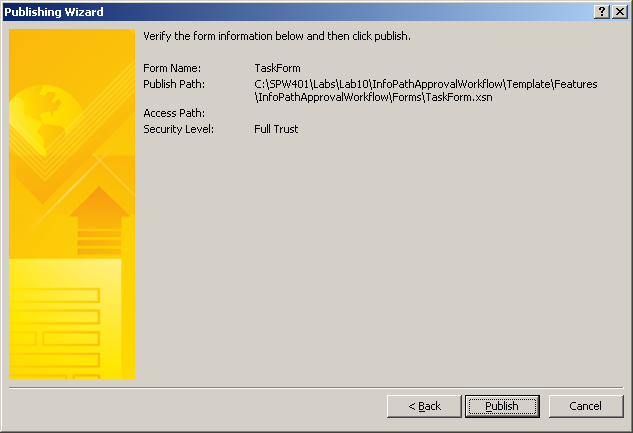
#### When asked for the public URL make sure the text box is empty and click **Next**.



#### On the final page of the wizard, verify the results and click **Publish** and then **Close**.



### Using the process in the previous step, publish the **Task.xsn** form to the **Template\Features\InfoPathApprovalWorkflow\Forms** folder in the project.



### Update the **feature.xml** file to indicate the location of the InfoPath forms and include a receiver assembly that will install those forms to Forms Services.

#### Open the **Feature.xml** file in the **Template\Features\InfoPathApprovalWorkflow** folder.

#### Add a **ReceiverAssembly** and **ReceiverClass** to the **Feature** element.

ReceiverAssembly="Microsoft.Office.Workflow.Feature, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"

ReceiverClass="Microsoft.Office.Workflow.Feature.WorkflowFeatureReceiver"

#### Add a **Property** element to the **Properties** element that will set a property of **RegisterForms** to the **Forms\\*.xsn** value.

<Properties>

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

<Property Key="RegisterForms" Value="Forms\\*.xsn" />

</Properties>

### Update the **workflow.xml** file to define new form Urls and content types for tasks.

#### Open the **Workflow.xml** file in the **Template\Features\InfoPathApprovalWorkflow** folder.

#### Add an **AssociationUrl** and **InstantiationUrl** to the **Workflow** element.

##### These values are special Forms Services hosting pages.

AssociationUrl="\_layouts/CstWrkflIP.aspx"

InstantiationUrl="\_layouts/IniWrkflIP.aspx">

#### Add a TaskListContentTypeId attribute to the Workflow element.

##### This is a special **Forms Services** content type with the appropriate view and edit forms already defined.

TaskListContentTypeId="0x01080100C9C9515DE4E24001905074F980F93160"

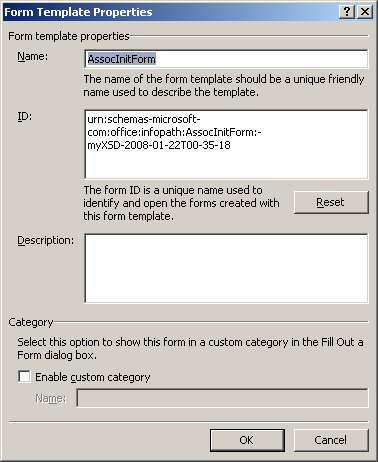
### Add form registration URNs that are used by the special Forms Services pages to identify the InfoPath form to load.

#### Lookup the URN of the **AssocInitForm.xsn** using the form’s **Properties** window.

##### Right click the form in **Windows Explorer** and select **Design**.

##### When the form is loaded, click **File -> Properties**.

##### Copy the **ID** property from the dialog.



#### Use the URN for the form to populate the Association and Instantiation Form URNs in the workflow metadata.

##### Open **Workflow.xml** and add the following metadata into the form.

##### Replace the values in **Association\_FormURN** and **Instantiation\_FormURN** with the URN you copied from InfoPath.

<MetaData>

<Association\_FormURN>urn:schemas-microsoft-com:office:infopath:AssocInitForm:-myXSD-2008-01-22T00-35-18</Association\_FormURN>

<Instantiation\_FormURN>urn:schemas-microsoft-com:office:infopath:AssocInitForm:-myXSD-2008-01-22T00-35-18</Instantiation\_FormURN>

<Task0\_FormURN>urn:schemas-microsoft-com:office:infopath:TaskForm:-myXSD-2008-01-22T03-21-33</Task0\_FormURN>

</MetaData>

#### Repeat the previous steps populate Task0\_FormURN with the URN of **TaskForm.xsn**.

### Rebuild the project to force redeployment of the forms.

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

### Activate the new **InfoPathApprovalWorkflow** 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 **InfoPath Approval Workflow** feature.



### Create an association between the **Shared Documents** document library and the new **InfoPath Approval Workflow** workflow.

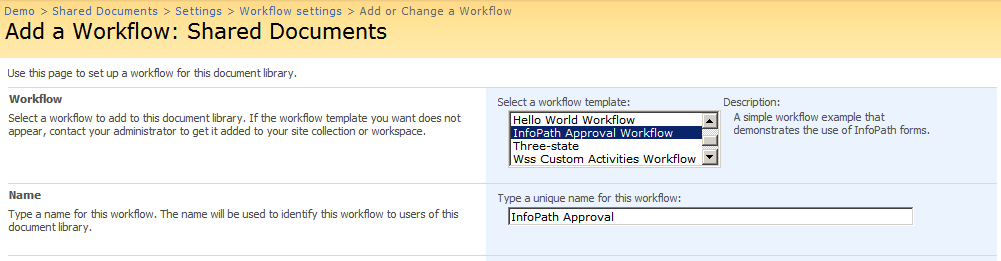
#### Navigate to the **Shared Documents** document library in the **Demos** site.

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

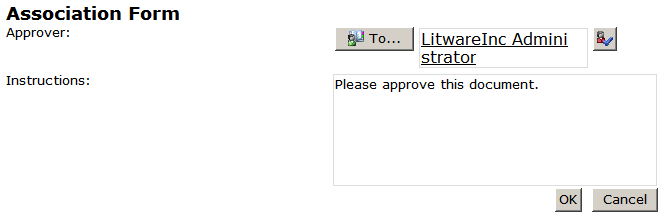
#### Click the **Workflow settings** link in the **Permissions and Management section**.

#### Create a new workflow using the **InfoPath Approval Workflow** workflow template and a name of **InfoPath Approval**.

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



#### In the custom association form, enter Administrator as the approver and add “Please approve this document.” as the instructions.



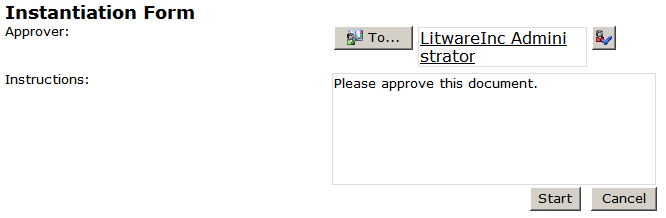
### Run the **InfoPath Approval** workflow on a document.

#### Navigate to the Shared Documents document library and hover over the new document and select **Workflows** from the drop down menu.

#### In the workflows page, click the **InfoPath Approval** to start the workflow.

#### In the **Shared Documents** document library, verify the workflow is running.

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

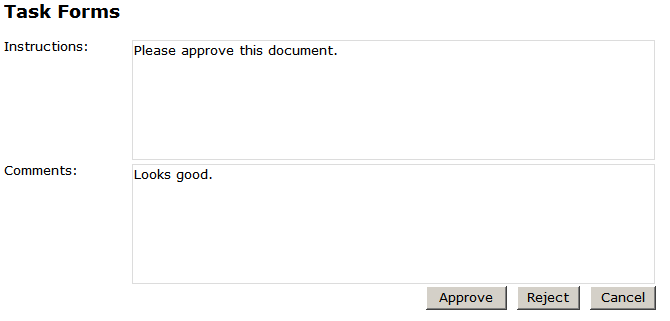


### Approve the document using the Custom Task Form.

#### Click the task’s title in the workflow status page.

#### Notice that the **Instructions** text box defaults to the value set in the **Instantiation** **Form**.

#### Click the **Approve** button to approve the task.



### Verify that the workflow is now completed and the task is marked as completed.

