Lab 13: Creating Custom Workflows for MOSS

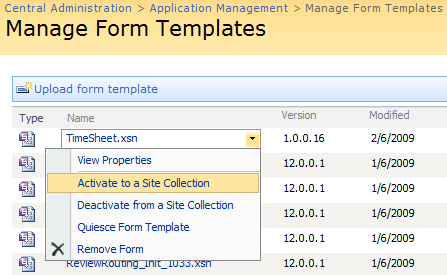
**Lab Time:** 60 minutes

**Lab Overview**: Litware is a consultancy company requiring each of their consultants to fill-in a daily timesheet summarizing all of their activities done during that day. The timesheet data is stored in a custom SharePoint list for later use. At the end of every week managers must review the timesheets to approve or disapprove the hours submitted.

* In Exercise 1 you will create a workflow to create a task assigned to the consultant's manager and provide them with a form that allows easy acceptance or rejection of the timesheet.
* In Exercise 2, you will use Visual Studio to create a custom workflow.

**Lab Setup—5 pieces**:

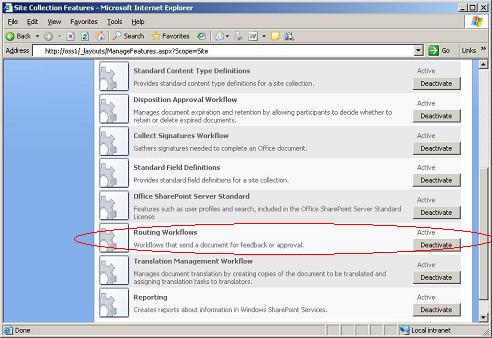
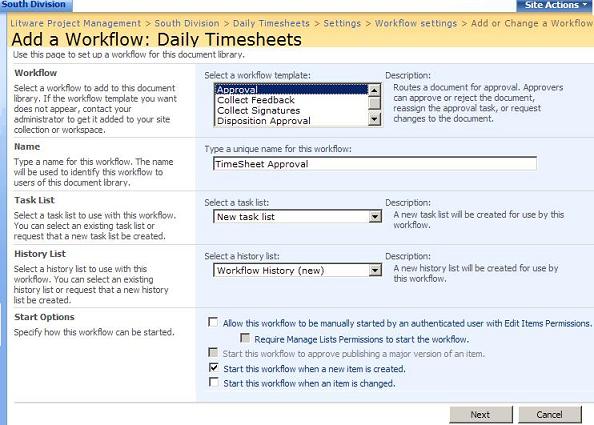
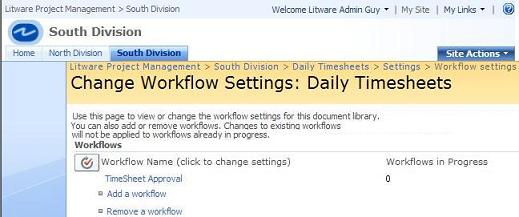
* **Lab 09** on Custom Content types **MUST** be completed before you can start this lab.
* Perform this step **ONLY** if you didn’t make **Lab 12** on **InfoPath**:
  + Open **Windows Explorer**.
  + Navigate to the starter files of this lab: **C:\Student\Labs\13\_Workflow\Starter Files\Setup**.
  + Double-click the **Mod11Starter.bat**. This file will alter Active Directory to set the Department to Development and also set the manager field for certain users. It will also extend the profile database in SharePoint to include a setting for **HourlyRate** and it will pre-populate it for those in the Development Department.
    - Navigate to **SharePoint 3.0 Central Administration** (Start - All Programs - Microsoft Office Server - SharePoint 3.0 Central Administration)
    - In Central Administration on your **Quick Launch** bar (left side of screen) click on **Litware SSP**
    - In the **Litware SSP User Profiles and My Sites** section click on **User profiles and properties**
    - Click on **Start full import** (this will pull the new settings from Active Directory into SharePoint so that we might utilize them in this lab).
  + Double-click the **deploytimesheetsolution.bat** file to add and deploy the InfoPath timesheet solution.
  + To make the form template available to users, the form must be activated to a site collection.
    - Navigate to **Central Administration** > **Application Management** > **Manage Form Templates** page.
    - Point to the **TimeSheet.xsn** form template that you want to activate, click the arrow that appears, and then click **Activate to a Site Collection**.

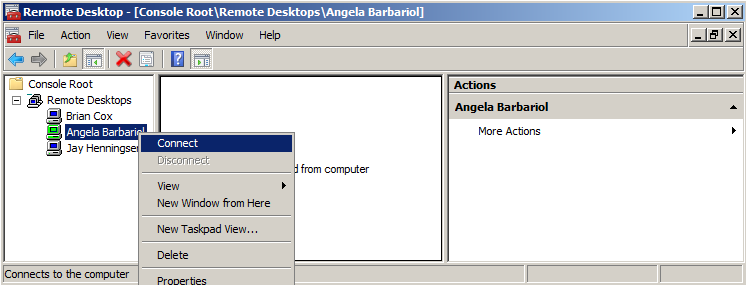


* + - On the **Activate Form Template** page, click the **Site Collection URL**, then click **Change Site Collection**.
    - Using the **Select Site Collection** page, click the **/sites/ProjectManagementLab** site, and then click **OK**.
* Perform the following steps **ONLY** if you attempted but did not complete **Lab 12** on **InfoPath**:
  + Open **Windows Explorer**.
  + Navigate to the starter files of this lab: **C:\Student\Labs\13\_Workflow\Starter Files\Setup**.
  + Double-click the **retracttimesheetsolution.bat** file to retract and delete the InfoPath timesheet solution.
  + Double-click the **deploytimesheetsolution.bat** file to add and deploy the InfoPath timesheet solution.
  + Finally we can now utilize the form template as a template for a library
* Perform this step **ONLY** if you didn’t make **Lab 12** on **InfoPath** orif you attempted but did not complete **Lab 12** on **InfoPath**:
  + Open Internet Explorer and navigate to **http://litwareinc.com/sites/ProjectManagementLab/SouthDivision**.
  + If you have a **Daily Timesheets** forms library:
    - Open the **Daily Timesheets** forms library.
    - Select and delete each existing test form in this library to make way for the new template.
  + If you have no **Daily Timesheets** forms library:
    - Create a new library based on the **Forms Library** template and give it the name **Daily Timesheets**.
  + On the top navigation bar, click **Settings**, and then click **Form Library Settings**.
  + On the **Customize Daily Timesheets** page, click **Advanced settings**.
  + On the **Form Library Advanced Settings: Daily Timesheets** page, under **Allow management of content types**, select **Yes**, and then click **OK**.
  + On the **Customize Daily Timesheets** page, in the **Content Types** section, click **Add from existing site content types**.
  + On the **Add Content Types: Daily Timesheets** page, in the **Available Site Content Types** list, select the **TimeSheet** content type to use as the template for this library, click **Add**, and then click **OK**.
  + On the **Customize Daily Timesheets** page, in the **Content Types** section, click on the **Form** content type.
  + On the **List Content Type: Form** page, select **Delete this content type** and select **OK** to the confirmation message.
* **EVERYONE** must set the necessary permissions on both the **Litwareinc** site and the **ProjectManagementLab** site**:**
  + Open Internet Explorer and navigate to **http://litwareinc.com**.
  + Select **People and Groups** from the **Quick Launch**.
  + Select the **Litware Inc Members** group.
  + Click the **New** button and choose **Add User**.
  + Add the following string in the **Add Users** section: **BrianC;AngelaB;JayH**
  + Click the **Check** button to validate the users.
  + Click the **OK** button.
  + Navigate to the **http://litwareinc.com/sites/ProjectManagementLab** site.
  + Select **People and Groups** from the **Quick Launch**.
  + Select the **Litware Project Management Members** group.
  + Click the **New** button and choose **Add Us**er.
  + Add the following string in the Add Users section: **BrianC;AngelaB;JayH**
  + Click the **Check** button to validate the users.
  + Click the **OK** button.
  + Check if you have a **Daily Timesheets** forms library in the **ProjectManagementLab** site. A custom workflow feature is a site collection scoped feature. You will need this library to be able to properly create a custom workflow project in **Visual Studio 2008**. If you have no **Daily Timesheets** forms library, create a new library based on the **Forms Library** template and give it the name **Daily Timesheets**.
  + On the top navigation bar, click **Settings**, and then click **Form Library Settings**.
  + On the **Customize Daily Timesheets** page, click **Advanced settings**.
  + On the **Form Library Advanced Settings: Daily Timesheets** page, under **Allow management of content types**, select **Yes**, and then click **OK**.
  + On the **Customize Daily Timesheets** page, in the **Content Types** section, click **Add from existing site content types**.
  + On the **Add Content Types: Daily Timesheets** page, in the **Available Site Content Types** list, select the **TimeSheet** content type to use as the template for this library, click **Add**, and then click **OK**.
  + On the **Customize Daily Timesheets** page, in the **Content Types** section, click on the **Form** content type.

# Exercise 1: Associating and Initiating SharePoint Workflows

SharePoint provides several workflow types out of the box. One of those workflow types is an approval workflow. Whenever a user needs to interact with the workflow, a task is added to a **Tasks** list. To better facilitate user interaction, the approval workflow provides InfoPath forms to simplify interaction with the tasks. In this exercise you'll attach an approval workflow to the **Daily Timesheets** document library to facilitate manager approval of the timesheets.

1. Before you can use the **Approval** workflow which is included with SharePoint, you'll need to enable the **Routing Workflows** feature on the site collection.
   1. Open the site at **http://Litwareinc.com/sites/ProjectManagementLab** and navigate to the site settings using the **Site Actions -> Site Settings** menu.
   2. In the **Site Collection Administration** page find and click the **Site Collection Features** link.
   3. In the features list, find the **Routing Workflows** feature and make sure that it is activated. 
2. Now that all the preliminary work is done, it's time to associate the **Approval** workflow with the **Daily Timesheet** Form library.
   1. Open the site at **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision** and navigate to the **Daily Timesheets** Form library.
   2. Open the Form library settings by using the **Settings -> Form Library Settings** menu item.
   3. On the Form library settings page, navigate to **Workflow Settings** in the **Permissions and Management** section.
   4. Select the **Approval** workflow template and name the workflow **TimeSheet Approval**.
   5. Choose **New task list** from the Task List dropdown.
   6. Leave the default setting **Workflow History (new)** so a new workflow history list will be created for you.
   7. Be sure to uncheck **Allow this workflow to be manually started by an authenticated user with Edit Items Permissions**.
   8. Check the box next to **Start this workflow when a new item is created**. 
   9. Click the **Next** button to move to the next step.
   10. This page allows you to configure many settings related to the approval process. You're only interested in the **Approvers** text box. Enter **LITWAREINC\Administrator** so the administrator must approve the document. Press **Ctrl + k** to Check the name.
   11. Finally click **OK** to complete the workflow creation. 
3. You've successfully created the **Approval** workflow and associated it with the form library. All that's left to do is test the approval process.
   1. Minimize all Windows so you can see the desktop. You should be able to see four shortcuts that allow you to start Remote Desktop sessions under the identities of different users. Click the shortcut with the caption of RemoteDesktop.msc to launch the Remote Desktop management console. Make sure the management console is maximized.
   2. Click the node **Angela Barbariol** and choose **Connect**.

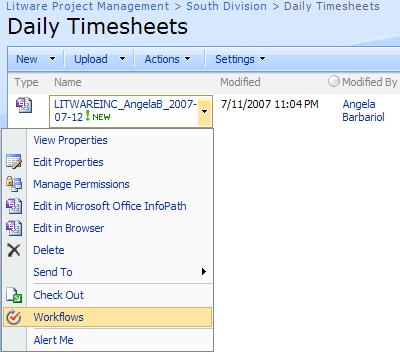


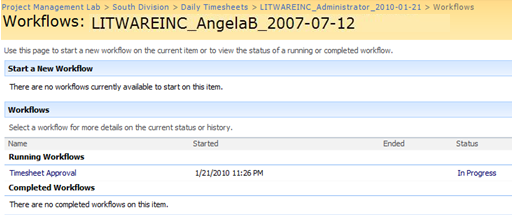
* 1. Enter the password **pass@word1**. This launches a remote desktop session for the user Brian Cox.
  2. You should notice that you have opened a remote desktop session on this computer for **AngelaB**.
  3. Open **Internet Explorer** and open the site at **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision** and navigate to the **Daily Timesheets** document library.
  4. Create a **New** Timesheet for Angela according to the following picture.

*Note: if you ran the setup instead of making the InfoPath lab, you will have a textbox instead of a dropdown in the Project column. In that case fill out the project name you see in the picture.*

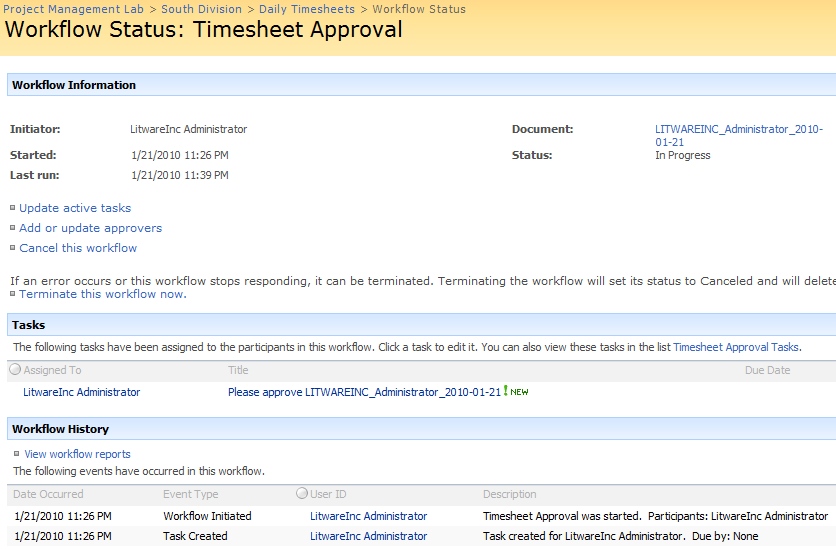


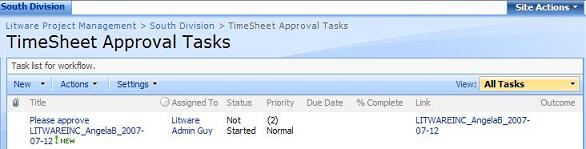
* + 1. Click on **Submit** for the timesheet.
    2. Then click **OK** to the completion message.
    3. Notice that the **TimeSheet Approval** for Angela's Timesheet is **In Progress** (the workflow automatically starts as a result of the document being created)
  1. Click on Angela's **Start** menu and select the **Log Off** choice.
  2. Click **OK** to the message that appears. This will allow us to act as the approver (i.e. Administrator).

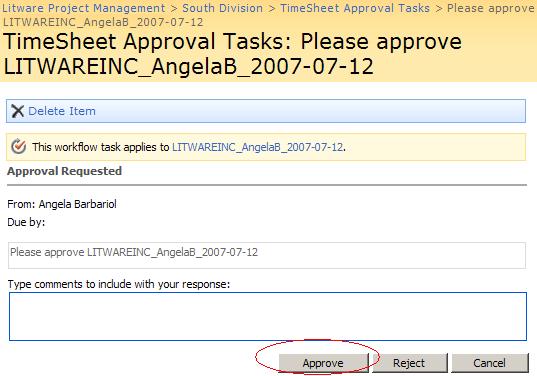
1. As the **Administrator**, navigate to the **Daily Timesheets** Form library in the **Litware Project Management - South Division** site.
   1. Select the **Workflow** menu item for the AngelaB timesheet in the document library. 
   2. Notice that the workflow is running with a status of **In Progress**.



* 1. Click on the **TimeSheet Approval** hyperlink and take a look at the workflow status. As part of the process the workflow has created a task for the user **LITWAREINC\Administrator** to resolve.



* 1. Navigate to the **Tasks** list to see the task that has been created by the workflow. 
  2. From the **South Division** site click on **View All Site Content**
  3. Select the **Tasks** list with a description of **Task list for workflow** that was auto-generated for us upon the creation of our workflow.

1. The next step in the workflow is to complete the task so the workflow can continue.
   1. On the **TimeSheet Approval Tasks** list edit the properties of the task in the list either by selecting **Edit Item** in the menu or clicking the item link.
   2. On the approval form, you can enter comments and either approve or reject the document. In this case **approve** the document by clicking the **Approve** button. 
2. The last step is to verify that the workflow has completed successfully and that the timesheet is approved. Navigate to the **Daily Timesheet** list on the South Division Site. You'll notice that the **Timesheet Approval** column for the document you approved is set to **Approved**. 

# Exercise 2: Create a sequential workflow with Visual Studio 2008

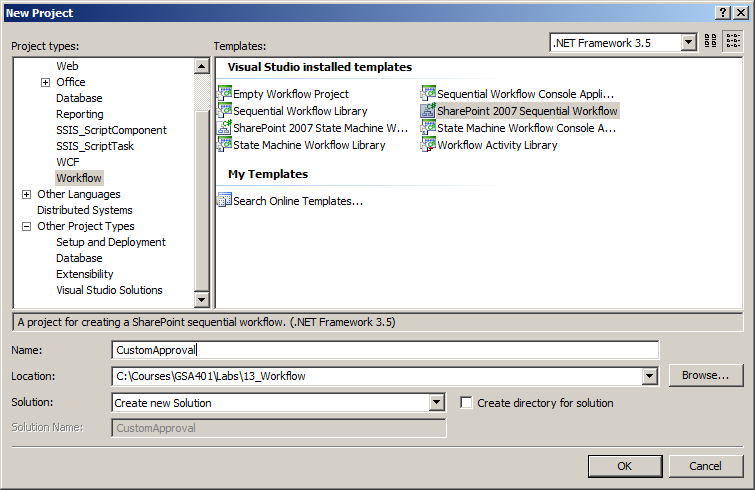
Even though SharePoint provides several workflow solutions, sometimes you need more functionality than is provided out of the box. In this case developers can create custom workflow solutions as well as custom InfoPath 2007 forms for managing workflow initialization and task completion. This exercise will help you to create a custom approval workflow that utilizes Visual Studio 2008 to create the workflow and InfoPath 2007 to create the user interface components.

**There are several Main Tasks in this lab**:

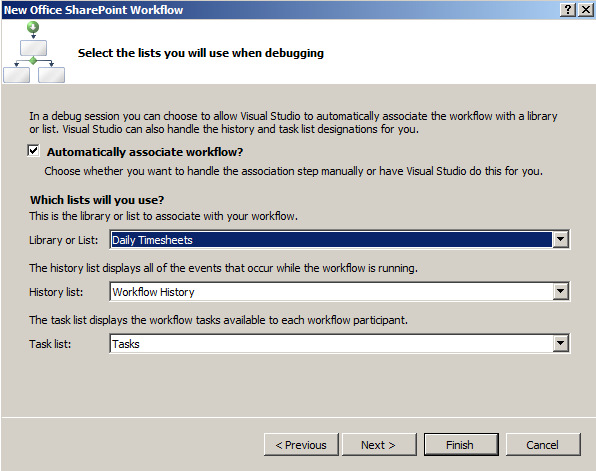
* Task A: Create the Basic SharePoint Workflow using the STSDEV utility
* Task B: Create and Publish the InfoPath Support Forms needed by the Workflow
* Task C: Modify the Basic workflow to make use of the InfoPath Forms
* Task D: Configure the workflow Feature for Deployment

## Task A: Create the Basic SharePoint Workflow

1. Start by creating a new project in **Visual Studio 2008** using the **SharePoint 2007 Sequential Workflow** template. Make sure the **.NET Framework 3.5** is selected. Give it the name **CustomApproval**. Save it in the **C:\Student\Labs\13\_Workflow\Lab** directory.



1. This starts the **New Office SharePoint Workflow** wizard.
   1. Fill out **Custom Approval** as name for the workflow.
   2. Fill out the URL to the SharePoint site **http://litwareinc.com/sites/ProjectManagementLab**.
   3. Click the **Next** button.
   4. Select the **Daily Timesheets** library as the library you want to associate with your workflow.
   5. Leave the two other selections (**Workflow History** and **Tasks**) as is.



* 1. Click the **Next** button.
  2. In the last step of the wizard ensure that the checkboxes for **Manually by users** and **When an item is created** are checked. Click the **Finish** button.

1. The first step in implementing the workflow is to retrieve the creation data information such as the manager's user name that needs to approve the document and any special comments associated with the document approval process.
   1. Start by adding three member fields to the **Workflow1.cs** or the **Workflow1.vb** class.
      1. View the code for **Workflow1.cs** by right clicking **Workflow1.cs** in **Solution Explorer** and selecting **View Code**.
      2. Add the following three member fields to the class as **strings: user, comments,** and **taskStatus**. Also, add a member field called **workflowId** as **Guid**.

C#

namespace CustomApproval  
{  
 public sealed partial class SequentialWorkflow01:   
 SharePointSequentialWorkflowActivity  
 {  
 public SequentialWorkflow01()  
 {  
 InitializeComponent();  
 }

public Guid workflowId = default(System.Guid);  
 public SPWorkflowActivationProperties workflowProperties =   
 new SPWorkflowActivationProperties();

public string user = default(string);  
 public string comments = default(string);  
 public string taskStatus = default(string);

VB.NET

Public Class Workflow1

Inherits SharePointSequentialWorkflowActivity

Public Sub New()

MyBase.New()

InitializeComponent()

End Sub

Public workflowProperties As SPWorkflowActivationProperties = \_

New Microsoft.SharePoint.Workflow.SPWorkflowActivationProperties

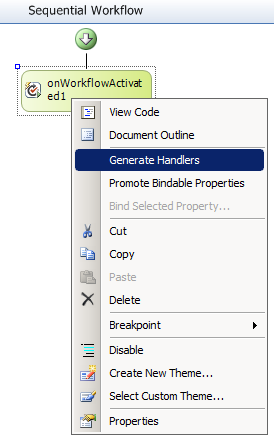
Public workflowId As Guid

**Public user As String**

**Public comments As String**

**Public taskStatus As String**

* 1. Next using your **Solution Explorer** open **Workflow1.cs** in designer mode.
  2. Right-click on **onWorkflowActivated1** activity and choose **Generate Handlers** to generate a handler for the **Invoked** event.



* 1. Implement the **onWorkflowActivated1\_Invoked** handler by storing the **workflowId** of the new SharePoint workflow in the internal field.

C#

private void onWorkflowActivated1\_Invoked(  
 object sender, ExternalDataEventArgs e)  
 {  
 // store the new workflow's id  
 workflowId = workflowProperties.WorkflowId;  
 }

VB.NET

Private Sub onWorkflowActivated1\_Invoked(ByVal sender As Object, \_

ByVal e As ExternalDataEventArgs)

**' store the new workflow's id**

**workflowId = workflowProperties.WorkflowId**

End Sub

1. Next you'll need to create a new task using the **CreateTask** activity. Start by creating several member fields in the **Workflow01.cs** or the **Workflow1.vb** class to store information about the tasks properties before and after editing. These will be used to retrieve the results of the task's completion. Use the code sample below to help you add the code to the class. Add these after your other Class member variables.

C#

public Guid taskId = default(System.Guid);  
public SPWorkflowTaskProperties taskProperties =  
 new SPWorkflowTaskProperties();  
public SPWorkflowTaskProperties beforeProperties =  
 new SPWorkflowTaskProperties();  
public SPWorkflowTaskProperties afterProperties =  
 new SPWorkflowTaskProperties();

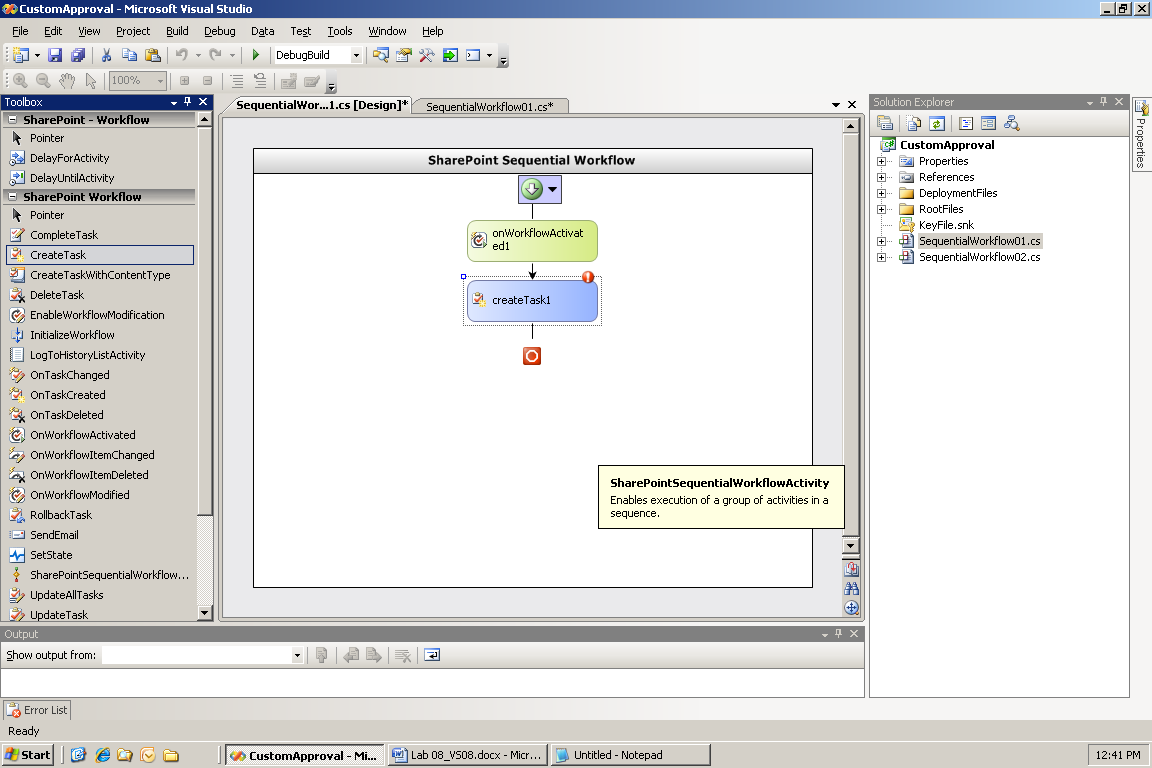
VB.NET

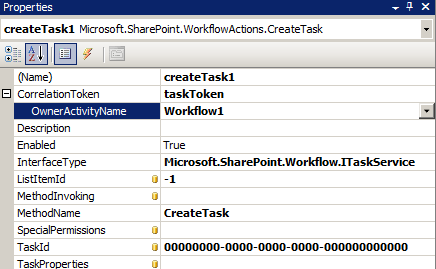
Public taskID As Guid

Public taskProperties As New SPWorkflowTaskProperties()

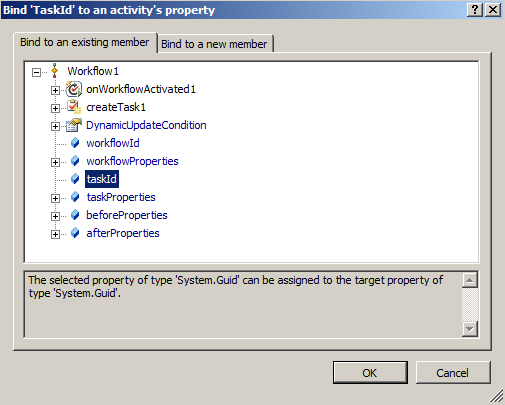
Public beforeProperties As New SPWorkflowTaskProperties()

Public afterProperties As New SPWorkflowTaskProperties()

* 1. Open **Workflow.cs** or **Workflow1.vb** in designer mode and drag a **CreateTask** activity onto the canvas after the activation activity. **Hint**: this item should be located in the **Toolbox** in the **SharePoint Workflow** section. 
  2. Now that the activity is created, you'll need to configure its **CorrelationToken** property:
     1. First, in the **Workflow1** Design Window select the **CreateTask1** item. Examine your properties window to ensure that it is now showing properties for **createTask1**.
     2. In the properties set the **CorrelationToken** to **taskToken** (i.e. type **taskToken** in the textbox)
     3. Next expand out the **CorrelationToken** (click the + to the left of **CorrelationToken**), and then set the **CorrelationToken's** **OwnerActivityName** to **Workflow1** (using the dropdown choices).



* + 1. Set its **TaskId** property to **taskId**—click on the ellipsis (…) and choose **taskId** from the dialog, then click **OK**.



* + 1. Follow those same instructions to set its **TaskProperties** property to **taskProperties**. These values are used by the activity to manage the lifetime of the workflow and to provide parameters for the task creation.
  1. Finally you'll need to populate the properties of the task using the **taskProperties** variable. To access the code, right click **createTask1** (in the **Workflow1** Design window) and click **Generate Handlers**. Using the code below, populate the task properties. The only property that is required is the **taskID** which was set in the designer.

C#

private void createTask1\_MethodInvoking(object sender, EventArgs e)  
{  
 // initialize the task ID  
 taskId = Guid.NewGuid();  
  
 // populate the properties of the task  
 taskProperties.AssignedTo = user;  
 taskProperties.Description = "Approve the document.";  
 taskProperties.Title = "Timesheet Approval";  
  
 // populate the type of the form and extended properties  
 taskProperties.ExtendedProperties["User"] = user;  
 taskProperties.ExtendedProperties["Comments"] = comments;  
}

VB.NET

Private Sub createTask1\_MethodInvoking(ByVal sender As Object, \_

ByVal e As EventArgs)

**' initialize the task ID**

**taskID = Guid.NewGuid()**

**' populate the properties of the task**

**taskProperties.AssignedTo = user**

**taskProperties.Description = "Approve the document."**

**taskProperties.Title = "Timesheet Approval"**

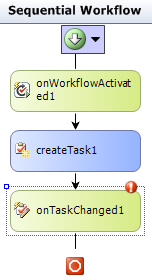
**' populate the type of the form and extended properties**

**taskProperties.ExtendedProperties("User") = user**

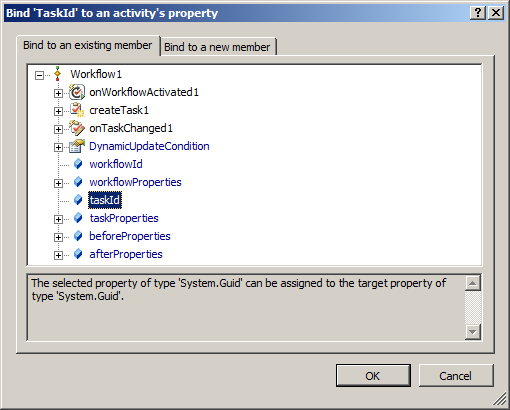
**taskProperties.ExtendedProperties("Comments") = comments**

End Sub

1. The previous step created a new task, now it's time to wait for the task to change and store the results of the task.
   1. Open **Workflow1.cs** or **Workflow1.vb** in designer mode and drag a **OnTaskChanged** activity onto the canvas after the **createTask1** activity.



* 1. Now that the activity is created, you'll need to configure its **CorrelationToken**.
     1. First, in the **Workflow1** Design window select the **OnTaskChanged1** item. Examine your properties window to ensure that it is now showing properties for **OnTaskChanged1**.
     2. In the properties set the **CorrelationToken** to **taskToken** (i.e. type **taskToken** in the textbox).
     3. Expand out the **CorrelationToken** property and set the **CorrelationToken's** **OwnerActivityName** to **Workflow1**.
     4. Then set its **TaskId** property to **taskId** by clicking the **…** button and choosing tasked from the treeview in the binding dialog.



* + 1. Set the **BeforeProperties** to **beforeProperties**, and its **AfterProperties** to **afterProperties** using the same way.

1. Finally you'll need to write some code associated with the **OnTaskChanged1** handler that will retrieve the status value from the after properties and store it in the member variable **taskStatus**.
   1. To access the code, right click **onTaskChanged1** (in the **Workflow1** **Design** window) and choose **Generate Handlers**.

C#

private void onTaskChanged1\_Invoked(object sender, ExternalDataEventArgs e)  
{  
 // retrieve the TaskStatus property from the infopath form  
 taskStatus = this.afterProperties.ExtendedProperties["TaskStatus"].ToString();  
}

VB.NET

Private Sub onTaskChanged1\_Invoked(ByVal sender As Object, \_

ByVal e As ExternalDataEventArgs)

**' retrieve the TaskStatus property from the infopath form**

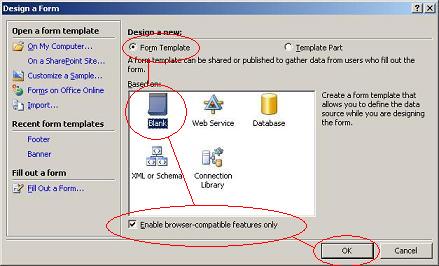
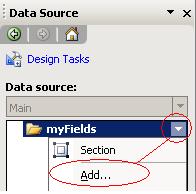
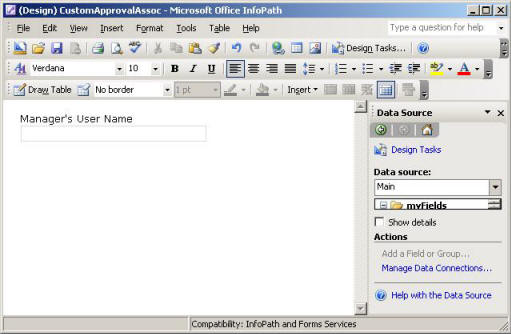
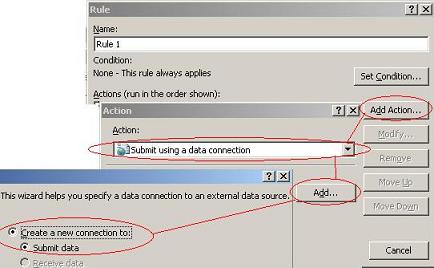
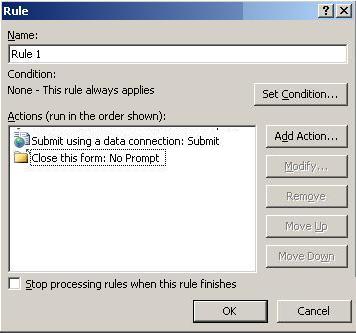
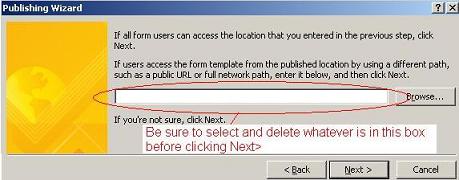
**taskStatus = Me.afterProperties.ExtendedProperties("TaskStatus").ToString()**

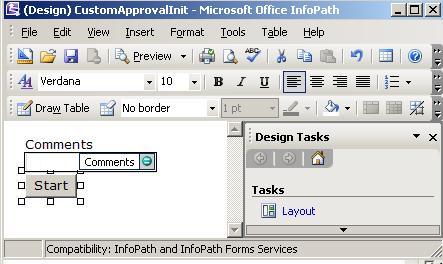
End Sub

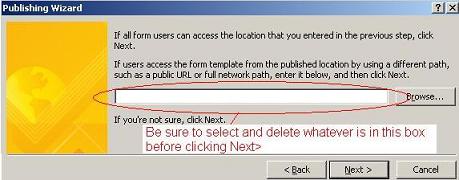
1. The last step in defining the workflow is to complete the task and set its status to the **TaskStatus** value retrieved in the previous step.
   1. Open **Workflow1.cs** or **Workflow1.vb** in designer mode and drag a **CompleteTask** activity onto the canvas after the **onTaskChanged1** activity.
   2. Once again, you'll need to set the **CorrelationToken** property to **taskToken** and then verify/set the **CorrelationToken's** **OwnerActivityName** to **Workflow1**. Set its **TaskId** property to **taskId**, its **TaskOutcome** to **taskStatus**. The **TaskOutcome** property is the text value that is used when setting the status value of the completed task.
2. Now that the workflow is complete, you'll need to sign the assembly and build the workflow assembly.
   1. To sign the assembly, navigate to the **Signing** tab on the project properties window, by right-clicking on **CustomApproval** in the Solution Explorer and clicking on **Properties**. Check the **Sign the assembly** checkbox, and use the dropdown to browse for the **CustomApproval.snk** file in the directory **C:\Student\Labs\13\_Workflow\Starter Files\Lab**.
   2. Save your changes and close the **Properties** page.
3. Finally **build** the assembly and fix any compile errors that may come up.

## Task B: Create and Publish the InfoPath Support Forms needed by the Workflow

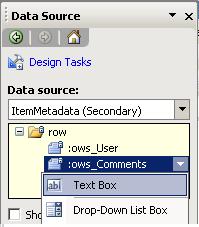
Now that you've finished the workflow, it's time to create the InfoPath 2007 forms that will be used by the workflow. This task is focused on creating these forms.

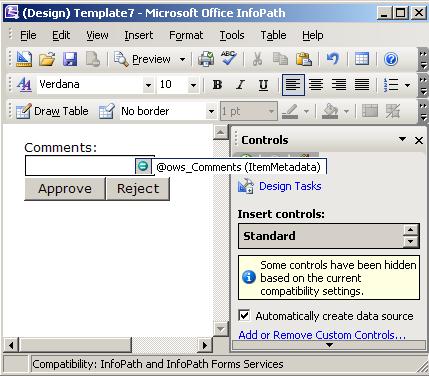
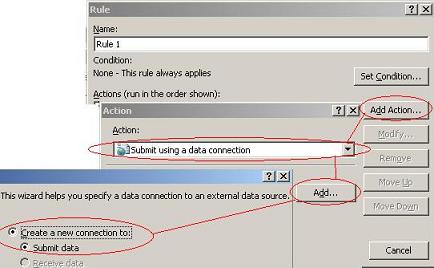
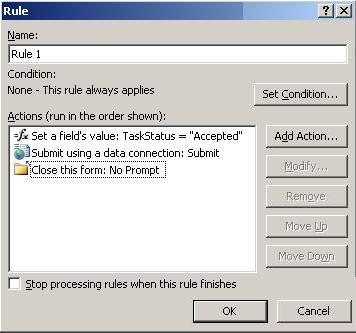
1. The first step in using a workflow is to associate one with a document library or list. In this form, you'll need to retrieve the manager's username to assign the approval tasks to.
   1. Start by opening InfoPath 2007 and designing a new blank form that is web enabled, by choosing **Design a Form Template** from the pop-up starter window and choosing the options in the picture below: 
   2. Select the **Data Source** view in the **Design tasks** pane and add two **string** fields named **User** and **Comments** nodes to the data source. 
   3. Next add the text **Manager's User Name** to the top of the form followed by a text box bound to the **User** data source field.
      1. **Hint**: After typing the text in go to the **Data Source** panel and click on the **User** field then click on the **dropdown arrow** and select **Text Box**. This will insert a textbox but you may need to remove the label **User**: 
   4. The last control to add to the form is the **Create** button that will submit the form back to SharePoint. Switch to the controls task pane and drag a button out onto the form.
   5. Right click on the new button and click **Button Properties**
      1. Change the button’s **Label** and **ID** to **Create**.
      2. Next click **Rules**... (in the following steps you will create two actions, **submit** and **close form**).
         1. Then click **Add - Add Action**... and select **Submit** **using a data connection**.
            1. Click **Add**... then **Create a new connection to: Submit data** and click **Next>**
            2. On the **Destination for submitting your data** page select **To the hosting environment**... and click **Next>** and **Finish**. Then click **OK**.
         2. Add another **action** that **closes the form** and click **OK**. 
         3. Click **OK** (3 more times)
   6. Set the Security for the Form
      1. In InfoPath on the **Tools** menu - select **Form Options** - select **Security and Trust** - Remove the check from **Automatically determine security level (recommended)** and select **Domain** ...
      2. Click **OK**
   7. Now that the form is done, save it to **C:\Student\Labs\13\_Workflow\Lab\CustomApprovalAssoc.xsn**. (**Note**: you will have to type **CustomApprovalAssoc.xsn** in the **File Name:** textbox.)
   8. Finally **Publish** the completed form to the **network location** 
      1. From the **File** menu select **Publish**... then specify **To a network location** and click **Next>**
      2. Click **Browse**... to specify the location to publish your form and use the path **C:\Student\Labs\13\_Workflow\Lab**\**CustomApproval\ CustomApprovalAssoc.xsn**  (**Note**: you will have to type in **CustomApprovalAssoc** in the **File Name:** textbox) and click **OK** and then **Next>**
      3. Delete the alternate access path. Then click **Next>**
      4. Click **OK** on the pop-up window.
      5. Click **Publish** and then click **Close** to complete publication.
2. Next you'll need to define the form that is displayed when a new instance of the workflow is started.
   1. Start by **Opening** the **C:\Student\Labs\13\_Workflow\Lab**\**CustomApprovalAssoc.xsn** template in **Design** mode and doing a **Save As...** to save it as **C:\Student\Labs\13\_Workflow\Lab**\**CustomApprovalInit.xsn**. This is done so the primary data source will have the same namespace in both forms.
   2. Next you will modify this form.
      1. Change the phrase **Manager's User Name** to **Comments** at the top of the form.
      2. Change the binding of the text box to the **Comments** data source field.
      3. Right-click on the **User** TextBox and select **Change Binding...**
      4. Change the field binding from **User** to **Comments** and click **OK**
      5. Then change the **Label** and **ID** of the button to **Start**, by double clicking on the button and setting the two fields.

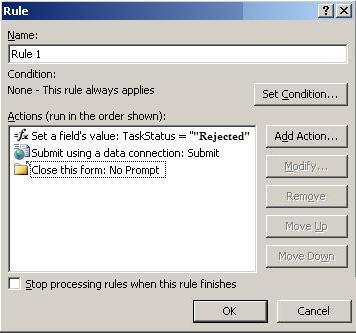


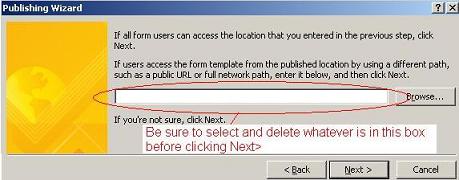
* 1. Set the Security for the Form
     1. In InfoPath on the **Tools** menu - select **Form Options** - select **Security and Trust** - Remove the check from **Automatically determine security level (recommended)** and select **Domain** ...
     2. Click **OK**
  2. Now that the form is done, save it (again to **C:\Student\Labs\13\_Workflow\Lab**\**CustomApprovalInit.xsn**).
  3. Finally **Publish** the completed form to the **network location** 
     1. From the **File** menu select **Publish**... then specify **To a network location** and click **Next>**
     2. Click **Browse**... to specify the location to publish your form and use the path **C:\Student\Labs\13\_Workflow\Lab**\**CustomApproval\CustomApprovalInit.xsn** and click OK and then Next>
        1. **Important**: As you have already published this form under a different name you **MUST** edit the **Form template path and file name**: and **Form template name**: to be **CustomApprovalInit** or you will overwrite your other form.
     3. Make sure that you remove alternate access path so that it is blank. Then click **Next>**
     4. Click **OK** on the Warning message that appears.
     5. Click **Publish** and then click **Close** to complete publication.

1. The last form to create is the form that is used when performing a task. In this form the manager will be presented with the comment entered in the Init form and two buttons, one to approve the document and another to reject the document.
   1. Start by opening InfoPath 2007 and design a new blank form that is web enabled.
   2. Select the **Data Source** view in the **Design Tasks** pane and add a **string** field named **TaskStatus** to the data source.
   3. Now you'll need to add a **Receive XML** data connection.
      1. Select the **Tools -> Data Connections** menu item.
      2. Click **Add** then **Create a new connection to: Receive Data.**
      3. Click **Next**.
      4. Then specify **XML document** for your data source and click **Next>**
      5. On the **XML data file details** page click **Browse**... then use the **C:\Student\Labs\13\_Workflow\Starter Files\Lab\ItemMetadata.xml** file to populate the data source. This data connection is used to provide the form with the initialization information created in the other forms.
      6. Leave the **Include the data as a resource file** checked.
      7. click **Next>** and then **Next>** again and **Finish**. Click **Close** to close your **Data Connections** window.
   4. Next add the text **Comments** to the top of the form followed by a text box bound to the **ows\_Comments** field in the **ItemMetadata** source. (Note: Remove the **OWS Comments**: label if necessary)



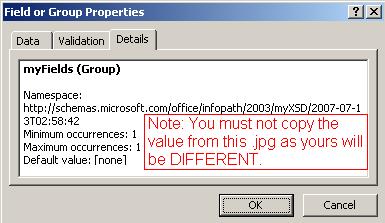
* 1. The last controls to add to the form are the **Approve** and **Reject** buttons that will submit the form back to SharePoint. In **Design** **Tasks**, switch to the **Controls** task pane and add two buttons onto the form. **Rename** the buttons and set their **ID** to **Approve** and **Reject** using the properties window. 
  2. Configure the actions on the **Approve** button by right clicking the button and selecting properties. Then add a new rule to the button.
  3. In that new rule, add three actions:
     1. First, add an action to **Set a field's value**. Use the **TaskStatus** field and set its value to **Accepted**.
     2. Second, on the Rule click **Add Action...** again and select **Submit using a data connection**.
        1. Click **Add...** then **Create a new connection to: Submit data** and click **Next>**
        2. On the **Destination for submitting your data** page select **To the hosting environment**... and click **Next>** and **Finish**. Then click **OK**.
     3. Thirdly, Add another **action** that **closes the form** and click **OK**. 
     4. Click **OK** (3 more times)
  4. Repeat the previous two steps (i.e. **F** and **G**) for the Reject button, but set the **TaskStatus** to **Rejected** instead of **Accepted**. Also, be sure to use the “Submit” connection that you already created for the **Approve** button for this button as well.



* 1. Set the Security for the Form
     1. In InfoPath on the **Tools** menu - select **Form Options** - select **Security and Trust** - Remove the check from **Automatically determine security level (recommended)** and select **Domain** ...
     2. Click **OK**
  2. Now that the form is done, save it to **C:\Student\Labs\13\_Workflow\Lab**\**CustomApprovalTask.xsn**.
  3. Finally **publish** the completed form to the **network location** using the path **C:\Student\Labs\13\_Workflow\Lab**\**CustomApproval\CustomApprovalTask.xsn.** When you publish, make sure that the alternate access path is blank 

## Task C: Modify the Basic workflow to make use of the InfoPath Forms

The final step in completing the workflow is to read the values entered in the **CustomApprovalAssoc** and **CustomApprovalInit** Forms. To do this you'll need to add a few lines of code to the **onWorkflowActivated1\_Invoked** method that will parse the XML in the **workflowProperties.InitiationData** property. This property is set using the XML document returned from the **CustomApprovalInit** form.

1. First you'll need the namespace for the document that was generated by Infopath. To get this namespace open **CustomApprovalInit** in design mode, by right-clicking on the file name and clicking on **Design**, and switch to the **Data Source** task pane.
   1. Next right click the **myFields** node in the **Main** data source and select **Properties**. Switch to the **Details** pane and copy the namespace from the text box. 
   2. Next return to your **Visual Studio CustomApproval** Workflow and **open** the **Workflow1.cs** in **Code view**.
   3. Add **using System.Xml;** to the using statements at the top of the code file.
   4. Then add the code below to the **onWorkflowActivated1\_Invoked** to initialize the internal user and comments fields.
      1. ***IMPORTANT****: MAKE SURE YOU* ***UPDATE*** *THE* ***NAMESPACE*** *IN THE CODE BELOW TO MATCH THE NAMESPACE YOU COPIED IN STEP* ***A*** *OR YOUR SOLUTION WILL NOT WORK!*

C#

// InitiationData now contains the data that came from the form.   
// We can store this data into local variables.  
XmlDocument doc = new XmlDocument();  
doc.LoadXml(workflowProperties.InitiationData);  
  
//BE SURE TO REPLACE THE FOLLOWING NAMESPACE WITH THE ONE YOU COPIED IN STEP A OR //THE SOLUTION WILL NOT WORK!!!

XmlNamespaceManager nsmgr = new XmlNamespaceManager(doc.NameTable);  
nsmgr.AddNamespace( "my",   
 "http://schemas.microsoft.com/office/infopath/2003/myXSD/2007-07-13T02:58:42");  
  
user = doc.SelectSingleNode("my:myFields/my:User", nsmgr).InnerText;  
comments = doc.SelectSingleNode("my:myFields/my:Comments", nsmgr).InnerText;

VB.NET

Dim doc As New Xml.XmlDocument()

doc.LoadXml(workflowProperties.InitiationData)

Dim nsmgr As New Xml.XmlNamespaceManager(doc.NameTable)

nsmgr.AddNamespace("my", \_

"http://schemas.microsoft.com/office/infopath/2003/myXSD/2007-07-13T02:58:42")

user = doc.SelectSingleNode("my:myFields/my:User", nsmgr).InnerText

comments = doc.SelectSingleNode("my:myFields/my:Comments", nsmgr).InnerText

## Task D: Configure the workflow Feature for Deployment

Now that the workflow assembly and the InfoPath 2007 forms are completed, you'll need to define the feature that will install the workflow.

1. In **Visual Studio Solution Explorer** open the **feature.xml**.
   1. Change the **Title** into **Litware Timesheet Approval Workflow Feature**.
   2. Inside the **ElementManifests** element, add a 3 **ElementFile** tags:

<ElementFile Location=**"**CustomApprovalAssoc.xsn**"** />

<ElementFile Location=**"**CustomApprovalInit.xsn**"** />

<ElementFile Location=**"**CustomApprovalTask.xsn**"** />

* 1. Notice that the feature references the **Microsoft.Office.Workflow.Feature.dll** as entry point for this workflow.
  2. Notice also the **<Properties>** element that contains the following children:

**<Properties>**

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

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

**</Properties>**

1. Open the **workflow.xml** file and follow these instructions to fill in the template.
   1. Find the **Workflow** element named **Workflow1** and set its attributes to these values:
      1. **Name= " Litware Timesheet Approval Workflow "**
      2. **Description="This workflow does a whole bunch of great stuff..."**
      3. **AssociationUrl="\_layouts/CstWrkflIP.aspx"**
      4. **InstantiationUrl="\_layouts/IniWrkflIP.aspx"**
      5. **ModificationUrl="\_layouts/ModWrkflIP.aspx"**
      6. **TaskListContentTypeId="0x01080100C9C9515DE4E24001905074F980F93160"**
   2. Inside the **<Workflow>** element, find the **<MetaData>** element with its child elements:

<Workflow Name="Litware Custom Timesheet Approval". . .>

<MetaData>  
 <Association\_FormURN></Association\_FormURN>  
 <Instantiation\_FormURN></Instantiation\_FormURN>  
 <Task0\_FormURN></Task0\_FormURN>  
 <AssociateOnActivation>false</AssociateOnActivation>  
 </MetaData>

</Workflow>

* 1. Remove the **<StatusPageUrl>** element.
  2. Populate the **MetaData** section with InfoPath 2007 form URNs for the **Assoc**, **Init**, and **Task** forms we created earlier.
     1. To find the URN, open each InfoPath 2007 form in **Design** mode by doing **Right Click > Design** on the file name, and select **File -> Properties** in the menu. **Copy** the **ID** field.
     2. When finished your MetaData Element should look similar to the code below, except with different URNs and with the Workflow attributes included. After examining your modifications **Save** and **Close** the **WorkflowTemplates.xml** file. **NOTE:**  Ensure that there is **NO** leading or trailing whitespace inside the **Association\_FormURN**, **Instantiation\_FormURN**, or **Task0\_FormURN** tasks or you will get an error when the form is viewed in SharePoint.

<Workflow Name="Litware Custom Timesheet Approval". . .>

<MetaData>  
 <Association\_FormURN>

urn:schemas-microsoft-com:office:infopath:CustomApprovalAssoc:-myXSD-2007-07-13T02-58-42

</Association\_FormURN>

<Instantiation\_FormURN>

urn:schemas-microsoft-com:office:infopath:CustomApprovalInit:-myXSD-2007-07-13T02-58-42

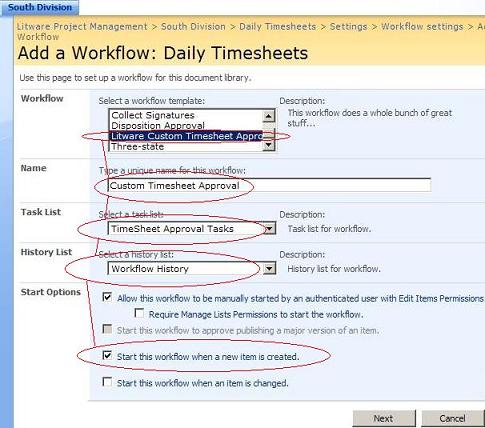
</Instantiation\_FormURN>  
 <Task0\_FormURN>

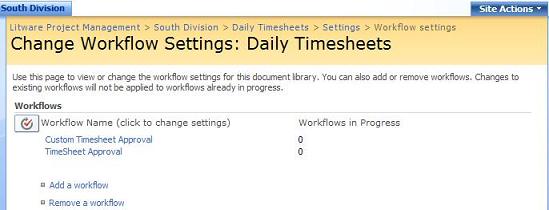
urn:schemas-microsoft-com:office:infopath:CustomApprovalTask:-myXSD-2007-07-13T06-48-40

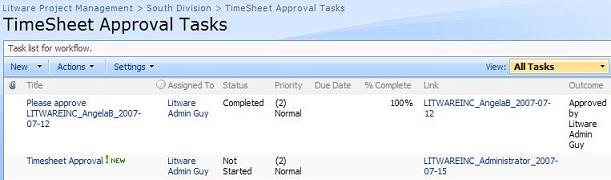
</Task0\_FormURN>  
 <AssociateOnActivation>false</AssociateOnActivation>  
 </MetaData>

</Workflow>

1. **Build** your project and correct any errors.
2. There are still 3 files to copy from the **C:\Student\Labs\13\_Workflow\Starter Files\Lab** directory to the project directory. Take your time to inspect them:
   1. Manifest.xml
   2. Wsp\_structure.ddf
   3. install.bat
3. **Build** your **CustomApproval** project.
4. Open **Windows Explorer** and double-click the **intall.bat** file to deploy your custom approval workflow solution.
5. Now that all the preliminary work is done, it's time to activate the feature and associate the **Custom Timesheet Approval** workflow with the **Daily Timesheet** document library.
   1. Open the site at **http://Litwareinc.com/sites/ProjectManagementLab** and choose **Site Settings** from the **Site Actions** menu.
   2. Click the link for **Site Collection Features** under the **Site Collection Administration** section.
   3. Find the feature named **Litware Timesheet Approval Workflow** and click its **Activate** button.
   4. Open the site at **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision** and navigate to the **Daily Timesheets** document library.
   5. Open the Form library settings by using the **Settings -> Form Library Settings** menu item.
      1. On the Form library settings page, navigate to **Workflow Settings** in the **Permissions and Management** section.
      2. If you are routed to the **Change Workflow Settings** screen, click the **Add a Workflow** link to associate the workflow with the document library. If you are not routed to this page, then move on to the next step. [Note: You will only be routed to the **Change Workflow Settings** screen if there is already another workflow associated with this document library.]
      3. Select the **Litware Custom Timesheet Approval** workflow template and name the workflow **Custom Timesheet Approval**.
      4. Select the **Timesheet Approval Tasks** list from the **Tasks List** dropdown. This list was automatically created as part of the original Timesheet Approval workflow in the **Select a task list:** drop down (or select to just create a new one), and select the **Workflow History** choice from the **Select a history List:** dropdown.
      5. Place a check in the **Start this workflow when a new item is created** box



* + 1. Finally click the **Next** button to move to the next step.
  1. This next page utilizes your **CustomApprovalAssoc** InfoPath Form to set the Managers user name.  Enter **LITWAREINC\Administrator** so the administrator must approve the document then click **Create** to complete the workflow creation. 

1. You've successfully created the **Custom Timesheet Approval** workflow and associated it with the document library.  Now we need to remove the original workflow.
   1. From your **Change Workflow Settings** page select **Remove a workflow**.
   2. On the **Remove Workflows** page specify **No New Instances** for the **TimeSheet Approval** workflow and click **OK**.
2. Now all that's left to do is test the approval process.
   1. In the **Remote Desktop** console right-click **Angela Barbariol** to open a Remote Desktop connection.
   2. On Angela's desktop, open the site at **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision** and navigate to the **Daily Timesheets** document library.
   3. Create a **New Daily Timesheet**, and fill it in with sample values and click **Submit**.
   4. The workflow automatically starts.  You should see **In Progress** underneath the **Custom Timesheet Approval** column on the **Daily Timesheets** form library.
   5. This workflow created a task for the user **LITWAREINC\Administrator** to resolve.  In order to complete the workflow we must shut down Angela's RDP session and once again become the Administrator.
      1. Go to Angela's **Start** menu select **Log off** and click **OK**
   6. As the Administrator navigate to the **Tasks** list to see the task that has been created by the workflow.
      1. Navigate to **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision**
      2. Select **View All Site Content**
      3. Select the **Tasks** List that you selected for this workflow. 
3. The next step in the workflow is to complete the task so the workflow can continue.
   1. Click on the **Timesheet Approval** task in the list.
   2. On the approval form, you can either **Approve** or **Reject** using the appropriate buttons. 
4. The last step is to verify that the workflow has completed successfully and that the timesheet is approved. Navigate back to the **Daily Timesheets** form library.  You'll notice that the **Custom Timesheet Approval** column for the document you approved is set to **Completed**. 

# Student Challenge: Conditional branching in a workflow

If a timesheet is approved, the Regional manager needs to make sure the consultant is paid.  Extend the workflow using the **Windows Workflow** **If-Else** activity to create a new task for the Regional manager only if the **TaskStatus** is set to **Accepted**.  Update the workflow and redeploy the feature then test your changes.  You will know you've implemented it correctly, if the manager approves the timesheet and a new task is created for the Regional manager.  Use **LITWAREINC\JayH** as the Regional manager.