## Developing Workflows

**Lab Time**: 45 minutes

**Lab Folder**: C:\Student\Labs\Workflow

**Lab Overview**: The SharePoint 2010 platform has a new level of integration between the different workflow development tools. Visio 2010, SharePoint Designer 2010, and Visual Studio 2010 can all work together to create a complete workflow. Each of these tools has its strengths. By combining all three to create a complete workflow, power users and developers can work together to structure an organization’s processes. Visio 2010 is used to create the high level process and is focused on the visual aspects of the process rather than the details. Visio exports this model to SharePoint Designer 2010 which is used to fill in the details. At this point the workflow can be deployed and used within the organization. Visual Studio 2010 is used when a full application is needed in a single managed install package. The existing workflow created in Visio 2010 and SharePoint Designer 2010 can be exported to a WSP file and imported into Visual Studio. This workflow can then be integrated into a larger solution and deployed in a single SharePoint solution package.

In this lab exercise you will implement a simple timesheet approval workflow using Visio 2010 and SharePoint Designer 2010. The initial process layout will be implemented in Visio 2010 and imported into SharePoint Designer. The process will check the number of hours that were submitted for the week and request an approval process if they are over a specified threshold. Once this process is working, it will be exported to a WSP and imported into Visual Studio 2010 for extensions that require user code. Using Visual Studio 2010, the workflow will be integrated into a larger solution and deployed as a single packaged WSP.

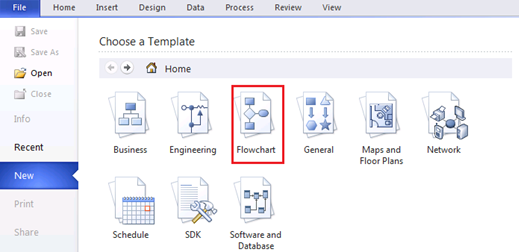
Lab Setup Requirements

* Before you begin this lab, you must run the batch file named **SetupLab.bat** located inside the lab folder. This batch file creates a new blank site collection at the location **http://intranet.wingtip.com/sites/Workflow**. This is the site you will use to test and debug the code you are going to write with the Visual Studio 2010 SharePoint Tools.

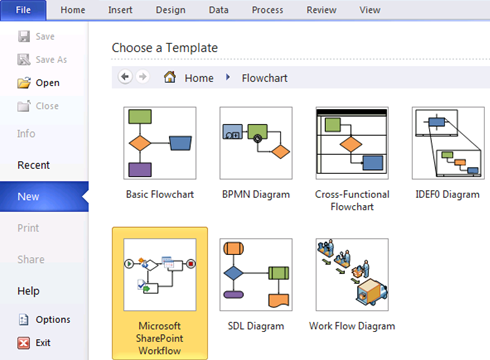
### Exercise 1: Building workflows with Visio 2010

In this first exercise you will create a simple process in Visio 2010 and export it to SharePoint Designer 2010.

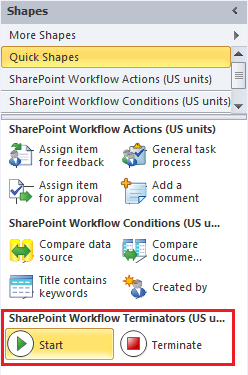
1. Launch **Visio 2010** and select the **FlowChart** folder.



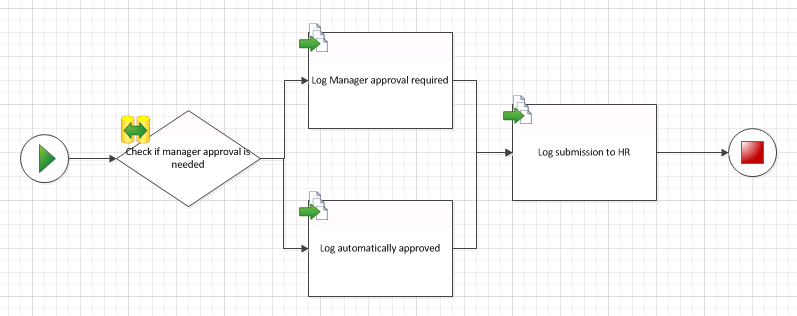
1. Within the **FlowChart** folder create a new file using the **Microsoft SharePoint Workflow** template.



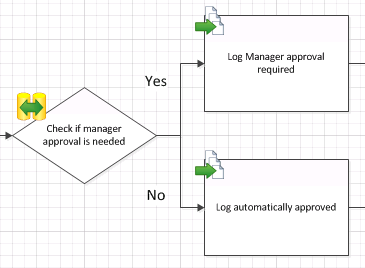
1. Click the **Create** button in the right pane.
2. Using the toolbox on the left hand side of the window, locate the **SharePoint Workflow Terminators** section and add a **Start** and **Terminate** shape to the diagram.



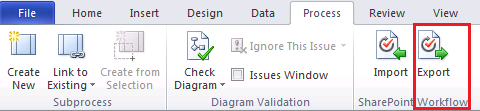
1. Open the **SharePoint Workflow Conditions** section and add a **Compare Data Source** shape to the diagram.
   1. Double click and change its name to **Check if manager approval is needed**.
2. Open the **SharePoint Workflow Actions** section and add three **Log to History List** shapes to the diagram.
3. Double click the first and change its name to **Log Manager approval required**.
   1. Double click the next and change its name to **Log automatically approved**.
   2. Double click the last and change its name to **Log submission to HR**.
4. Add the process flow connections between the shapes. Use the image below as an example.
5. Add connections by hovering over the source shape, clicking and dragging one of the arrows to the target shape.



1. Label the connections flowing from the condition yes or no depending on the process flow.
2. Right click the connection from **Check if manager approval is needed** to **Log Manager approval required** and select **Yes**.
3. Right click the connection from **Check if manager approval is needed** to **Log automatically approved** and select **No**.



1. Save the process diagram to [[LAB FILES]]\Solution with a name of TimeSheet Submission Workflow.vsd.
2. Export the process by selecting the **Process** ribbon in Visio 2010 and clicking the **Export** button.



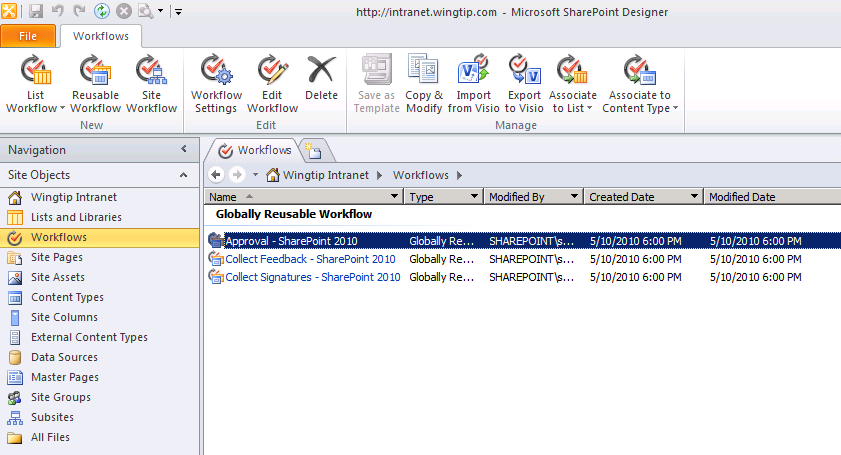
1. Save the exported workflow to [[LAB FILES]]\Solution with a name of TimeSheet Submission Workflow.vwi.

In this exercise you created a SharePoint workflow using Visio 2010 for use in future lab exercises.

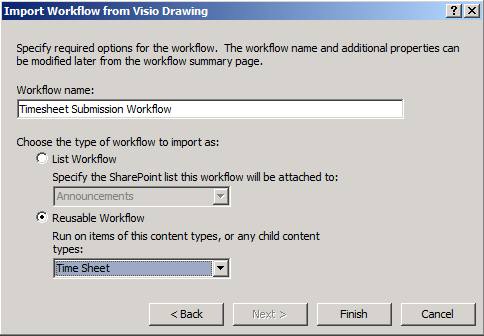
### Exercise 2: Building workflows with SharePoint Designer 2010

In this exercise you will import the Visio workflow into SharePoint Designer and complete it. This includes defining the parameters of each step as well as adding some new steps. In the end you will publish your complete workflow and export it back to Visio.

1. Launch **Microsoft Office SharePoint Designer 2010** and open the lab site at http://intranet.wingtip.com/sites/Workflowusing the **Open Site** button on the main page.
2. Click the **Workflows** item on the **Navigation** links on the left hand side of the window to display the current Workflows.

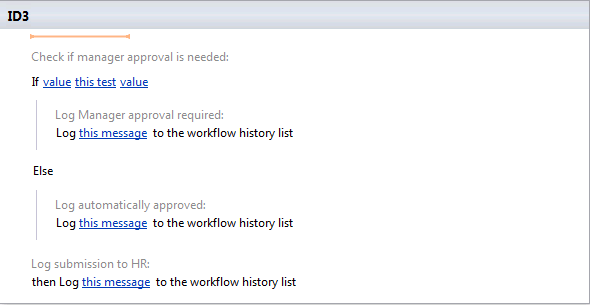


1. Click the **Import from Visio** button in the ribbon and select the \*.vwi file at [[LAB FILES]]\Solution\TimeSheet Submission Workflow.vwi then click **Next**.
2. Verify the **Name** is **Timesheet Submission Workflow**.
3. Choose the **Reusable Workflow** option and select the **Time Sheet** content type.

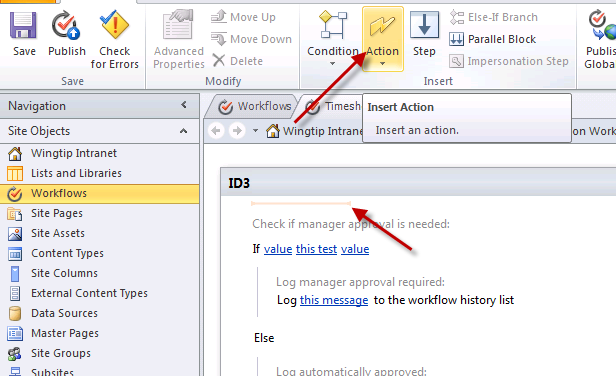


**Note:** If you don’t have **Time Sheet** as an option in the **Reusable Workflow** selector, you likely did not run the lab setup file or there was an error. Do not select the **Time Card** content type.

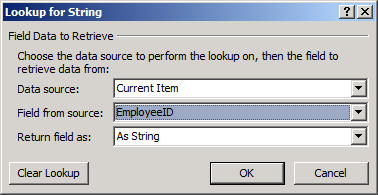
* 1. Click **Finish** to import the workflow. This can take a while.



1. Add two new local variables to store the maximum hours before approval is required and the employee name and set it. For now we’ll store the employee id instead of the name. We’ll look up the employee name in a later exercise.
2. Click the **Local Variables** button in the **Workflow** ribbon.
3. In the dialog, click the **Add** button and create a new variable with a name of **EmployeeName**.
4. Set the **Type** to **String**.
5. Click **OK** on all dialogs to create the variable.
6. Repeat the above steps to create a variable named **MaxHours** of type **Integer**.
7. Store the **EmployeeID** as the employee’s name. This will be replaced later with the employee’s real name requested from the HR system.
8. Place the selector at the top of the main step.
9. Click the **Action** button in the **Workflow** ribbon and select **Set Workflow Variable**.

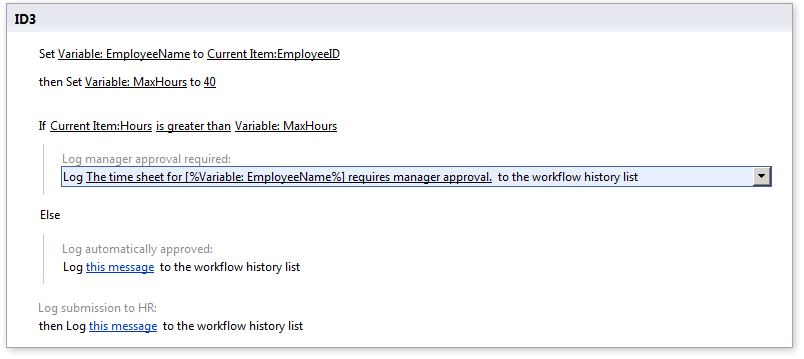


1. Click the **workflow variable** link and select **Variable: EmployeeName**.
2. Click the **value** link and then click the **fx** button.
3. In the dialog select **EmployeeID** in the **Field from source** drop down list.



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

1. Store **40** in the **MaxHours** variable.
2. Place the selector just below the action you just added, right before the first **If** statement.
3. Click the **Action** button in the Workflow ribbon and select **Set Workflow Variable**.
4. Click the workflow variable link and select **Variable: MaxHours**.
5. Click the value link type in **40**.
6. Define a condition that will be true if the number of hours worked is greater than **MaxHours**.
7. Click the first **value** link in the **If** statement then click the **fx** button to assign a value.
8. Leave **Data Source** as **Current Item** and select **Hours** in the **Field from source** drop down list. Click **OK** when you are done.
9. Click this test and select **is greater than**.
10. Click the second **value** link in the **If** statement then click the **fx** button to assign a value.
11. Set **Data Source** to **Workflow Variables and Parameters**.
12. Set **Field** from source to **Variable: MaxHours**.
13. Click **OK** to close the dialog.
14. Define the message logged when a timesheet requires manager approval. You’ll be performing a lookup in the Employees list to find the name of the employee later.
15. Click the **this message** link on the first **Log** action that immediately follows the **If** condition.
16. Click the **…** button to define the text of the logged message.
17. Enter the text **The time sheet for**.
18. Click **Add or Change Lookup** to retrieve information from the environment.
19. Select **Workflow Variables and Parameters** in the **Data Source** drop down list.
20. Select **Variable: EmployeeName** in the **Field from source** drop down list.
21. Click **OK** to close the dialog.
22. Finish the message by entering **requires manager approval**.
23. Click **OK** to close the dialog.

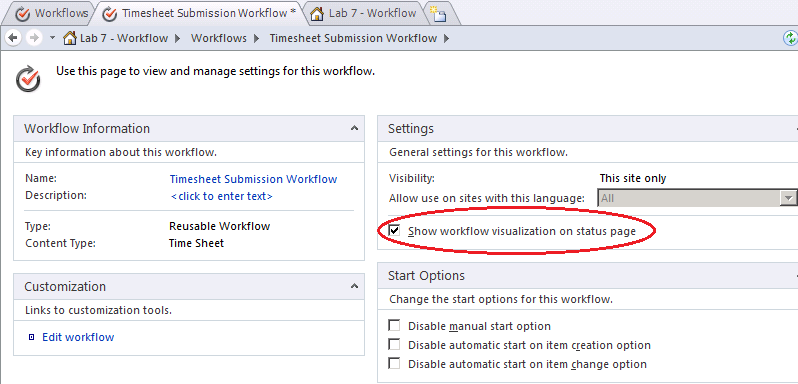


1. Repeat the previous step for the second **Log** action with the following exceptions
2. When finishing the message instead of entering requires manager approval, enter does not require manager approval.
3. Log that the time sheet has been approved and submitted to HR. It hasn’t actually been submitted at this point, that will happen in a later exercise.
4. Click the **this message** link on the final **Log** actions.
5. Click the **[…]** button and enter the message **The time sheet has been submitted to HR**.
6. Click **OK** to close the dialog.
7. Finish the workflow’s final settings and publish it to the SharePoint site.
8. Click the first **Timesheet Submission Workflow** item on the workflow header to return to the workflow’s home page.

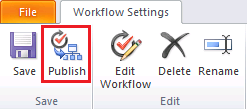


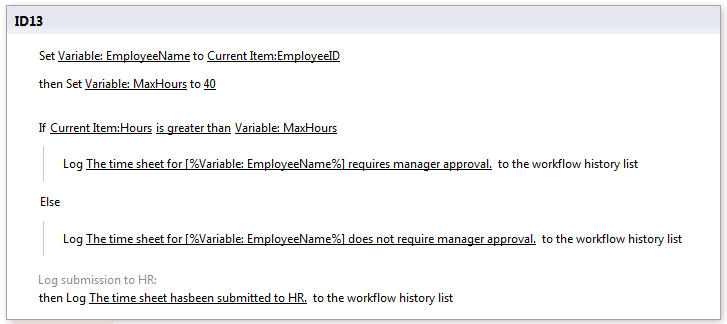
1. Check the **Show workflow visualization** on status page check box (found in the **Settings** group).

If the **Show workflow visualization** option is either disabled or does not appear, ensure the following features are activated: **Visio Web Access** (at the **farm** level… do this from within **Central Administration » System Settings » Manage Farm Features**), **SharePoint Server Enterprise Site Collection Features** (at the **site collection** level… do this from within the site’s **Site Settings** page) and **SharePoint Server Enterprise Features** (at the **site** level… do this from within the site’s **Site Settings** page).



* 1. Click the **Save** button to save the workflow settings.
  2. Click the **Publish** button in the **ribbon** to publish the workflow to the SharePoint site.





* 1. Click the **Associate to Content Type** button in the ribbon and select **Time Sheet**.

If the list of content types is blank when clicking this button, you may need to **Save** the workflow, and then click the **Content Types** section in the left-hand **Navigation** section of **SharePoint Designer 2010** to refresh the content types. Then you can go back to the workflow you created and repeat this step.

If this work around does not resolve the issue, you can create the association through the browser. In the browser navigate to the **http://intranet.wingtip.com/sites/workflow** site and select Site Actions » Site Settings » Site Content Types. Then find the **Time Sheet** content type, select it & on the following page click **Workflow Settings**. On the Workflow Settings page click Add Workflow. Select the workflow you just created and published (**TimesheetSubmissionWorkflow**) and give it a name of **Timesheet Submission Workflow** and click **OK**.

At this point the workflow has been associated with the Time Sheet content type.

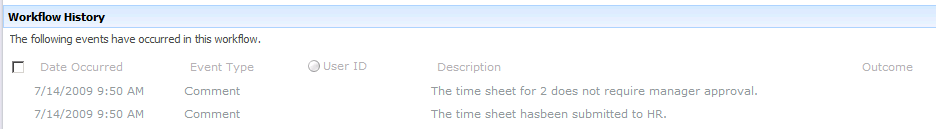
* 1. When the browser window loads, scroll to the bottom and click **OK**, next click **Save**.

1. Return to the Internet Explorer page pointing to **http://intranet.wingtip.com/sites/Workflow**.
2. Click the link in the quick launch bar for **Time Sheets**.
3. Switch to the **Documents** ribbon and click the drop down under the **New Document** button and select Time Sheet to create a new document.

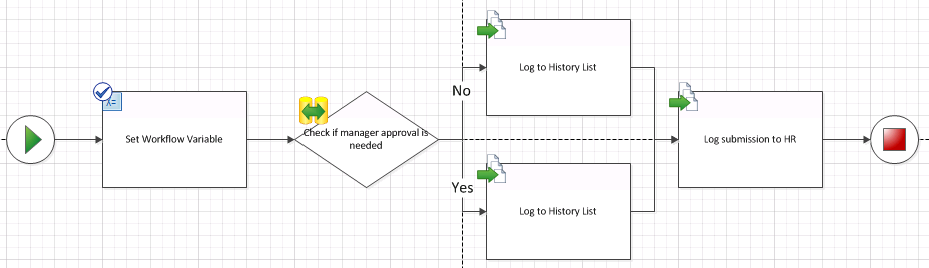
If Microsoft Word successfully opens the document then proceed with step 15.

Otherwise you will need to create a new document using Word (not by launching it from the browser) and upload the file manually to SharePoint. After doing this update the new item in the document library by modifying its properties like a regular list item to the values listed in step 15. Then you can jump to step 16.

1. Save the new document to the **Time Sheets** document library.
2. When saving the document, set the **EmployeeID** to **1**.
3. Set the **Hours** to **38**.
4. In the document library select the new document’s drop down menu and click **Workflows**.
5. Click **Timesheet Submission Workflow** to start the new workflow.
6. On the **Initiation** page click **Start** to start the workflow.
7. When the workflow has started, navigate back to the **Time Sheets** list and click the **Completed** link on the item.
8. View the **Workflow Information** page and see that the steps in the workflow were completed.
9. Repeat the process if you’d like using a Time Sheet that has hours that are greater than 40. Keep in mind that you’ll now have to approve the task that is created.



1. Since the process was changed by your work in SharePoint Designer, export those changes back to Visio to keep the process models in sync.
2. Switch back to **SharePoint Designer** and verify the workflow is still in the active window.
3. Click the **Export to Visio** button in the ribbon.
4. Save the updated \*.vwi file as Timesheet Submission Workflow.vwi inside the lab folder.
5. Switch back to **Visio** and import the update .vwi file.
6. Click the **Import** button in the **Process** ribbon tab.
7. Select the Submission Workflow.vwi file from your student folder.
8. The Visio diagram should now be updated to include the **Set Workflow Variable** tasks added in **SharePoint Designer**.

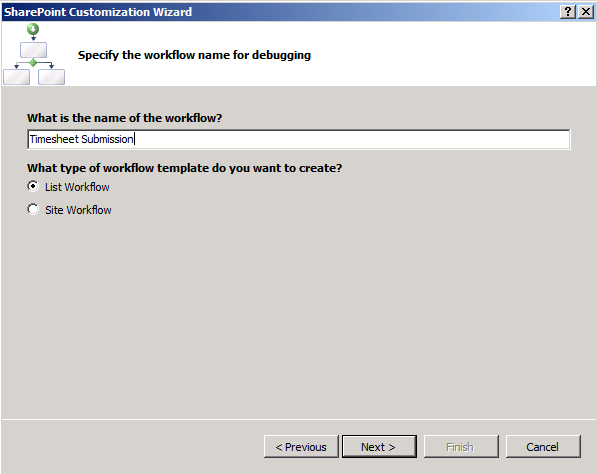


In this exercise you imported a workflow created in Visio 2010 into SharePoint Designer, made a few changes and then deployed it to SharePoint. You then sync’d those changes with the original Visio diagram.

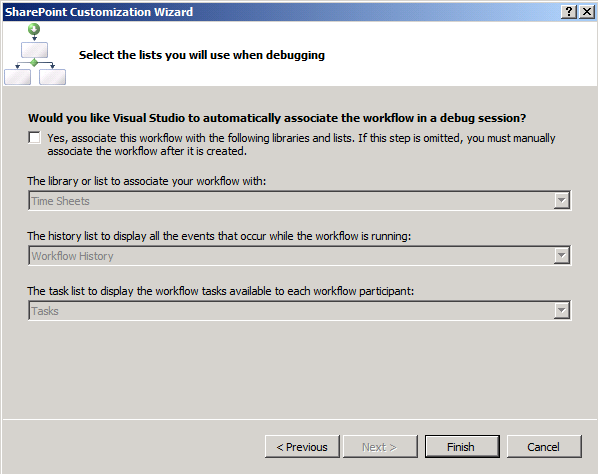
### Exercise 3: Developing Workflows with Visual Studio 2010

In this exercise you will build a workflow in Visual Studio 2010 that submits the timesheet information to the back end HR system and places the payment reference back into the original time sheet.

1. Open **Visual Studio 2010** and create a new workflow project:
2. Select the **Sequential Workflow** template in the **Visual C# » SharePoint » 2010** templates folder and give it the name **SubmissionWorkflow**.
   1. Verify the URL. The workflow should be deployed to **http://intranet.wingtip.com/sites/Workflow**. Indicate that the workflow must be deployed as a farm solution.
   2. In the wizard set the workflow name to **Timesheet Submission** and select the **List Workflow** radio button. Click the **Next** button.



* 1. Clear the check box indicating a **workflow association should be created on deployment**. An association will be created manually to test the association page.



* 1. Click the **Finish** button.

1. Add a new **Code Activity** from the **Windows Workflow 3.0** section of the toolbox and add it immediately following the **onWorkflowActivated1** activity.
2. Set the name of the new code activity to **SubmitToHR** using the **Properties** window.
3. Double-click the activity to generate the **ExecuteCode** event handler.
4. Add the following code to simulate the calculation of a payment reference using a GUID, to be sure to have a unique reference. Save this reference in the time sheet on which the workflow runs:

private void SubmitToHR\_ExecuteCode(object sender, EventArgs e)

{

SPListItem item = workflowProperties.Item;

item["Payment Ref"] = Guid.NewGuid().ToString();

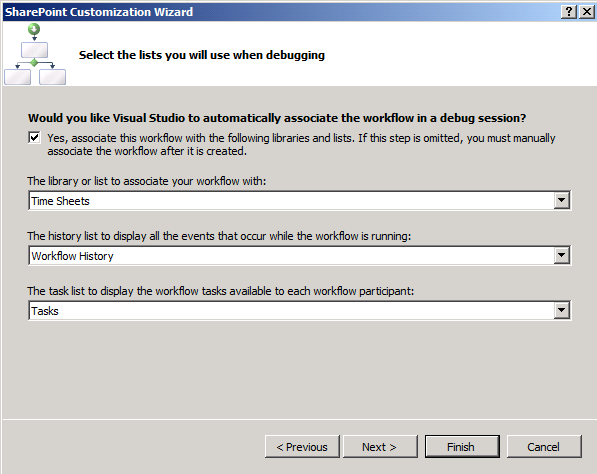
item.Update();

}

1. At this time you are going to deploy the workflow.

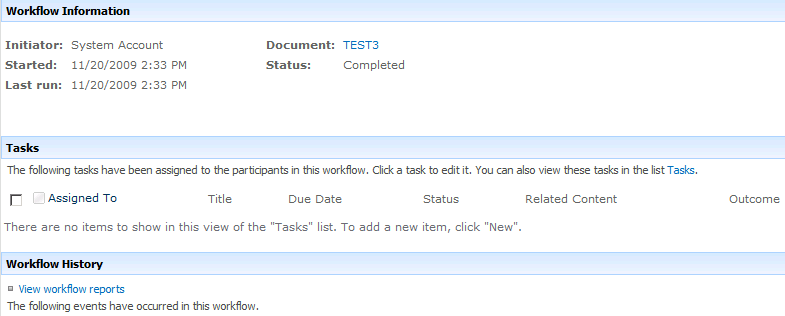
Setup the auto association parameters that will tell Visual Studio to automatically associate your workflow with the Time Sheets list when it is deployed.

1. Select Workflow1 in the solution explorer.
2. In the **Properties** window, set the Auto Associate property to True.
3. In the History List property click the [...] button to start the wizard.
4. On the next page select a list of **Time Sheets** in the first dropdown and click **Next**.



* 1. On the final page check the first check box and clear the rest and click **Finish**.

1. Deploy the workflow to SharePoint and verify that it works:
2. Right click the project in **Solution Explorer** and click **Deploy**.
3. When the deployment is complete, open **Internet Explorer** and navigate to **http://intranet.wingtip.com/sites/Workflow/Lists/TimeSheets**.
4. Click the drop down menu on one of the documents in the list and select **Workflows**.
5. Click the **Timesheet Submission** link to start the workflow.
6. When the workflow is complete verify the **Payment Ref** was assigned and then click the **Completed** link.
7. Verify the workflow history.



In this exercise you created a workflow using Visual Studio 2010.

### Exercise 4: Adding Association Forms in Visual Studio 2010

In this exercise you will add an association form to the timesheet submission workflow. This form will allow you to configure an hourly rate.

1. Add a new association form to the **Workflow1** workflow.
2. Right click on the Workflow1 folder and click **Add » New Item**.

**Note:** Make sure that you select the workflow node and not the project node because you will be able to add an association form but it will never be associated with the workflow.

* 1. Select **Workflow Association Form** from **Visual C# » SharePoint » 2010**, name it AssociationForm.aspx and click **Add**.
  2. In the <asp:Content> control with the ID of Main, add the following markup to create the page.

<asp:Content ID="Main" ContentPlaceHolderID="PlaceHolderMain" runat="server">

<table>

<tr>

<td>Hourly rate:</td>

<td><asp:TextBox ID="HourlyRateTextBox" runat="server" /></td>

</tr>

</table>

<asp:Button ID="AssociateWorkflow" runat="server" OnClick="AssociateWorkflow\_Click" Text=”Associate Workflow” />

* 1. Right click the markup window and choose **View Code**.
  2. Add the following code to GetAssociationData to save the max hours settings to the association data.

private string GetAssociationData()

{

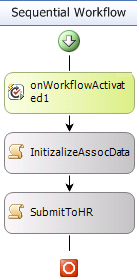
double rate = 0.0;

double.TryParse(HourlyRateTextBox.Text, out rate);

return rate.ToString();

}

1. Add a new code activity to the workflow to parse the association data and calculate the amount to pay.
2. Open Workflow1 in the designer.
3. Drag a new code activity from the toolbox and drop it between the **onWorkflowActivated1** activity and the **SubmitToHR** activity.
4. Rename the activity to **InitializeAssocData**.



* 1. Double-click the **InitializeAssocData** activity to generate the event handler.
  2. Add a double field named amountToPay just above the event handler.

private double amountToPay = 0.0;

private void InitizalizeAssocData\_ExecuteCode(object sender, EventArgs e)

{

}

* 1. Add the following code to retrieve the association data from the association form and to calculate the due amount based on the hours filled out on the time sheet and the hourly rate specified in the association form.

private void InitizalizeAssocData\_ExecuteCode(object sender, EventArgs e)

{

// retrieve the association data from the association form

OnWorkflowActivated activity =

this.Activities["onWorkflowActivated1"] as OnWorkflowActivated;

string associationData = activity.WorkflowProperties.AssociationData;

// retrieve the hourly rate from the association data

double.TryParse(associationData, out amountToPay);

// retrieve the list item

SPListItem item = workflowProperties.Item;

// get the hours on the timesheet

int hours = 0;

int.TryParse(item["Hours"].ToString(), out hours);

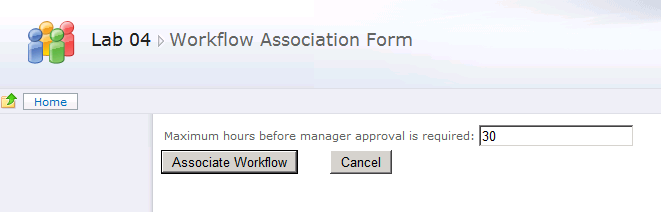
// calculate and save the amount to pay

item["Payment Amount"] = amountToPay \* hours;

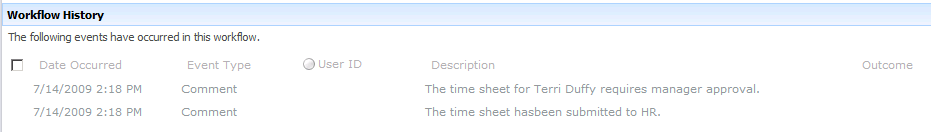
item.Update();

}

1. Deploy the workflow to SharePoint by right clicking the project in solution explorer and clicking **Deploy**.
2. Use the new association page to fill out an hourly rate.
3. Open the browser and navigate to **http://intranet.wingtip.com/sites/Workflow/Lists/TimeSheets**.
4. Using the ribbon select **Library » Library Settings**.
5. Click the **Workflow Settings** link in the **Permissions and Management** section.
6. Click the **Timesheet Submission** link to manage the workflow association.
7. Click **Next** to view the custom association page.
8. Enter a value of **30** and click **Association Workflow**.



1. Navigate back to **http://intranet.wingtip.com/sites/Workflow/Lists/TimeSheets** and start a new workflow.
2. Use the drop down menu on a timesheet and the **Edit Properties** link to change hours to **35**.
3. Click the drop down menu on one of the documents in the list and select **Workflows**.
4. Click the **Timesheet Submission** link to start the workflow.
5. When the workflow is complete click the **Completed** link.
6. Verify the workflow history shows the timesheet required approval.



In this exercise you added an association form to the Visual Studio 2010 workflow you created in the previous exercise and tested the new settings.