Lab 12: Working with InfoPath Forms

**Lab Time**: 45 Minutes

**Lab Directory**: C:/Student/Labs/12\_InfoPath

**Lab Overview**: In this lab, you will learn how to create and deploy an InfoPath Form. In addition, you will gain experience with the new managed InfoPath object model—embedding InfoPath within your own applications. The first two exercises are based on WSS technologies and the last two are based on what MOSS brings to the table.

* In Exercise 1, you are responsible for creating an InfoPath template that will be used by the Litware consultants to fill in their daily timesheets.
  + NOTE: it is our recommendation that you skip Exercise 1 for now and start with Exercise 2, as you have limited time to complete this lab in class and this is not an InfoPath design class. We have included the instructions for Exercise 1 so that you may explore this over lunch or at a later point in time.
* In Exercise 2, you will configure data connections back to SharePoint to facilitate efficiently filling in these forms.
* In Exercise 3, you will deploy the template so that the consultants will use the InfoPath smart client application to fill in the timesheet data.
* In Exercise 4, you will transform your template so that it becomes a content type deployable via MOSS Forms Services.

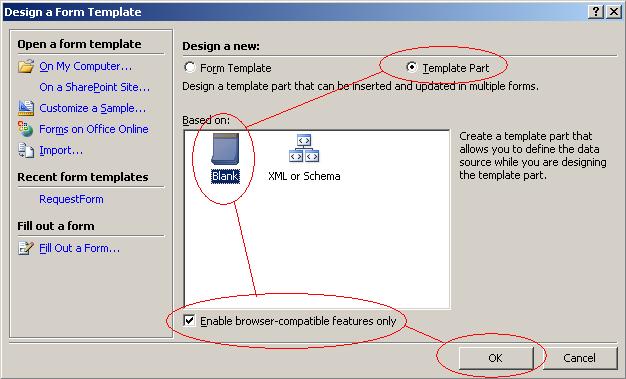
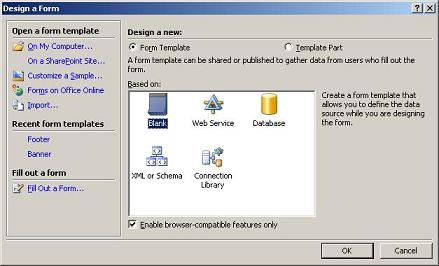
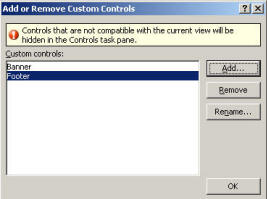
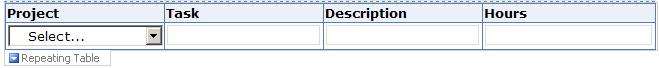
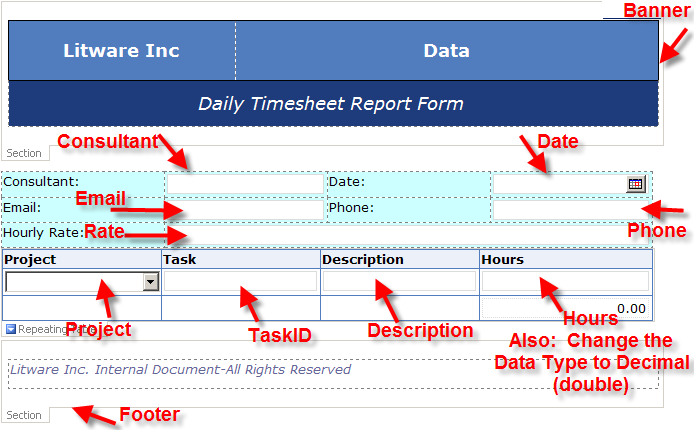
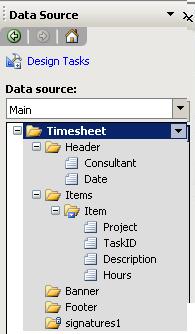
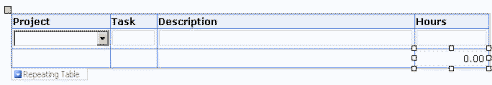
**Important Note**: This lab MUST BE COMPLETED as written, through Exercise 4, for Lab 12 (on workflow) to function correctly.

**Lab Setup—2 pieces**:

1. Lab 08 on Custom Content types must be completed before you can start this lab.
2. This will alter Active Directory to set the Department to Development and also set the manager field for certain users. This 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.
   1. Note: if you want to review the source code that injects information into the user profiles via the object model this is located in the **…\12\_Infopath\Starter Files\Data\ProfileMod** folder.
   2. Run the .bat file located in the InfoPath lab directory \**Starter Files\Mod11Starter.bat**. When finished type exit and press enter to close the cmd window
   3. Navigate to **SharePoint 3.0 Central Administration** (Start - All Programs - Microsoft Office Server - SharePoint 3.0 Central Administration)
   4. In Central Administration on your **Quick Launch** bar (left side of screen) click on **Litware SSP**
   5. In the **Litware SSP User Profiles and My Sites** section click on **User profiles and properties**
   6. 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).

# Exercise 1: Create an InfoPath form to capture timesheet info

NOTE: it is our recommendation that you skip Exercise One for now and start with Exercise 2, as you have limited time to complete this lab in class and this is not an InfoPath design class. We have included the instructions for Exercise 1 so that you may explore this over lunch or at a later point in time.

1. As a starter, you are going to create two InfoPath template parts - one for the Litware banner and one for a footer containing the text Litware uses on all of its internal documents.
   1. Open InfoPath 2007 and in the Getting Started dialog, choose the Design a Form Template link.
   2. In the Design a Form dialog, select the option to create a Template Part based on the Blank template and also check the Web browser enabled check box since you will later examine issues with Forms Server deployment. Click OK to close the dialog. 
   3. In the Design Tasks pane, select Layout and double-click the Two-Column Table option.
   4. In the first cell, add Litware Inc as text and in the second one Our code is so tasty, it will melt in your mouth. Center the Text in both cells. Apply some border and shading to the table.
   5. Add a second row to the table and merge the two cells. Type **Add here the title of the template** in the merged cell. \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\Header.gif
   6. Save the template part in your **/Lab/Template Parts** folder as Banner.xtp and close InfoPath.
   7. Open InfoPath and follow the steps A and B again to create the second template part.
   8. In the Design Tasks pane, select Layout and double-click the One-Column Table option.
   9. Add the text **Litware Inc. Internal Document - All Rights Reserved**. Use the font and italic option to change the look and feel. \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\footer.JPG
   10. Save the template part in your /**Lab/Template Parts** folder as Footer.xtp and close InfoPath.
2. Now that you have your two re-usable InfoPath blocks, start to create the main InfoPath template to capture the timesheet information. In the first instance, you just have to take care of the layout and the controls.
   1. Open InfoPath 2007 and in the Getting Started dialog, choose for Design a Form Template.
   2. In the **Design a Form** dialog, select the option to create a **Form Template** based on the **Blank** template and also check the Web browser enabled check box since you will later deploy for a Forms Server site. Click OK to close the dialog. 
   3. In the **Design Tasks** pane, click on Controls. Review the controls you can use.
      1. Q: Why are there so few controls?
      2. A: Because we are mandating that this form be usable both in InfoPath 2007 and Forms Services via the web, our control choices are limited to those that are usable both places.
   4. Click on the Some controls are not compatible with the current form and have been hidden box at the bottom of the pane. Review the message box.
   5. Click on the Add or Remove Custom Controls link at the bottom of the pane.
   6. Add the two template parts you have created in step 1 and close this dialog. 
   7. Drag and drop the banner on the blank view
   8. Press Enter 2 times to add a two blank lines
   9. Drag and drop the footer.
   10. Add **Daily Timesheet Report Form** as the text in the banner for the second row. 
   11. Place your insertion point (the flashing vertical line on the screen) in the first blank line (i.e. using the mouse click on the first blank line.)
   12. Go back to the **Design Tasks** pane and select Layout.
       1. Hint: on the right hand side of your InfoPath design window there is the Controls list. At the top of this list is a link that says "Design Tasks" click on this to return to the main Design Tasks pane and then select the Layout choice from this main screen.
   13. Add a **Custom Table** with 4 columns and 3 rows between the banner and footer.
       1. Add **Consultant**: in the first cell of row 1
       2. Add **Email**: in the first cell of row 2
       3. Add **Date**: in the third cell of row 1
       4. Add **Phone**: in the third cell of row 2
       5. Add **Hourly Rate**: in the first cell of row 3
   14. **Merge** \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\MergeCellsTBB.JPG the other three cells of row 3 into 1 cell.
   15. Go back to the **Design Tasks** pane and select **Controls**.
       1. Hint: on the right hand side of your InfoPath design window there is the **Controls** list. At the top of this list is a link that says "**Design Tasks**" click on this to return to the main **Design Tasks** pane and then select the **Controls** choice from this main screen.
   16. Add the following controls to the view:
       1. A **Text Box** in the second cell of row 1
       2. A **Date Picker** in the fourth cell of row 1
       3. We will leave the second cell and fourth cell of row 2, and the second cell of row 3 empty for the moment.
   17. Select the table and use the border and shading \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\bordersAndShadingTBB.JPG to create a nice looking table. 
   18. Place your insertion point (the flashing vertical line on the screen) in the second blank line (i.e. using the mouse click on the blank line immediately beneath the table.)
   19. Add a **Repeating Table** with **4 columns** after the table you just created.
       1. Hint: from your **Insert Menu** select **Repeating Table**...
   20. Add **Project**, **Task**, **Description** and **Hours** as headers. 
   21. Right-click the **Text Box** under **Project** and **Change To: Drop-Down List Box**.
   22. Click the **Preview** button in the toolbar to check the result of your work.
   23. Close the preview.
   24. Save the template as **TimeSheet.xsn** under your **/Lab** folder.
   25. You have started really from scratch. While you were busy in these previous steps adding the controls to the view, InfoPath generated an XSD schema for you. The only thing to do now is make changes to the default names and types.
3. Go back to the Design Tasks pane and select Data Source.
   1. Double-click each the elements in the data source and make the following changes:
      1. rename **myFields** to **timesheet**
      2. rename **group1** to **items**
      3. rename **group2** to **item**
      4. Using the image below rename the remaining items as follows: 
   2. Next add a group under the **Timesheet** element called Header (use the **Move Up** to position the new group)
      1. Hint: on the right side of the screen using the Data Source panel, expand the drop down off of timesheet and click Add...
   3. Move the **Consultant and Date** field to the Header group.
   4. Move the Banner and the Footer all the way to the bottom. Hint:
      1. In the **Data Source Panel** select **Banner** and then select **Move Down** off the drop down menu.
      2. Repeat for the **Footer**.
      3. Select **Header** and then select **Move Up** off the drop down menu.
      4. select the Date field and select Move... off the drop down menu select the Header folder and
      5. Then select **OK**.
      6. Your **Data Source Main** should look as follows: 
   5. Right-click the table and open the **Repeating Table Properties**. On the **Display** tab check the **Include Footer** checkbox. Click OK.
   6. Add an **Expression Box** in the Hours column in the footer row. Use the formula editor to insert a Sum function for the Hours field. Change the type of the expression box content to Decimal. Hint:
      1. Select **Design Tasks** in your **Data Source** panel and then select **Controls**
      2. Place your insertion point in the last cell in the last row of your repeating table (i.e. 4th column, row 3)
      3. Select **Expression Box** from the Controls Panel
      4. Select the Function button \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\FxButton.JPG
      5. Select Insert Function...
      6. Select sum and click **OK**
      7. double click to insert field and pick the Items - Item - Hours field
      8. Click **OK**
      9. In the Result section Format as: drop down box select Decimal
      10. Click **OK**
   7. Be sure to format both the **Hours** Repeating field and the **Sum of Hours** field so that they have a Right Alignment and always display 2 decimal places. 
   8. Save the template.

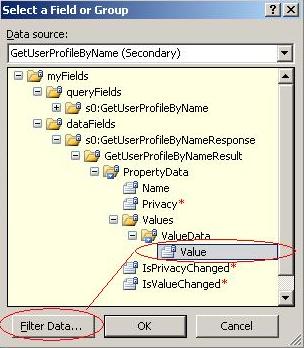
# Exercise 2: Pre-populate the InfoPath form with SharePoint Data

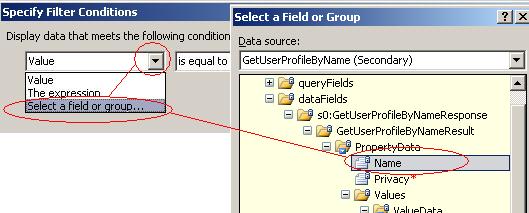
In this exercise we will be setting up security on our InfoPath form to allow data connections with the SharePoint Server and we will set this form up to pull data from the SharePoint Profile Database and the Projects list that exists on our Litware Project Management Site.

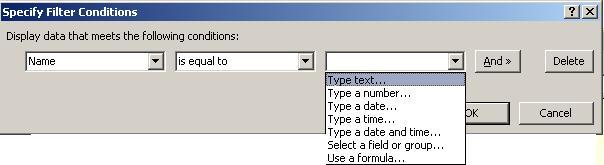
**Exercise Setup**:

* If you did not complete Exercise 1 then you must first copy the starter files located at ...**12\_InfoPath\Starter Files\exercise2\** to your **...\12\_InfoPath\Lab\** directory.
* You will then need to open the **TimeSheet.xsn** in **Design mode** by right clicking on it and selecting **Design**.

**Exercise**:

1. First you'll need to create a Data connection to pull user profile information into our form.
   1. Add a connection to receive data by using the **Tools > Data Connections...** menu item.
   2. Click the **Add...** button.
   3. Select **Create a new connection to:** and then **Receive data** as the source type.
   4. Click the **Next >** button.
   5. Select **Web Service** on the data source screen.
   6. Click the **Next >** button.
   7. When asked for the location of the web serice type **http://litwareinc.com/\_vti\_bin/UserProfileService.asmx**
   8. Click the **Next >** button.
   9. In the **Select an operation:** list select **GetUserProfileByName**
      1. *Note: this method will retrieve a* ***PropertyData*** *array (i.e. like a repeating table of key/value pairs)*
   10. Click the **Next >** button.
   11. Click **Next >** again
       1. *Note: we could pass a parameter here to select a certain user account. However, we are using the default functionality of this method, which is to find the current users account if no parameters are passed.*
   12. Click **Next >** a third time
       1. *Note: as we do not wish to store an offline copy of this data up front there is nothing to do here.*
   13. Click the **Finish** button then click **Close** to return back to your InfoPath Form.
       1. *Note: Automatically retrieve data when form is opened is selected by default.*
2. Now that we have the **PropertyData** array we will populate several fields on our form with this data
   1. Double click on the **TextBox** in the cell next to **Consultant**: (located in row 1 column 2 of the first table)
      1. *Note: when you repeat this step later you will be using an expression field instead of a text box and using the table from step H. to identify the field location next to which you will add the expression field. DO NOT FOLLOW THIS NOTE THE FIRST TIME YOU RUN THIS... (i.e. for the Consultant choice)*
   2. click the **Fx** button \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\FxButton.JPG
   3. On the **Insert Formula** dialog box click **Insert Field or Group**...
   4. In the **Data Source:** dropdown box select the secondary data source **GetUserProfileByName**
   5. Drill down through the fields until you select the **Value** Field (see Graphic below) and then click Filter Data...
   6. On the **Filter Data** dialog box click the **Add...** button.
   7. Then fill this form in using the graphics below. **Note**: be sure to change the first drop down box from **Value** to **Name** by selecting the "**Select a field or group...**" and then using the graphics below to complete the task.



* 1. Next set the third drop down to **Type text...** 
  2. After Selecting **Type text...** type "**AccountName**" and Click **OK** four to five times to close all dialogs.
     + 1. *Note: when you repeat this step later you will be replacing AccountName with the value found in the* ***Type Text*** *column from the table found in step j. but DO NOT DO THIS THE FIRST TIME THROUGH...*
  3. Now you will add an expression field next to **Email**: (i.e. Row 2 column 2)
     1. Put your insertion point into row 2 column 2 of the first table
     2. using your **Insert** menu select **More controls...** and then select **Expression** box from the **Advanced** section.
  4. Repeat the steps (a—h), for the three Expression fields in the table below.
     1. **Note**: After completing these steps for **Email** you will once again need to insert an expression field next to **Phone**, and then do steps a—h again. Finally you will need to insert an expression field next to **Hourly Rate**: and run steps a—h again.
     2. **Important**: As InfoPath 2007 is very temperamental when it comes to Forms Services (i.e. browser enabled forms) these steps MUST BE COMPLETED in the exact order they are written (i.e. add 1 expression field and then immediately configure it before moving to the next field)
     3. **Further Note**: even the act of editing an existing Expression field will cause validation errors in browser compatibility. The easiest way to resolve these is to delete and re-create the Expression field and the expression it contains.

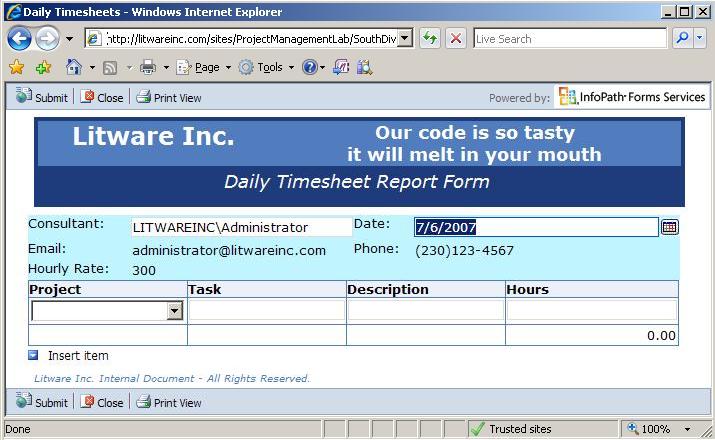
|  |  |
| --- | --- |
| **Field** | **Type Text** |
| Email | WorkEmail |
| Phone | WorkPhone |
| Hourly Rate | HourlyRate |

* 1. Double click on the **Consultant** **TextBox** and from the **Display** tab select **Read-only** to ensure that users cannot alter this important information.
  2. Save the changes to the form template, and click **Overwrite** if the warning message comes up.
     1. *Note: You may receive a warning regarding the fact that when you change data sources on an existing form template existing forms will also be converted to this new template starting point possibly causing a permanent loss of data. This warning does not apply to us as we have yet to publish this form.*
  3. Finally you'll need to create a data connection for the list of projects as well. This time you'll be using a SharePoint list as the starting point (this list should already exist on our **Project Management Site** **http://litwareinc.com/sites/ProjectManagementLab/Lists/Projects**).

1. First add a connection to receive data of type **SharePoint library or list**.
   1. Add a connection to receive data by using the **Tools > Data Connections...** menu item.
   2. Click the **Add...** button.
   3. Select **Create a new connection to:** and then **Receive data** as the source type.
   4. Click the **Next >** button.
   5. Select **SharePoint Library or List** and click **Next>**
   6. Use a path of **http://litwareinc.com/sites/ProjectManagementLab/** and click **Next**.
   7. Select the **Projects** list and then click **Next** to advance to the column selection page.
   8. Uncheck all of the columns except **Project** and click **Next** twice then **Finish** to create the new connection.
   9. Click **Close** to close the **Data Connections** window.
2. Now you will connect the **Project** drop down on the page to the data connection created in the previous step.
   1. Right click the **Project** drop down and select **Drop-Down List Box Properties**.
   2. In the **List box entries** section, select **Look up values** **from an external data source** and select the **Projects** data source.
   3. Next click the button to the right of the **Entries** text box \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\EntriesSelectButton.JPG and select the **:Project** node.
   4. Click **OK (two times)** to assign the connection.
3. Now that the data is populated, you'll need to perform some validation checks on the data entered into the form.
   1. Set today's date as the default date
   2. Double click on the **Date** field (to the right of the **Date:** title field).
      1. Click the **Fx** button. \\litwareserver01\d$\Data\Courseware\SP_Html2Word\SPT401\Student\Labs\07_InfoPath\figs\FxButton.JPG
      2. Click on **Insert Function**... button
      3. select the **Today** function from the **Most Recently Used** Category
      4. Click **OK** three times
   3. Create a data validation rule for the date field. Restrict users from entering a date in the future.
      1. Double click on the **Date** field (to the right of the Date: title field).
      2. Click on **Data Validation**... then click on **Add**...
      3. Set the condition **Date is greater than** and then from the third drop down box select **Use a formula**...
      4. Type **today()** for the formula and click **OK**
      5. Set the **Screen Tip:** to **You cannot select a date in the future**
      6. Set the **Message:** to **You cannot record work for a future date... All work recorded must be for Today or earlier dates**.
      7. Click **OK (three times)**
   4. Change the background color of the expression box that calculates the total of hours to red if the amount is bigger than 8
      1. Right Click on the **Total Hours Expression Box** (i.e. the one with **0.00** in it) and select **Expression Box Properties**...
      2. On the **General** tab select the expression in the **XPath**: box and copy it (i.e. use **Ctrl+C**)
      3. Select the **Display** tab then click on **Conditional Formatting**... then click **Add**...
      4. Set the condition to **The expression**
      5. Set the condition Text box to **sum(my:Items/my:Item/my:Hours) > 8**
         1. Hint: Put your insertion point into this text box and press **Ctrl+v** to paste your formula in and then add **> 8**
      6. if this condition is true set the **Font color**: to a shade of **Red**
      7. Click **OK (Three times)**
4. **Save** the template as **TimeSheet.xsn** in the Lab folder. Test your form by clicking the **Preview** button.
   1. Note: You will be Informed about a potential security concern, click **Yes**.
   2. If you are also questioned about working offline select the **Try to connect** button in response to this.
   3. Take your time to inspect the form. Close the **Preview**.
5. The last part of this exercise is to prepare to save the data in the InfoPath form into a SharePoint Form Library. To do this, you'll need to create a new Form Library named **Daily Timesheets** in the **http://Litwareinc.com/sites/ProjectManagementLab/SouthDivision** site.
   1. Navigate to the **South Division** site using the **South Division** tab at the top of the **Litware Project Management** site
   2. Select **Site Actions - Create - Form Library** (from the **Libraries** section)
   3. Name the new Library **Daily Timesheets**
   4. Click **Create**
6. As a final step you add a button control to the view that will be used to submit all of the data to SharePoint document library.
   1. Return to your InfoPath form in Design mode.
   2. Add a new **Data Connection** using the **Tools** and then **Data Connections** in the InfoPath menu. Click **Add**.
   3. This one will be a connection to **Submit data** (Create a new connection to **Submit data**).
   4. Click **Next.**
   5. Select **To a document library on a SharePoint site.**
   6. Click **Next.**
   7. Enter the URL to the document library **http://litwareinc.com/sites/ProjectManagementLab/SouthDivision/Daily%20Timesheets/**
   8. Click the **fx** button to the right of the filename to reset the name of the saved form.
   9. Set the name of the form to **concat(consultant, "\_", date)**. You'll need to build this string, don't just paste into the text box. Do this by double clicking each section of the formula and choosing “**Consultant**” and “**Date**” from the **Header** group. For the center section you must type **“\_”**.
   10. Click **OK** to close the **Insert Formula** dialog.
   11. Check the **Allow overwrite if file exists.**
   12. Click **Next** and **Finish** to close the **Data Connection Wizard**.
   13. Click **Close** to close the **Data Connections** dialog.
7. Drop a **Button** control on the Form between the **Repeating Table** and the **Footer**.
8. Double-click the control and change the **Label** and **ID** to **Submit**.
9. Use the **Rules** to add a new rule with no condition but with the following three actions:
   1. **Submit using a Data Connection** - the SharePoint one you just created.
   2. **Show a Dialog Box message** with a notice that "**Your timesheet was received successfully.**"
      1. Note: this action is not supported by InfoPath Forms Services...
   3. **Close** this form: No Prompt
10. Click **OK (three times)** to return back to your form design window.
11. Time to test out your work.
    1. **Save** the template and close InfoPath.
    2. Double-click the **TimeSheet.xsn** file in your **/Lab** folder.
    3. Click **Yes** in response to the security concern.
    4. Fill in some data and then click the **Submit** button.
    5. If you get the message box that you entered information for, then your information was processed with success by the Web service.
    6. You can verify if the information has been added to the database by using the **TimeSheetReport** application you worked with in the lab on template-based solutions.

# Exercise 3: Publish form to WSS forms library

In this exercise, you are playing the role of the administrator who is responsible for the deployment of the InfoPath template. You will be deploying the template to a SharePoint forms library and make it available via the Forms Server for those consultants that do not have InfoPath installed on their laptops (i.e. you will begin utilizing MOSS Forms Services)

1. Open the **TimeSheet.xsn** file you created in the previous exercise in design mode.
2. Since you'll be using **Forms Services** to render this form, you'll need to make some changes to the submission process.
   1. First remove the button from the form.
   2. Configure the submit options using **Tools -> Submit Options**.
   3. Check **Allow users to submit this form**
   4. Select **Send form data to a single destination** with a destination of **SharePoint document library**.
   5. Select **SharePoint Library Submit** as the data connection to use.
   6. Click **Advanced >>** and select **Close** the form in the **After submit** drop down list.
   7. Click **OK** to apply the changes.
3. Next you'll need to disable the save option on the form to force the user to submit, not save.
   1. Open **Tools -> Form Options.**
   2. Select the **Open and Save** pane.
   3. In this pane uncheck **Save and Save As** to disable saving.
   4. Click **OK.**
4. In the **Design Tasks** pane (right side of InfoPath screen) click on the **Design Checker**.
   1. You should not see any red icons indicating a compatibility error.
   2. You might see one or more blue icons indicating a warning.
5. Go back to the **Design Tasks** pane and click the **Publish Form Template** link. This wizard will guide you through all the needed steps to make your template available via SharePoint.
   1. Click **OK** to the modification message.
   2. Select **To a SharePoint server with or without InfoPath Forms Services.**
   3. Click **Next.**
   4. Give the URL to the SharePoint server (**http://litwareinc.com/sites/ProjectManagementLab/SouthDivision/**)
   5. Click **Next**
      1. **Note**: if you have a problem publishing your site you may have to run "**net stop sens**" from a cmd prompt. This is only an issue on a single machine development environment and only affects Office 2007 items interacting with SharePoint.
   6. You will note that you cannot currently enable this form to be filled out using a browser this is controlled via a MOSS site collection feature.
      1. Keep your InfoPath Design window open and in a browser navigate to **http://litwareinc.com/sites/ProjectManagementLab/**
      2. Select **Site Actions > Site Settings > Site collection Features** (from the **Site Collection Administration** column)
      3. Locate and Activate the **Office SharePoint Server Enterprise Site Collection** features choice (among other things this controls forms services)
      4. Return to your InfoPath Design window and click **Back** and then click **Next** you should see that the browser based option is now available.
         1. **Note**: If you now see an additional warning that our form requires administrator approval before it can be used via a browser it is likely for one of two reasons:
            1. you have browser incompatible items on your form (in your Design Tasks Pane switch to Design Checker and look for Errors (i.e. items with a red X)
            2. you have somehow enabled a VSTO project for your InfoPath form (i.e. created a code behind page) To check this, in InfoPath select Tools - Form Options - select Programming (from Category:) - look for the Remove Code button (if this is enabled you've got code behind problems) to remove them click Remove Code and then click OK
            3. After fixing the issues try Steps 5 a through f again, then move on to step g.
   7. Check **Enable this form to be filled out by using a browser** and select **Document Library** click **Next**.
   8. Select **Update the form template in an existing document library** and choose the **Daily Timesheets** library. Click **Next >.**
   9. Add **Consultant** and **Date** as new columns to be created.
   10. Click **Next** and **Publish.**
   11. In the complete dialog, click **Open this form template in the browser** to view the form in the browser.
   12. Once the publish process is completed, close InfoPath. 
6. Test the template by clicking **New** in the **Daily Timesheets** forms library. Notice that your form opens in InfoPath and not the browser. By default forms will use the browser only if InfoPath isn't installed. Fill in the form and submit your changes using the **Submit** button.
7. As an administrator of the forms library, you can specify that the InfoPath form always has to be filled in using the browser, even if the consultant has InfoPath installed on his machine.
   1. In the forms library, select **Settings** and then **Form Library Settings**.
   2. Click on **Advanced Settings.**
   3. Activate the **Display as a Web page** option in the **Browser-enabled Documents** section
   4. Click **OK** and using the breadcrumbs navigate back to the **Daily Timesheets** Forms Library
8. Test the template by clicking **New** in the forms library. The form will be made available in the browser now.

# Exercise 4: Creating an InfoPath Form Content Type for use with Forms Services

In Exercises 1-3 you have created an InfoPath Form Template that works well with the Browser and the InfoPath Client Application. Your next task is to set this Form up so that it might be deployed as a content type to enable re-use with multiple Forms libraries. In order to accomplish this you will need to use managed code to dynamically determine the server and library name where the form was launched and then modify the **FolderUrl** property of our submit data connection to ensure submission to the correct library. You will also need to modify the security settings to allow this content type to be deployed on the server and used with different Form Libraries.

**Important Note**: Up until now we have been able to switch between using InfoPath or the browser to submit our forms. However, once we give our Form the ability to automatically detect the location it was launched from (browser or InfoPath) things begin to change. One limitation is that once we decide to deploy our Form via Forms Services (i.e. Farm Administrator deployed forms), which gives us the ability to re-use these forms in many places and use many more features in the browser-based scenario, certain scenarios are not supported for InfoPath direct usage. One of these scenarios is setting up your form to dynamically determine the location from which it was launched. InfoPath has no way to determine where the form is currently coming from if the template is stored centrally in the SharePoint Forms Services location, therefore we are forced to utilize only the browser based form by the end of this Exercise.

1. If necessary open your **TimeSheet.xsn** in design mode in InfoPath 2007
2. Add a "hidden" string to your Form to hold submission location information (to allow future edits to existing documents).
   1. In your **Design Tasks** pane (right side of InfoPath window) select **Data Source.**
   2. In the **Data Source: Main** select the **Timesheet** drop down and then click **Add**...
   3. Set the Name: to **strPath** and click **OK.**
   4. Repeat step 2 this time setting the Name: to **strExists** also set the **Default value**: to **N**
3. Modify your **Submit Options** to allow for dynamically picking up the Forms starting location.
   1. Navigate to **Tools - Form Options - Select Programming** from the **Category**: list
   2. Set your Form template code language to either **C#** or **Visual** **Basic.**
   3. Set your Project Location to **...\Labs\12\_InfoPath\Lab\MOSS.**
   4. Click **OK.**
   5. Navigate to **Tools - Submit Options.**
   6. Select the option for **Perform custom action using Code** and click the **Edit Code** button. This will open a VSTA programming window.
      1. In the VSTA programming window we need to add code to discern the current location the form is being launched from.
4. Inside the **FormCode** Class file read over the comments, notice that you cannot use Member variables in browser-enabled forms. You will need to use the **FormState** property bag to store location information for our content type. Add the following code inside of your **FormCode Class**.

C#

private object \_libraryUri

{

get { return FormState["\_libraryUri"];}

set { FormState["\_libraryUri"] = value; }

}

VB.NET

Private Property \_libraryUri() As Object

Get

Return FormState("\_libraryUri")

End Get

Set

FormState("\_libraryUri") = value

End Set

End Property

1. Next you need to use the Forms Loading event handler to populate the \_**libraryUri** property. **Note**: The **EventManager.FormEvents.Submit** is already present in the code. You must add the Event wireup to the **InternalStartup()** method first:

C#

public void InternalStartup() {

EventManager.FormEvents.Submit += new SubmitEventHandler(FormEvents\_Submit);

EventManager.FormEvents.Loading += new LoadingEventHandler(FormEvents\_Loading);

}

public void FormEvents\_Loading(object sender, LoadingEventArgs e) {

// We need to Get the Uri (or SaveLocation in a browser form)

// of the library the form was opened from.

// First determine if the form was opened in the browser or not

if (Application.Environment.IsBrowser) {

// If true, test for and use the "SaveLocation" from the InputParameters

if (!String.IsNullOrEmpty(e.InputParameters["SaveLocation"].ToString())) {

\_libraryUri = e.InputParameters["SaveLocation"].ToString();

}

}

else {

// NOTE: Even though we will not be able to utilize this code path in

// our current scenario, you may have a situation where you want

// to just use InfoPath to fill in the form (and not Forms Services)

// So If the form was opened in the client, we will get the Uri

if (!String.IsNullOrEmpty(this.Template.Uri.ToString())) {

\_libraryUri = this.Template.Uri.ToString();

}

}

}

VB.NET

Public Sub InternalStartup()

AddHandler EventManager.FormEvents.Submit, AddressOf FormEvents\_Submit

AddHandler EventManager.FormEvents.Loading, AddressOf FormEvents\_Loading

End Sub 'InternalStartup

Public Sub FormEvents\_Loading(sender As Object, e As LoadingEventArgs)

' We need to Get the Uri (or SaveLocation in a browser form)

' of the library the form was opened from.

' First determine if the form was opened in the browser or not

If Application.Environment.IsBrowser Then

' If true, test for and use the "SaveLocation" from the InputParameters

If Not [String].IsNullOrEmpty(e.InputParameters("SaveLocation").ToString()) Then

\_libraryUri = e.InputParameters("SaveLocation").ToString()

End If

Else

'NOTE: Even though we will not be able to utilize this code path in

' our current scenario, you may have a situation where you want

' to just use InfoPath to fill in the form (and not Forms Services)

' So If the form was opened in the client, we will get the Uri

If Not [String].IsNullOrEmpty(Me.Template.Uri.ToString()) Then

\_libraryUri = Me.Template.Uri.ToString()

End If

End If

End Sub 'FormEvents\_Loading

1. Now you will modify the **FormEvents\_Submit** method to utilize the \_**libraryUri** to define the location for forms submission.

C#

public void FormEvents\_Submit(object sender, SubmitEventArgs e) {

//Create a temporary storage location for the final path

string strPath = "";

//Create a Navigator object for the main DOM

XPathNavigator xnDoc = this.MainDataSource.CreateNavigator();

//Create Navigator objects for each field

XPathNavigator xnPath =

xnDoc.SelectSingleNode("my:Timesheet/my:strPath", this.NamespaceManager);

XPathNavigator xnExists =

xnDoc.SelectSingleNode("my:Timesheet/my:strExists", this.NamespaceManager);

// First we must determine if the form is being edited or if it is new

if (xnExists.Value.ToString()=="Y") {

strPath = xnPath.Value.ToString();

}

else {

// New form

// Create a temporary storage location for Uri

string strUri = \_libraryUri.ToString();

if (Application.Environment.IsBrowser) {

// because the form is browser based, the libraryUri value is just

// the server name and library - so we just need to get

// the URL without the last "/"

strPath = strUri.Substring(0, strUri.LastIndexOf("/"));

}

else {

// because we opened the form in InfoPath

// Parse just the path to the document library

// The URL up to /Forms...

strPath = strUri.Substring(0, strUri.IndexOf("Forms") - 1);

}

xnExists.SetValue("Y");

xnPath.SetValue(strPath);

}

// If the submit operation is successful, set

// e.CancelableArgs.Cancel = false;

// Write your code here.

// Get a reference to the submit data connection

FileSubmitConnection fc =

(FileSubmitConnection)this.DataConnections["SharePoint Library Submit"];

// Modify the URL we want to submit to using strPath

fc.FolderUrl = strPath;

// Execute the submit connection

try {

fc.Execute();

e.CancelableArgs.Cancel = false;

}

catch {

e.CancelableArgs.Cancel = true;

}

}

VB.NET

Public Sub FormEvents\_Submit(sender As Object, e As SubmitEventArgs)

' Create a temporary storage location for the final path

Dim strPath As String = ""

' Create a Navigator object for the main DOM

Dim xnDoc As XPathNavigator = Me.MainDataSource.CreateNavigator()

' Create Navigator objects for each field

Dim xnPath As XPathNavigator = \_

xnDoc.SelectSingleNode("my:Timesheet/my:strPath", Me.NamespaceManager)

Dim xnExists As XPathNavigator = \_

xnDoc.SelectSingleNode("my:Timesheet/my:strExists", Me.NamespaceManager)

' First we must determine if the form is being edited or if it is new

If xnExists.Value.ToString() = "Y" Then

strPath = xnPath.Value.ToString()

Else 'new form

'Create a temporary storage location for Uri

Dim strUri As String = \_libraryUri.ToString()

If Application.Environment.IsBrowser Then

' Because the form is browser based, the libraryUri value is just

' the server name and library - so we just need to get

' the URL without the last "/"

strPath = strUri.Substring(0, strUri.LastIndexOf("/"))

Else

' Because we opened the form in InfoPath

' Parse just the path to the document library

' The URL up to /Forms...

strPath = strUri.Substring(0, strUri.IndexOf("Forms") - 1)

End If

xnExists.SetValue("Y")

xnPath.SetValue(strPath)

End If

' If the submit operation is successful, set

' e.CancelableArgs.Cancel = false;

' Write your code here.

' Get a reference to the submit data connection

Dim fc As FileSubmitConnection = \_

CType(Me.DataConnections("SharePoint Library Submit"), FileSubmitConnection)

' Modify the URL we want to submit to using strPath

fc.FolderUrl = strPath

' Execute the submit connection

Try

fc.Execute()

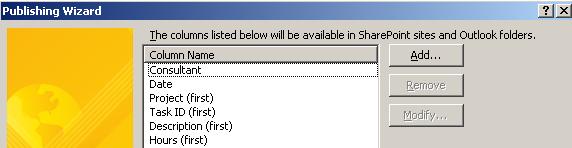
e.CancelableArgs.Cancel = False

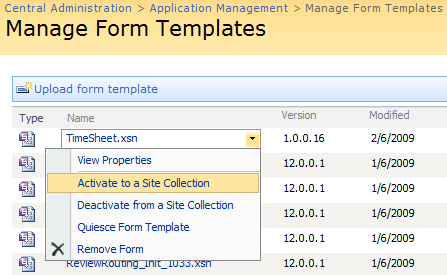
Catch

e.CancelableArgs.Cancel = True

End Try

End Sub 'FormEvents\_Submit

1. Build your Timesheet VSTA application
2. Close Visual Studio if the build was successful.
3. Click the **OK** button to close the **Submit Options** dialog.
4. Now you need to prepare the content type for deployment. Because this contains back end code this type must be administrator approved and signed with a certificate. **Note**: We will use a self-signed certificate for testing purposes. For the real world you would need to use a certificate granted from a trusted authority like Verisign.
   1. First you need to set the security level for this form and specify a certificate for the form to use.
      1. On the InfoPath Menu Bar select **Tools > Form Options**
      2. In the **Category:** list Select **Security and Trust**
      3. Remove the check from **Automatically** **determine security level (recommended)**
      4. Select **Full Trust**
         1. **Note**: this will ensure that we are not questioned about security. We are in essence stating that we have verified this form will do no harm once deployed. Forms with Full Trust MUST BE approved by an administrator prior to use.
      5. Select the **Sign this form template** check box (under Form Template Signatures)
      6. Click the **Create Certificate...** button and click **OK** to the message about Self-Signed certificates.
      7. Click **OK**.
   2. In InfoPath save the form
5. Now you will attempt to publish this form template as before.
   1. Go back to the **Design Tasks** pane and click the **Publish Form Template** link. This wizard will guide you through all the needed steps to make your template available via SharePoint.
   2. If you see a message warning you about modifying an existing Form, Click OK to the modification message.
   3. Select **To a SharePoint server with or without InfoPath Forms Services.**
   4. Click **Next.**
   5. Give the URL to the SharePoint server (**http://litwareinc.com/sites/ProjectManagementLab/SouthDivision**/)
   6. Click **Next**.
   7. Check **Enable this form to be filled out by using a browser** and notice that you now only have one choice available.
   8. Select **Administrator-approved form template (advanced).**
   9. Click **Next**.
   10. Create/Browse to the C:\ location and create a folder named **Forms** (you must create this folder before attempting to use it).
   11. Specify a filename of **TimeSheet.xsn**.
   12. Click **Save** (verify that your path is **C:\Forms\TimeSheet.xsn**).
   13. Click **Next**.
   14. Add the following columns (see figure) to those already there (i.e. **Consultant** and **Date**) as new columns to be created. 
   15. Click **Next** and **Publish**. **Note**: Take note of the published location. This is the location we will need later to upload the form. In a production environment this location is likely to be a network file share that a SharePoint Farm Administrator can easily navigate to.
   16. click **Close**.
   17. Once the publish process is completed, close InfoPath.
6. Next we will verify that the Form template is ready for uploading by using **SharePoint Central Administration.**
   1. Click **Start > All Programs > Microsoft Office Server > SharePoint 3.0 Central Administration**.
   2. In the top navigation bar, click the **Application Management** tab.
   3. On the **Application Management** page, under **InfoPath Forms Services**, click **Upload** form template.
   4. On the **Upload Form Template** page, click **Browse....**
   5. In the **Choose file** window, browse to **C:\Forms\TimeSheet.xsn**, click the template, and then click **Open**.
   6. Click **Verify**.
   7. In the **Report Details** section, look for any errors and warnings for the form template.
      1. **Note**: If the system warns you that the template already exists, click **Application Management**, click **Manage form templates**, click the arrow that appears next to the form template, and then click **Remove Form**. On the **Remove Form Template** page, click **Remove**. Try step 11 again.
   8. If you did not receive a warning, click **OK**.
7. After you have verified that there are no problems with your Form, you will upload the form template by using **SharePoint Central Administration.**
   1. You should still be on the **Upload Form Template** page, click **Browse....**
   2. In the **Choose file** window, browse to **C:\Forms\TimeSheet.xsn**, click the template, and then click **Open**.
   3. Click **Upload**.
      1. **Note**: The form template is uploaded to the server. Although uploaded, it is not yet available to users. It must be activated by a farm administrator who has site collection administration permissions. You will do this next.
   4. Click **OK**.
8. To make the form template available to users, the form must be activated to a site collection.
   1. You should be on the **Manage Form Templates** page.
   2. 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**.



* 1. On the **Activate Form Template** page, click the **Site Collection URL**, then click **Change Site Collection**.
  2. Using the **Select Site Collection** page, click the **/sites/ProjectManagementLab** site, and then click **OK**. If this worked, skip ahead to Step e.
     1. **Note**: If your site collection is not listed ensure that the web application is set to http://litwareinc.com.
        1. To fix this, click the Web Application URL, and then click Change Web Application.
        2. Click the http://litwareinc.com Web application on the Select Web Application page.
     2. Try step 8 again.
  3. Verify that the information looks correct on the **Activate Form Template** page, and then click **OK**.

1. Now we need to verify that the form template is available on the site collection.
   1. In **Internet Explorer**, browse to **http://litwareinc.com/sites/ProjectManagementLab.**
   2. Click **View All Site Content**.
   3. On the **All Site Content** page, in the **Document Libraries** section, click **Form Templates**. The **TimeSheet** template should be in the **Form Templates** list.
2. Finally we can now utilize the form template as a template for a library
   1. From the **Litware Project Management** site browse to your **South Division** - **Daily Timesheets Library**
   2. Select and delete each existing test form in this library to make way for the new template.
   3. On the top navigation bar, click **Settings**, and then click **Form Library Settings**.
   4. On the **Customize Daily Timesheets** page, click **Advanced settings**.
   5. On the **Form Library Advanced Settings: Daily Timesheets** page, under **Allow management of content types**, select **Yes**, and then click **OK**.
   6. On the **Customize Daily Timesheets** page, in the **Content Types** section, click **Add from existing site content types**.
   7. On the **Add Content Types: Daily Timesheets** page, in the **Available Site Content Types** list, select the **TimeSheet** form template to use as the template for this library, click **Add**, and then click **OK**.
   8. On the **Customize Daily Timesheets** page, in the **Content Types** section, click on the **Form** content type.
   9. On the **List Content Type: Form** page, select **Delete this content type** and select **OK** to the confirmation message.
3. Now you can test out this new form by entering some data. Note: If you edit an existing form you will receive a warning message stating that "**There has been an error while processing the form**." If you Show error details your are informed that "The given key was not present in the dictionary." This error is related to internal data on your InfoPath form and can be ignored with no ill effects, Click Continue.