



ENABLING NEXT GENERATION AGILE, ADAPTIVE AND PROCESS-MANAGED ENTERPRISE

Getting Started Tutorial For ASP.NET Developers

Table of Contents

Disclaimer	3
Introduction	4
Create ASP.NET Web Site	5
Advanced Scenarios	34
Summary	65

Disclaimer

AgilePoint Inc. makes no representations or warranties, either expressed or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Copyright © 2013, AgilePoint Inc.

All Rights Reserved

GOVERNMENT RIGHTS LEGEND: Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable AgilePoint Inc. license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

"AgilePoint Inc." and AgilePoint Inc.'s products are trademarks of . References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

This document is confidential and the property of AgilePoint Inc. Permission is required to re-distribute, copy, or use any of the text or image files.

Introduction

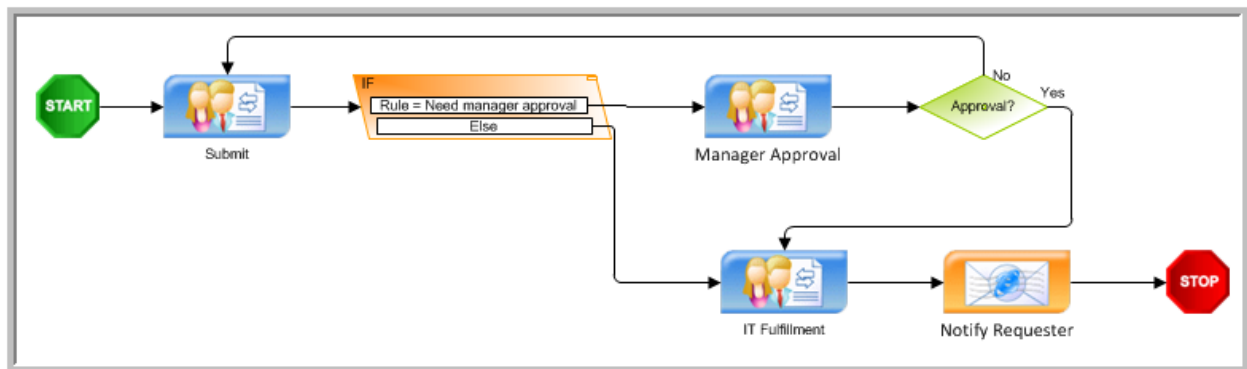
ASP.NET is a Web application framework developed and marketed by Microsoft to allow programmers to build dynamic Web sites, Web applications and Web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.

ASP.NET is one of the out of the box supported form technologies. AgilePoint leverages ASP.NET framework to deliver enterprise level BPM solution. AgilePoint provides its own web site template, which makes necessary infrastructure available to use in association with ASP.NET.

This document explains step by step process to create simple ASP.Net web site which uses AgilePoint process model.

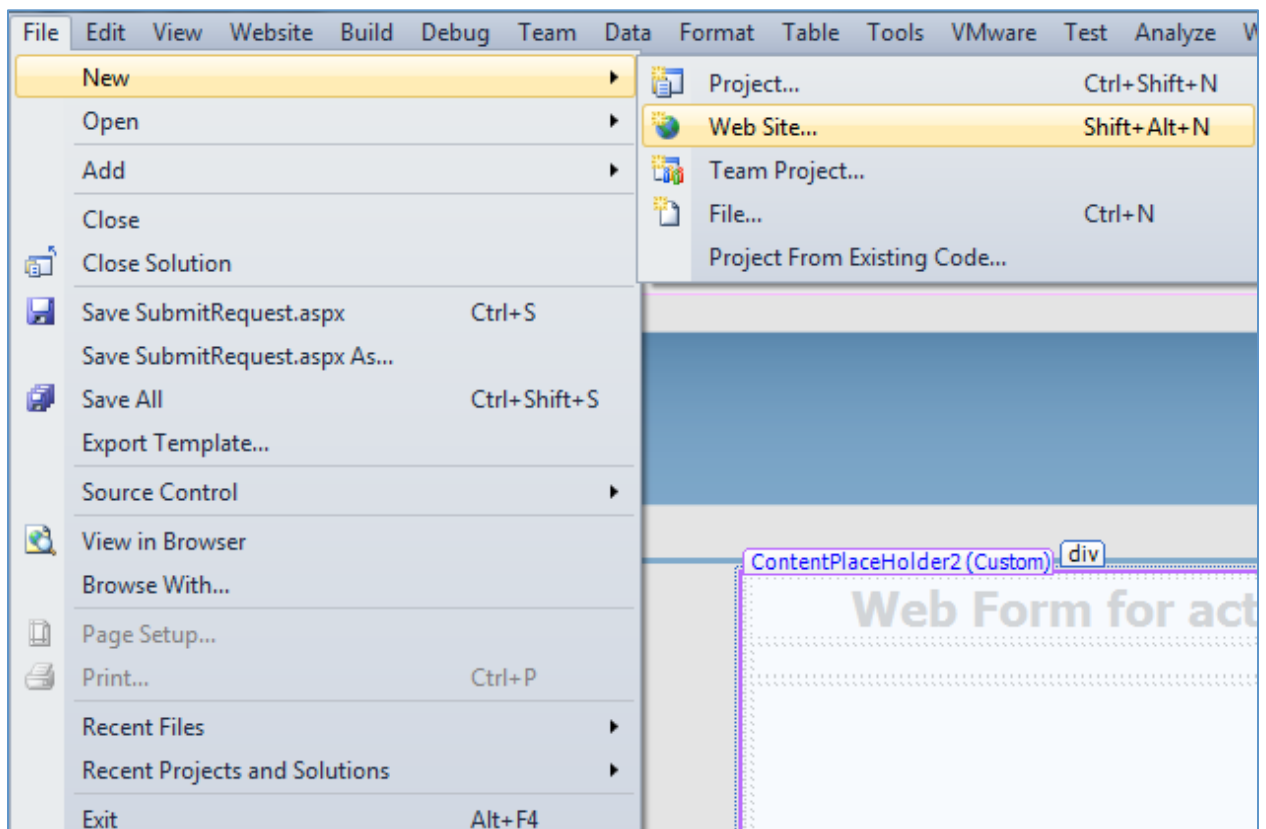
Create ASP.NET Web Site

This section explains the steps required to create an AgilePoint enabled ASP.Net web site for Windows based on the process model designed in “Process Designer” guide. The first step submits the IT equipment request, and the second manual step deals with Manger Approval. Once the request is approved by manager, it is sent to IT group to fulfill the request. Let’s create an ASP.Net web site using AgilePoint Developer component.



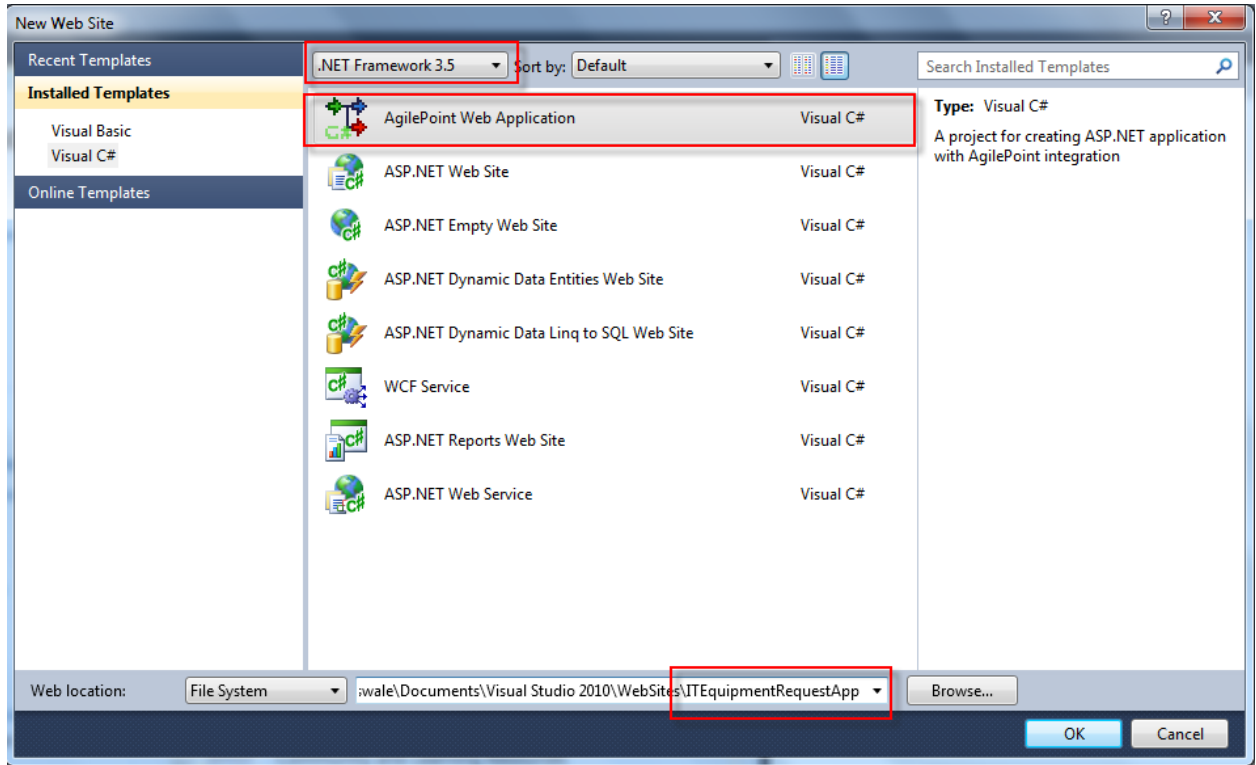
Picture: Process Design for IT Equipment

1. Launch Visual Studio.
2. Click on **File->New->Web Site...** from the menu bar.



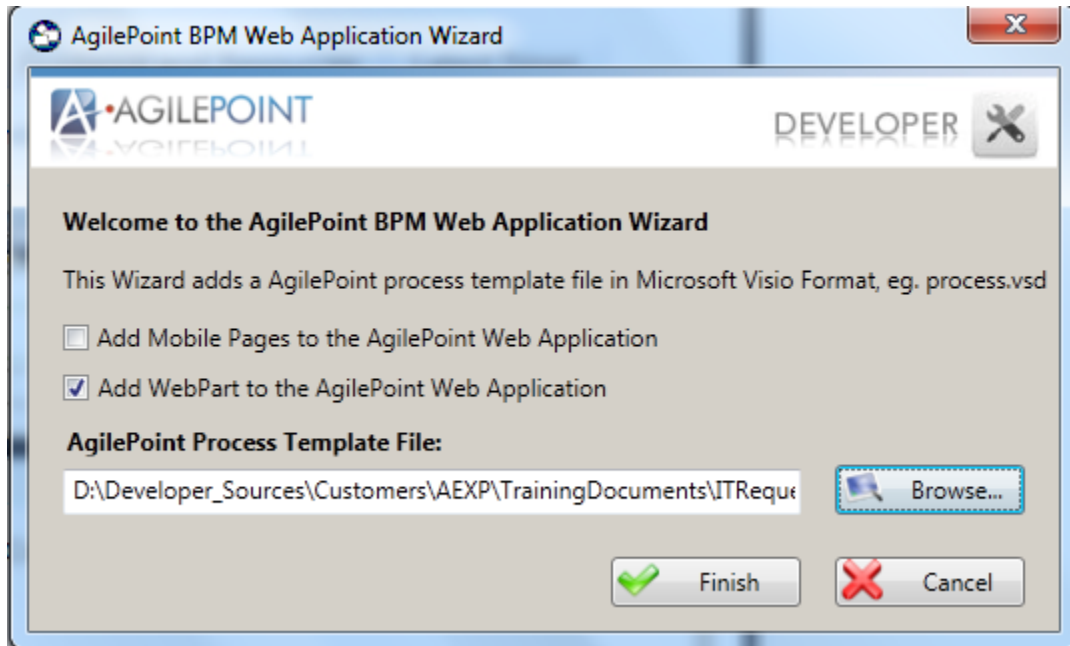
Picture: Creating ASP.Net Web Site

3. Select Visual C# templates in the left hand panel. Please make sure you have selected .Net Framework 3.5 as shown in screenshot below. Then select “**AgilePoint Web Application**” project template. Enter name of the web application as “**ITEquipmentRequestApp**”. Click “OK” button.



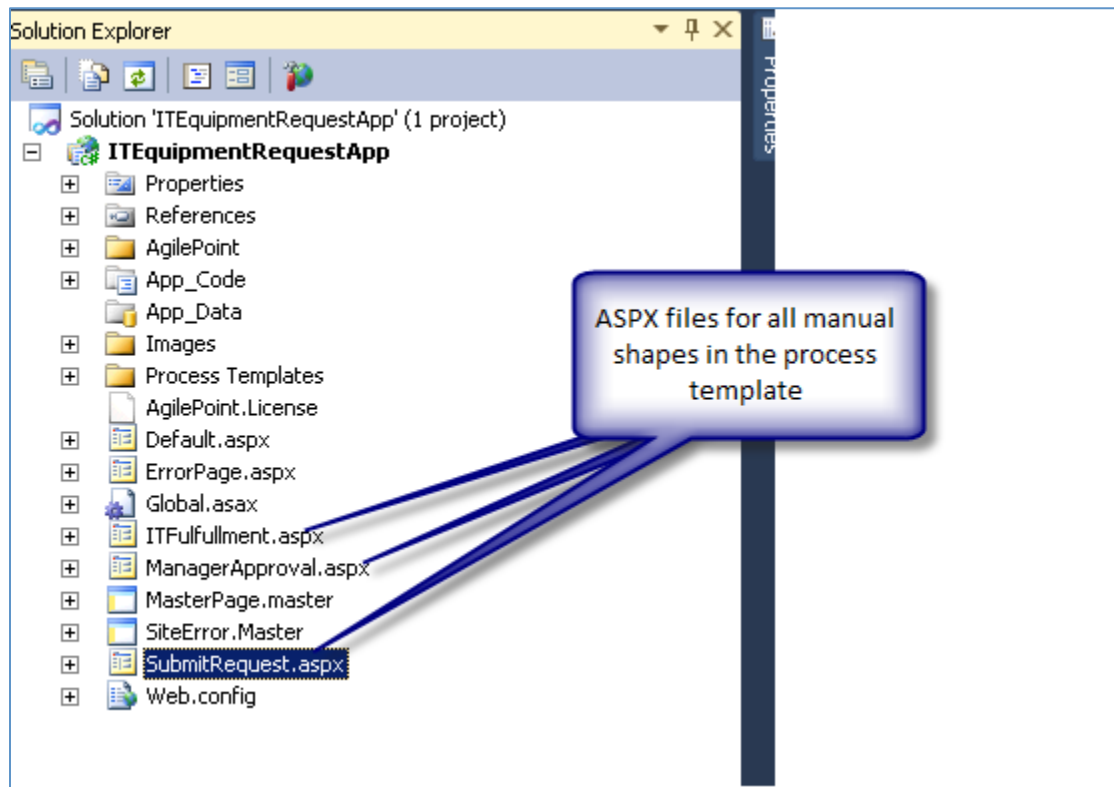
Picture: Creation of AgilePoint web application using AgilePoint Developer

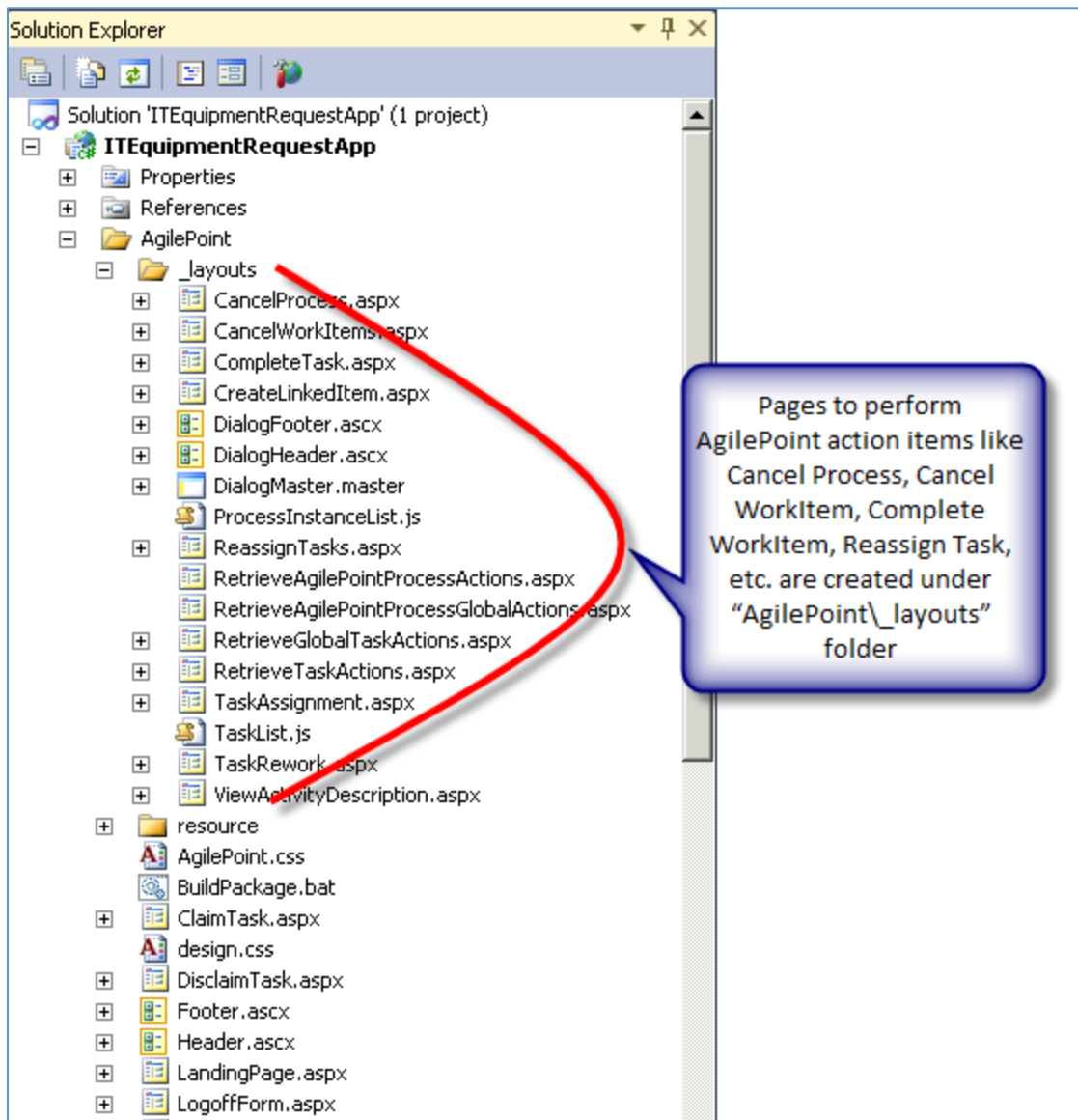
4. AgilePoint BPM Web Application Wizard starts and asks for Process Template file, which is nothing but the Visio process file. Browse to “ITRequest.vsd” file created following the “**Process Designer**” guide by clicking on “Browse” button. The check box “Add WebPart to the AgilePoint Web Application” is already selected. Do not select other check box to “Add Mobile Pages to the AgilePoint Web Application” as we are not covering that part in this guide. Click “Finish” button. It takes a minute (depending on how complex the process is) to generate the web site solution.



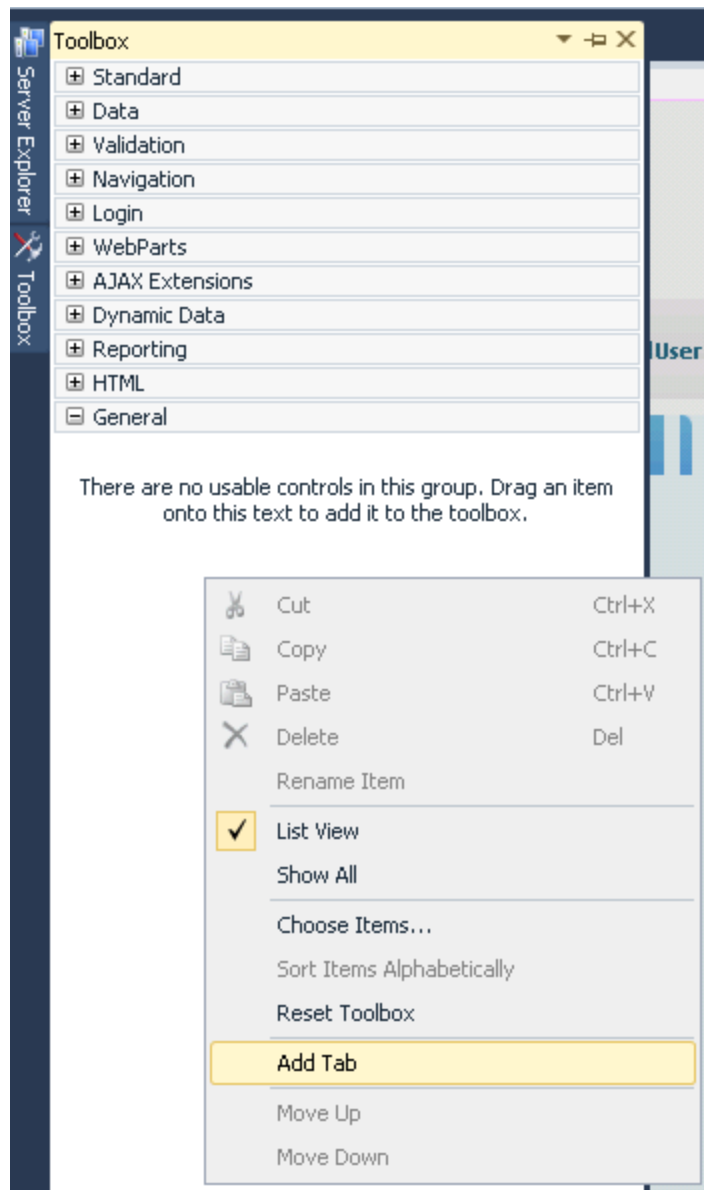
Picture: AgilePoint BPM Web Application Wizard

5. Let's have a closer look at Solution explorer and see what things are created automatically:
 1. ASPX files for all manual shapes in the process template. The name of the ASPX file is the value of “WorkToPerform” of the manual shape. E.g. SubmitRequest.aspx, ManagerApproval.aspx, and ITFulfillment.aspx. These would act as a UI for performing the manual tasks in the workflow. These are the pages where you would add your process specific code. The rest of the pages are general process administration pages and already have the required code.
 2. Pages to show AgilePoint Task List WebPart, Process Instance List WebPart, Process Viewer, log on page, log off page, etc. are created under “AgilePoint” folder.
 3. Pages to perform AgilePoint action items like Cancel Process, Cancel WorkItem, Complete WorkItem, Reassign Task, etc. are created under “AgilePoint_layouts” folder.
 4. Required resource files for multiple languages and image files.

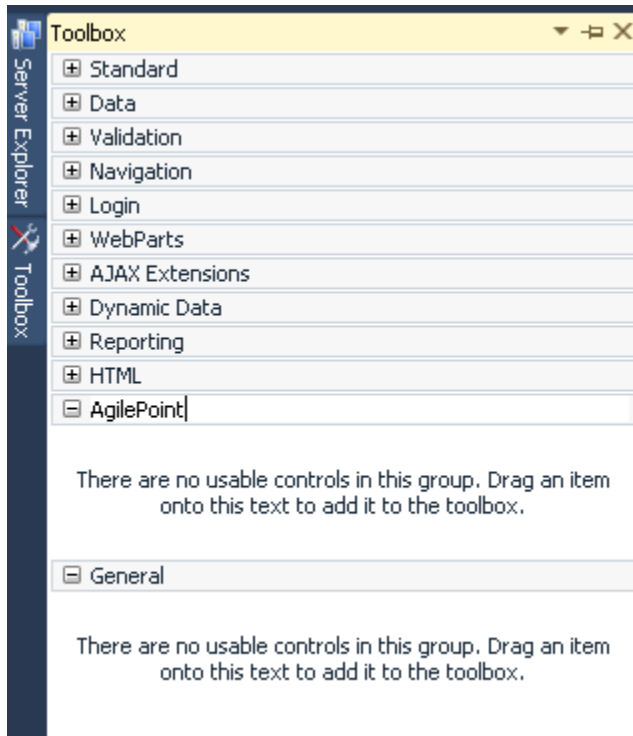




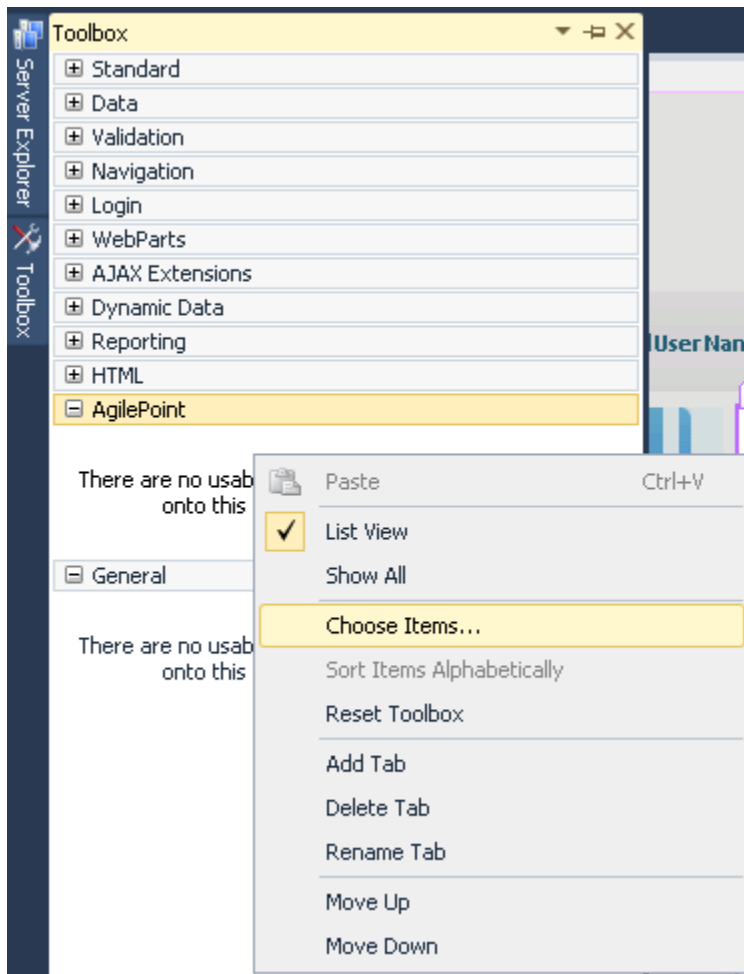
6. Since this is the first time you are using AgilePoint Website template in Visual studio, let's add AgilePoint controls on the Visual Studio toolbox. AgilePoint provides some very useful ASP.Net controls with data binding feature which can help you design your ASP.Net applications quickly. However you can also decide to use any third-party controls like any other regular ASP.Net application. However for this exercise we would use AgilePoint controls. Open Toolbox, right click and select "**Add Tab**" option.



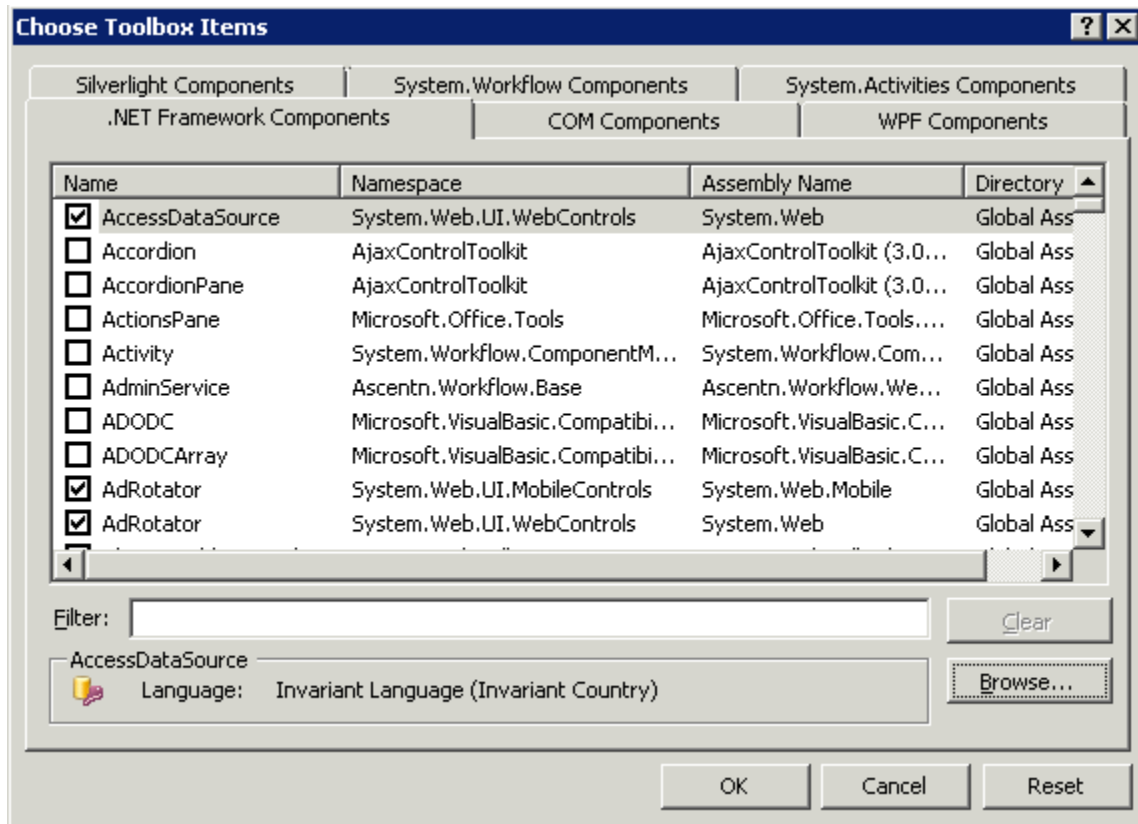
7. Name the tab as **"AgilePoint"**.



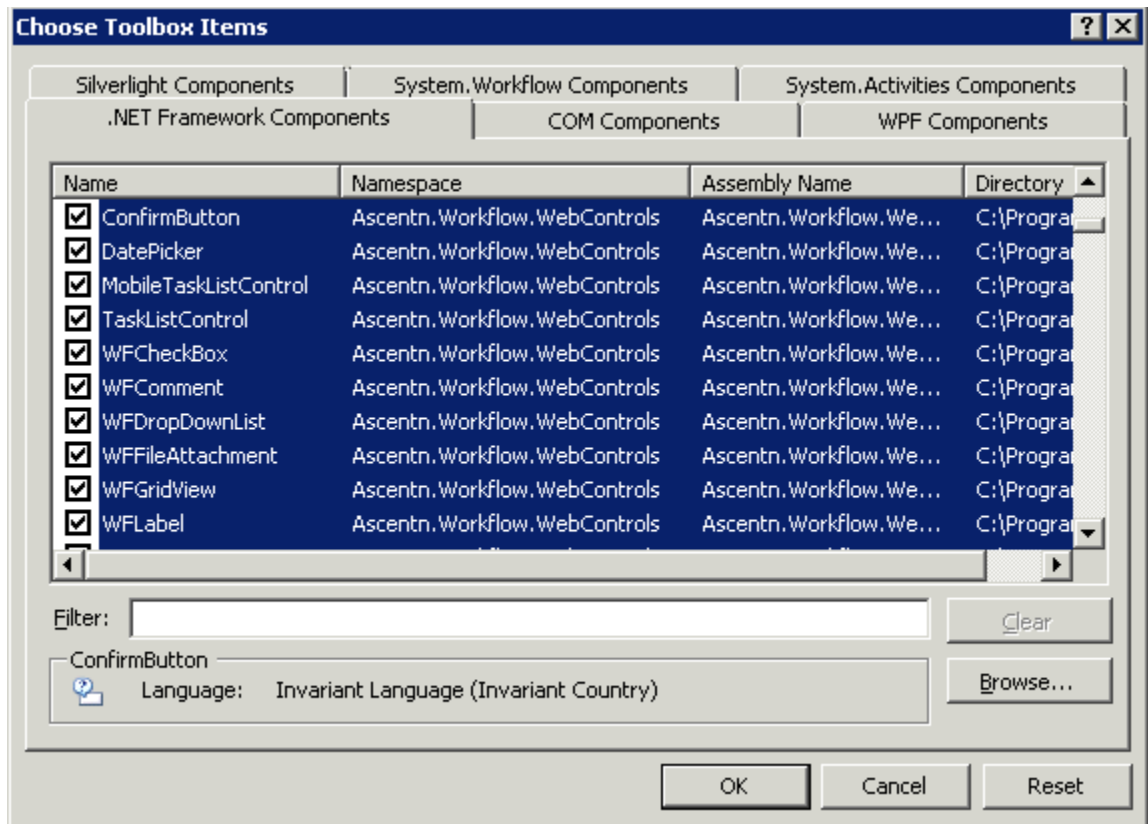
8. Right-click inside "AgilePoint" toolbox blank area and select "Choose Items..."

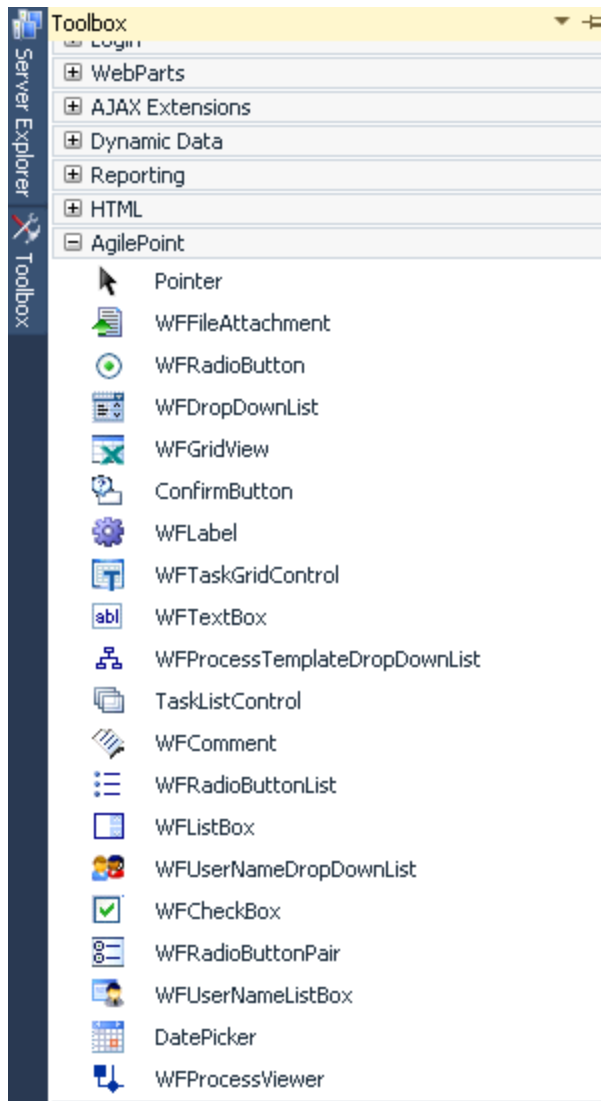


9. Click the Browse button, go to folder **"C:\Program Files (x86)\AgilePoint\AgilePoint Developer\Assembly"** and select the dll **"Ascentn.Workflow.WebControls.dll"**, and click OK.

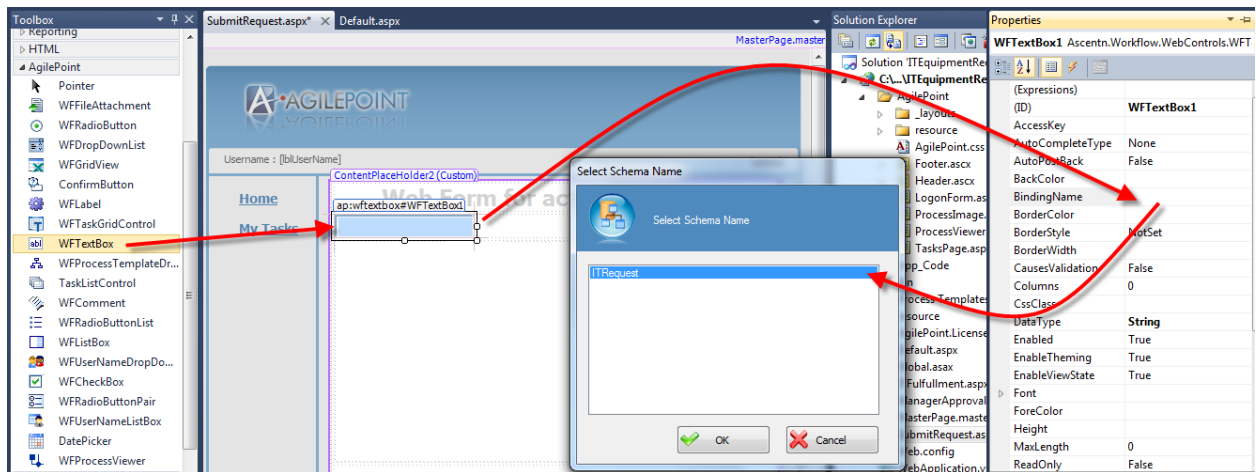


10. Select the check box of all selected items, and click OK to add those controls in AgilePoint toolbox. You can also sort the controls by namespace "**Ascentn.Workflow.WebControls**" and make sure all controls in that namespace are checked before you click OK.



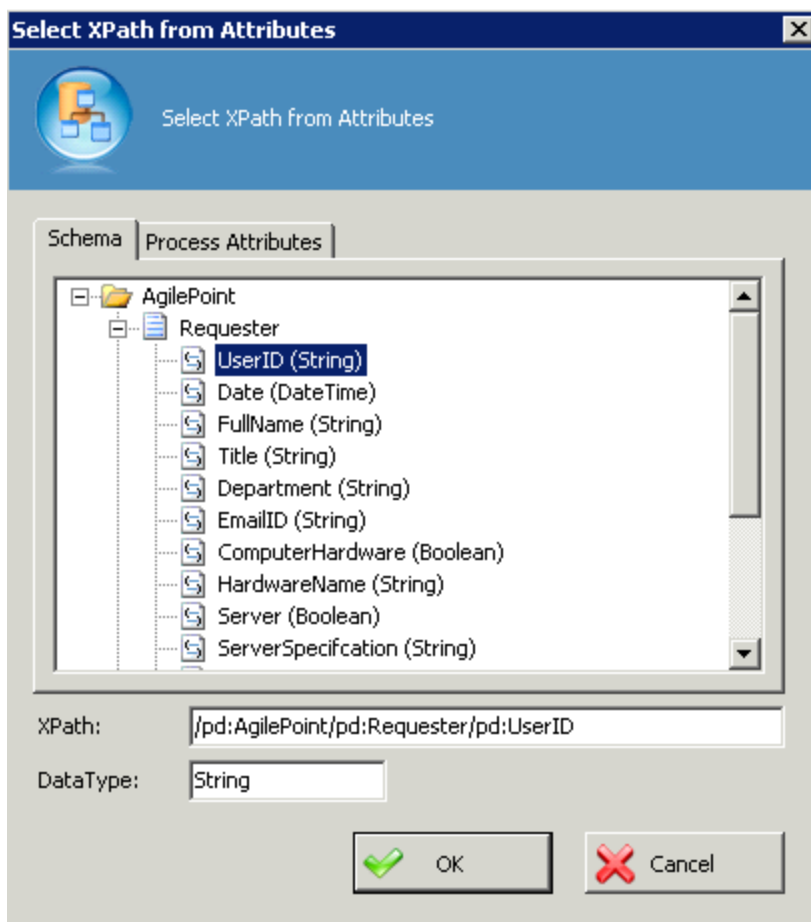


11. Now you are ready to design your task specific pages.
12. Right-click **Default.aspx** and click on “**Set as Start Page**”.
13. Open **SubmitRequest.aspx** page, and click on “**Design**” view at left bottom corner of the aspx page to open the design view of the page.
14. Drag and drop **WFTextBox** control from the AgilePoint toolbox on the page design area. Go to properties of the control and select **BindingName** property. Select the process template, “**ITRequest**” in your case when prompted to select the schema.



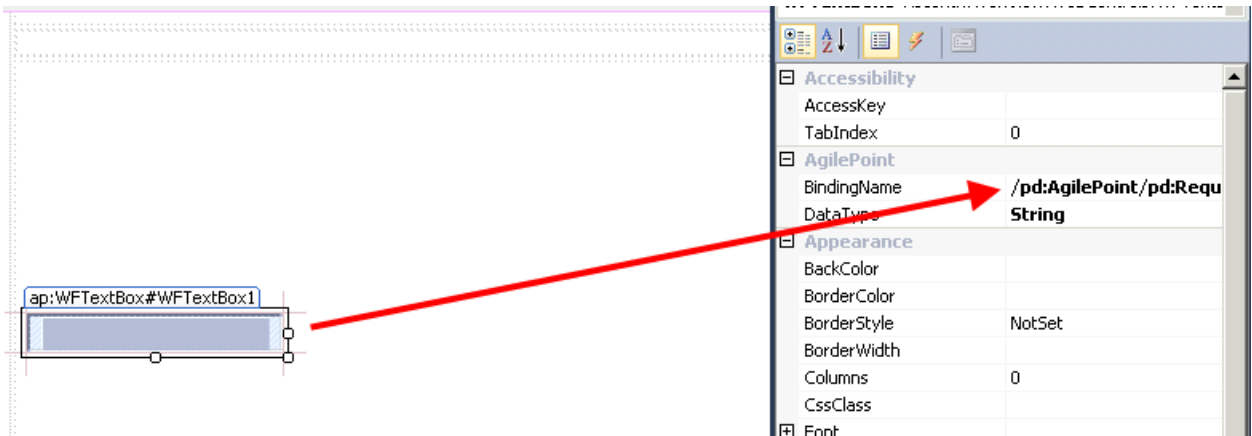
Picture: UserID textbox binding with process model

15. Select the **UserID** element from the XPath wizard.



16. This maps newly added text box to UserID element of process model schema and any value you enter in the textbox would automatically get persisted in the workflow database. BindingName is the only property different than the regular ASP.Net Textbox control and that property makes integration/binding seamless without writing any code.

Note: Leverage HTML design experience to design the page as per the requirement.



17. The following table shows the details of what other control you need to use and what it the corresponding BindingName property value. The extra properties required to configure the control are shown in Extra Properties column. Design the page by dragging and dropping these controls on design surface and set the corresponding properties as shown below. Optionally you can create a table structure in the SubmitRequest.aspx file and drag and drop the control and set the BindingName and other required properties for better layout of the screen.

Sr. No.	Label	Control	Binding Name	Extra Property
1	User ID	WFormText	/pd:AgilePoint/pd:Requester/pd:UserID	
2	Date	DatePicker	/pd:AgilePoint/pd:Requester/pd:Date	
3	Full Name	WFormText	/pd:AgilePoint/pd:Requester/pd:FullName	
4	Title	WFormText	/pd:AgilePoint/pd:Requester/pd:Title	
5	Department	WFormDropDownList	/pd:AgilePoint/pd:Requester/pd:Department	Items: Finance, HR, Marketing and Sales
6	Email Address	WFormText	/pd:AgilePoint/pd:Requester/pd:EmailID	
7	Computer Hardware	WFormCheckBox	/pd:AgilePoint/pd:Requester/pd:ComputerHardware	
8	Computer Hardware Required	WFormDropDownList	/pd:AgilePoint/pd:Requester/pd:HardwareName	Items: Monitor, Laptop, Desktop and Projector
9	Server	WFormCheckBox	/pd:AgilePoint/pd:Requester/pd:Server	
10	Technical Specification	WFormText	/pd:AgilePoint/pd:Requester/pd:ServerSpecifcation	TextMode: MultiLine
11	Mobile Device	WFormCheckBox	/pd:AgilePoint/pd:Requester/pd:MobileDevice	
12	Mobile	WFormDropDownList	/pd:AgilePoint/pd:Requester/pd:DeviceName	Items: iPad,

	Equipment Required			Cell Phone
13	Land Line Phone	WFCheckBox	/pd:AgilePoint/pd:Requester/pd:LandlinePhone	
14	Requester's Comments	WFTextBox	/pd:AgilePoint/pd:Requester/pd:RequesterComment	TextMode: MultiLine

18. After you configure the page, it will look like as shown below. Optionally you can add images, title and icons. These are regular ASP.Net features, and are not covered here

The screenshot displays a web form titled "Web Form for activity 'Submit'". The form is organized into a grid-like structure with the following elements:

- User ID**: A text input field.
- Date**: A date picker showing "4/10/2013" with a calendar icon.
- Full Name**: A text input field.
- Title**: A text input field.
- Department**: A dropdown menu with "Finanace" selected.
- Email Address**: A text input field.
- Computer Hardware**: A checkbox labeled "Computer Hardware Required" with a dropdown menu showing "Monitor".
- Server**: A checkbox labeled "Server".
- Technical Specification**: A large text area with a vertical scrollbar.
- Mobile Devices**: A checkbox labeled "Mobile Equipment Required Mobile Equipment Required" with a dropdown menu showing "iPad".
- Land Line Phone**: A checkbox labeled "Land Line Phone".

The screenshot shows a web form with a title bar that says "Requester's Comments". Below the title bar is a large, empty text area with a vertical scrollbar on the right side. At the bottom right of the form is a button labeled "Submit". The entire form is enclosed in a light blue border with a dotted pattern.

19. Once the design part is done, turn to code behind file. The Page_Load event method has some autogenerated code provided by AgilePoint component to save coding effort. Since SubmitRequest is the first page, process initialization needs to be done from the code of this page when user clicks Submit button. Change the code of SubmitRequest class as shown below. As you can see from code below, you just have to add a couple of extra lines of code and any UI logic like validation or conditionally enabling/disabling controls but rest of the data persistence logic is automatically handled for you using the data binding feature available in AgilePoint's ASP.Net framework

C# Code

```
public partial class SubmitRequest : Ascentn.Workflow.WebControls.WFWorksheetPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!string.IsNullOrEmpty(Request.QueryString["WID"]))
        {
            WFManualWorkItem workItem;
            try
            {
                workItem =
                GetAPI(Session).GetWorkItem(Request.QueryString["WID"]);
            }
            catch (Exception excp)
            {
                throw excp;
            }
            if (workItem != null &&
                !workItem.UserID.Equals(HttpContext.Current.User.Identity.Name,
                StringComparison.InvariantCultureIgnoreCase))
            {
                Response.Write("<script language='javascript'>alert('This task
is not assigned to you. Click OK to redirect to Tasks page.');"
                window.location.href='AgilePoint/TasksPage.aspx';</script>");
            }
        }

        protected void btnSubmit_Click(object sender, System.EventArgs e)
```

```

{
    if (string.IsNullOrEmpty(Request.QueryString["WID"]))
    {
        string processModel =
System.Configuration.ConfigurationManager.AppSettings["ProcessModel"];
        base.CreateProcInst(processModel, processModel + "-" + UUID.GetID());
        Response.Redirect("AgilePoint/TasksPage.aspx", true);
    }
    else
    {
        base.CompleteWorkItem();
        Response.Redirect("AgilePoint/TasksPage.aspx", true);
    }
}
}

```

Code: Page_Load and btnSubmit_Click event methods on SubmitRequest web page

20. Next Go to ManagerApproval.aspx page. We can reuse the design of SubmitRequest.aspx page to design ManagerApproval.aspx page. Copy the contents from SubmitRequest.aspx and paste it in ManagerApproval.aspx as show in images below.





21. Switch to the Design view of ManagerApproval.aspx page the page will look like this:

Web Form for activity ' Manager Approval'

User ID Error Creating Control - WFTextBox1 This control cannot be displayed because its TagPrefix is not registered in this Web Form.	Date Error Creating Control - DatePicker1 This control cannot be displayed because its TagPrefix is not registered in this Web Form.
Full Name Error Creating Control - WFTextBox2 This control cannot be displayed because its TagPrefix is not registered in this Web Form.	Title Error Creating Control - WFTextBox3 This control cannot be displayed because its TagPrefix is not registered in this Web Form.
Department Error Creating Control - WFDropDownList1 This control cannot be displayed because its TagPrefix is not registered in this Web Form.	Email Address Error Creating Control - WFTextBox4 This control cannot be displayed because its TagPrefix is not registered in this Web Form.
Error Creating Control - WFCheckBox1 This control cannot be displayed because its TagPrefix is not registered in this Web Form.	

22. This is because missing Assembly directive for AgilePoint controls in ManagerApproval.aspx page. Add this line at the top in ManagerApproval.aspx page:

```
<%@ Register Assembly="Ascentn.Workflow.WebControls"
Namespace="Ascentn.Workflow.WebControls" TagPrefix="ap" %>
```

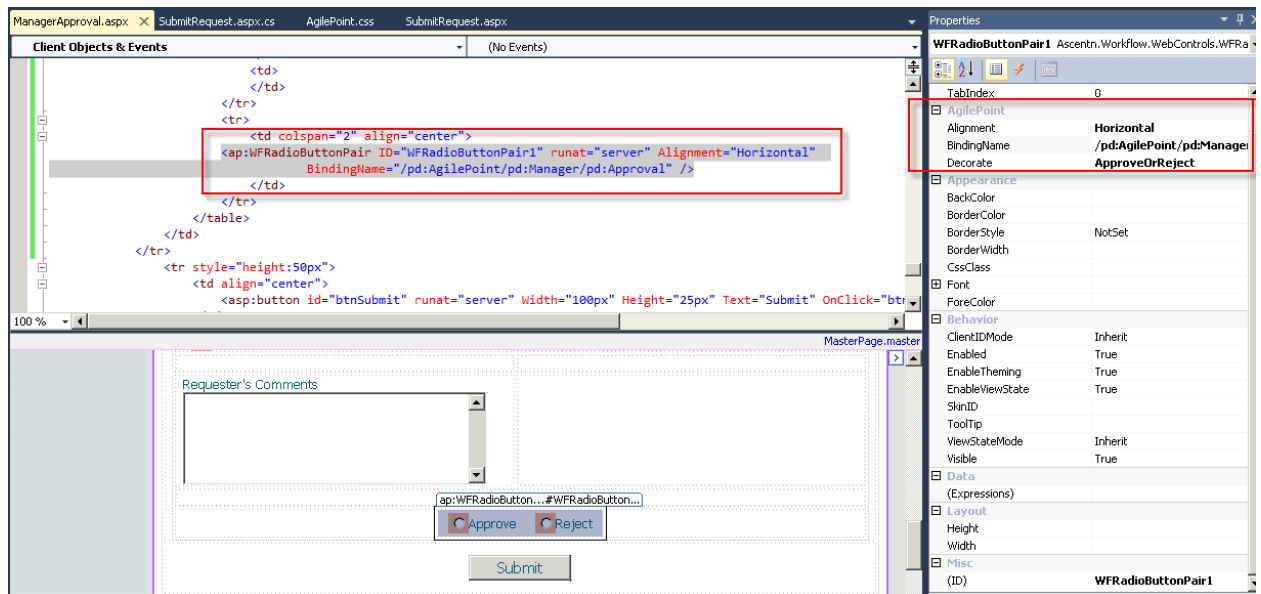
```
ManagerApproval.aspx SubmitRequest.aspx.cs AgilePoint.css SubmitRequest.aspx
Client Objects & Events (No Events)
<%@ Register TagPrefix="uc1" TagName="Header" Src="AgilePoint/Header.ascx" %>
<%@ Register TagPrefix="uc1" TagName="Footer" Src="AgilePoint/Footer.ascx" %>
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="false" Codebehind="ManagerAppro
<%@ Register Assembly="Ascentn.Workflow.WebControls" Namespace="Ascentn.Workflow.WebControls"
TagPrefix="ap" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">
<%--<form id="form1" onload="Page_Load" runat="server">--%>
```

ContentPlaceHolder2 (Custom)

Web Form for activity ' Manager Approval'

User ID Full Name Department Finance ☐ Computer Hardware Computer Hardware Required Monitor ☐ Server Technical Specification 	Date 3/16/201 ... Title Email Address
--	--

23. Make all the controls read only by setting either the ReadOnly property or Enabled property to False by right clicking on individual controls and going to their property panel.
24. Next we will add new controls required to perform work on manager approval screen. Add WFRadioButtonPair control and bind it to “/pd:AgilePoint/pd:Manager/pd:Approval” element from the schema. The page should look very similar to SubmitRequest page except additional approve/reject radio button.

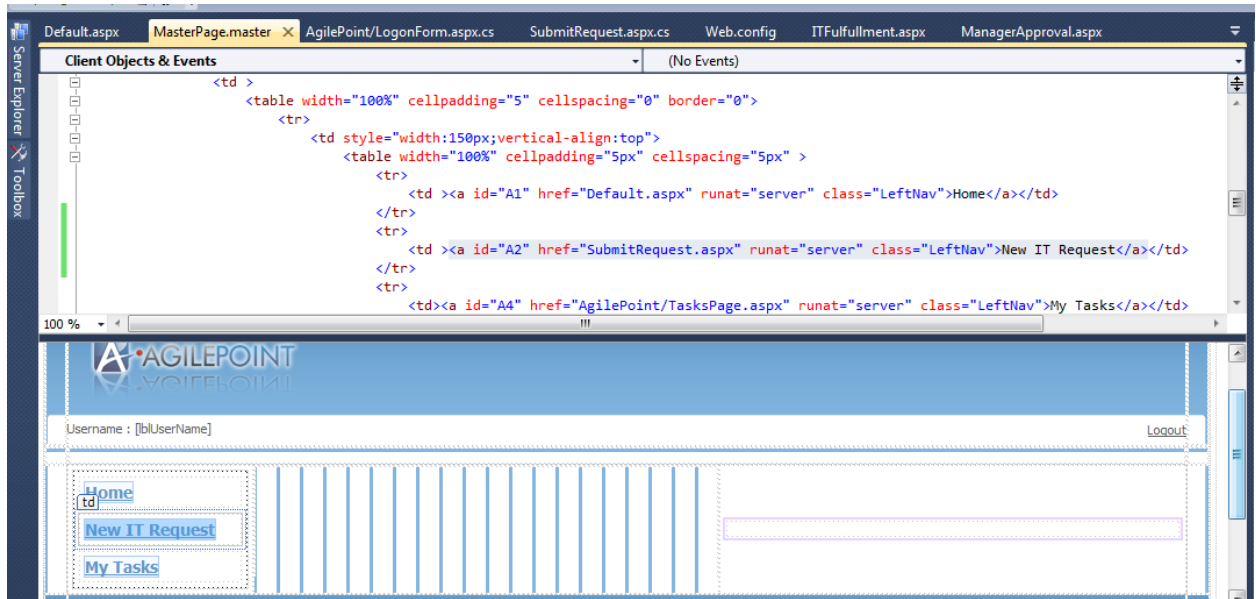


Note: There is no need to add or change any code in the code behind file since required code is already generated for you unless you wanted to add any UI logic for formatting. As you can see the AgilePoint framework automatically handles the entire logic of loading the controls with values entered on the submit screen and also persists back any changes made on this page, back to the workflow database. This is possible because of the data binding feature. This data can be optionally captured in your custom database for reporting purpose without writing any code using AgilePoint data services feature. However in this simple exercise we will not use that feature.

25. Repeat the same steps to design ITFulfillment.aspx page and change the text on Submit button to "Complete".

Web Form for activity 'IT Fulfillment'	
User ID	Date
	4/15/2013 ...
Full Name	Title
Department	Email Address
Finance	
☐ Computer Hardware	
☐ Server	
Technical Specification	
☐ Mobile Devices	
Mobile Equipment Required Mobile Equipment Required	
iPad	
☐ Land Line Phone	
Requester's Comments	
Complete	

26. Open MasterPage.master. In a “Split View” of the page, select the link “Home” , copy the corresponding `<tr></tr>` and paste it just below it. Modify the value of the href attribute to “SubmitRequest.aspx” which is the page to create new request. Also change the text to “**New IT Request**”.



27. Build the web application.

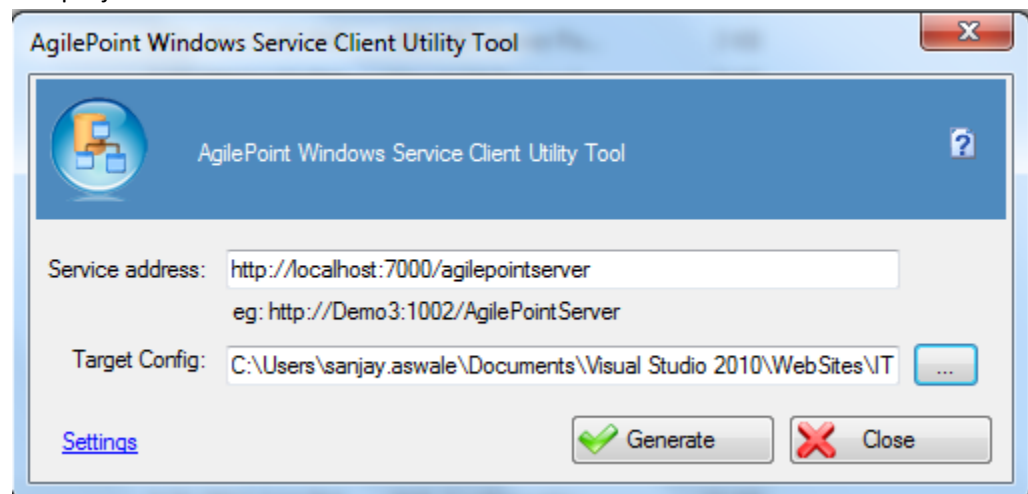
Running a deployed application

Special Notes:

1. If your AgilePoint Server is IIS version then specify your AgilePoint Server URL in “ServerUrl” node in appsettings. Also make sure “HostingMechanism” is set to “IIS”.



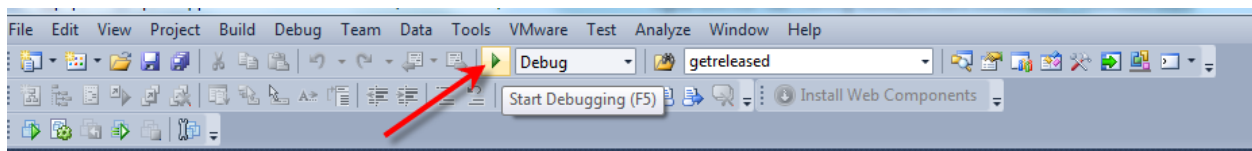
2. If your AgilePoint Server is WCF version then follow these steps:
 - a. Change "HostingMechanism" to "WCF".
 - b. Open AgilePointWindowsServiceClientUtilityTool.exe tool which will in AgilePoint Server's install location (C:\Program Files\AgilePoint\AgilePoint Server\SVCUtilityTool)
 - c. Specify THE AgilePoint Server URL. Click Browse to select the web.config file of our web site project.



- d. Click Generate. This will update the required binding information, url/port information in the web.config.

Now you are ready to run the application.

1. Click the green play button in the Visual Studio to start your application.

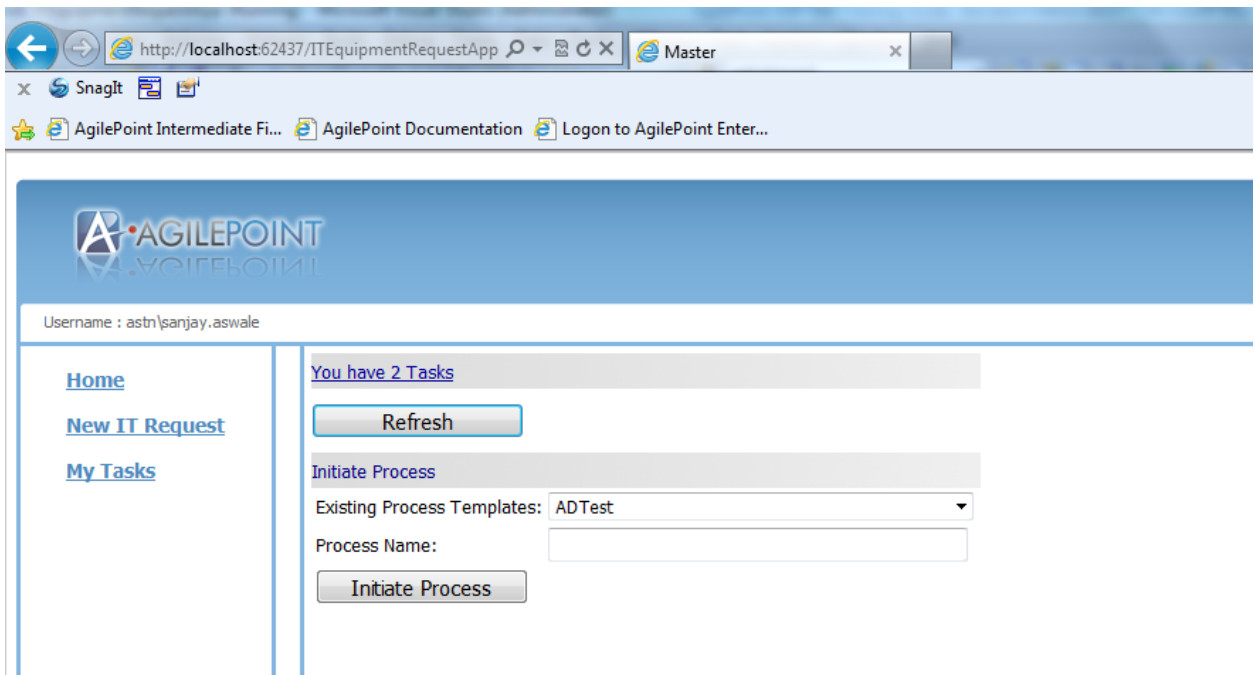


3. Provide your credentials to log in to the application and click on Logon Button



The screenshot shows a login form with three input fields and a button. The first field is labeled "Domain :" and contains the text "Demo3". The second field is labeled "User Name :" and contains the text "Bob.Hope". The third field is labeled "Password :" and contains seven dots. Below the password field is a button labeled "Logon". At the bottom of the form, there is a copyright notice: "Copyright © 2013 AgilePoint Inc. All Rights Reserved".

4. After log in you will see a default page and a built-in process initiator functionality which we are not going to use. You can modify the default.aspx page to change the layout of this page.



The screenshot shows the application's default page after login. The browser's address bar shows the URL "http://localhost:62437/ITEquipmentRequestApp". The page has a blue header with the AgilePoint logo and the text "AGILEPOINT". Below the header, the username "Username : astn\sanjay.aswale" is displayed. On the left side, there is a navigation menu with links: "Home", "New IT Request", and "My Tasks". On the right side, there is a section titled "You have 2 Tasks" with a "Refresh" button. Below this, there is a section titled "Initiate Process" with a dropdown menu for "Existing Process Templates" (currently set to "ADTest"), a text input field for "Process Name", and an "Initiate Process" button.

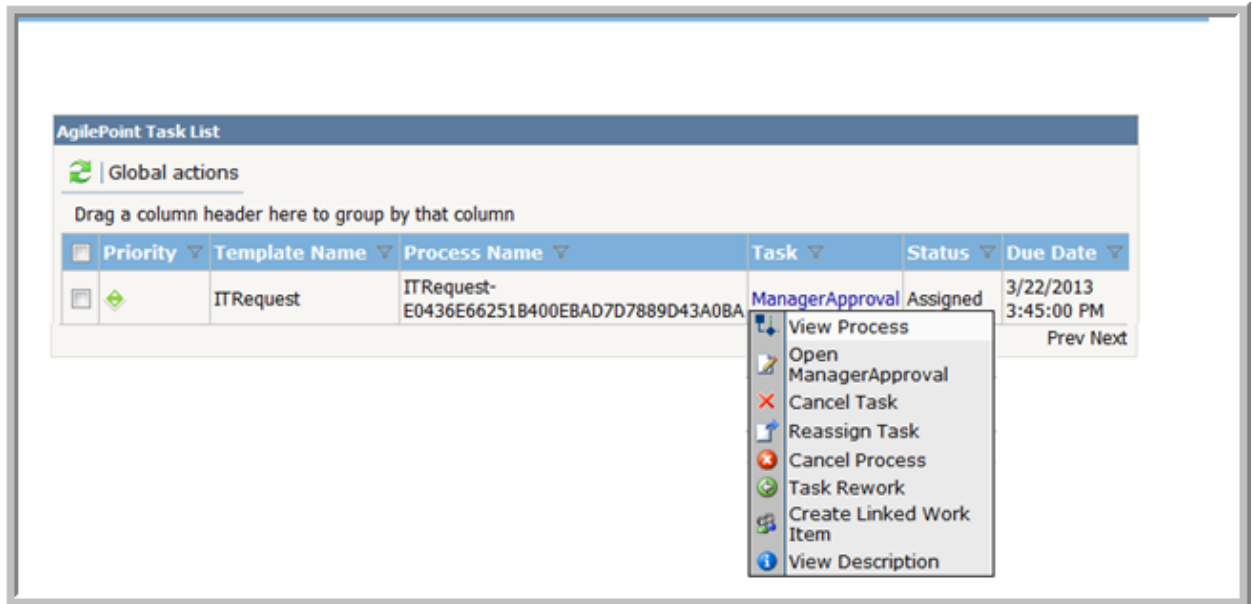
5. Click “New IT Request” to create new request. Provide the information, and click Submit.

User ID	Date
121	3/21/2013 ...
Full Name	Title
don	sales manager
Department	Email Address
Sales	don@tusca.com
☒ Computer Hardware Computer Hardware Required Monitor	
☐ Server Technical Specification 	
☒ Mobile Devices Mobile Equipment Required Cell Phone	
☐ Land Line Phone Requester's Comments 	
Submit	

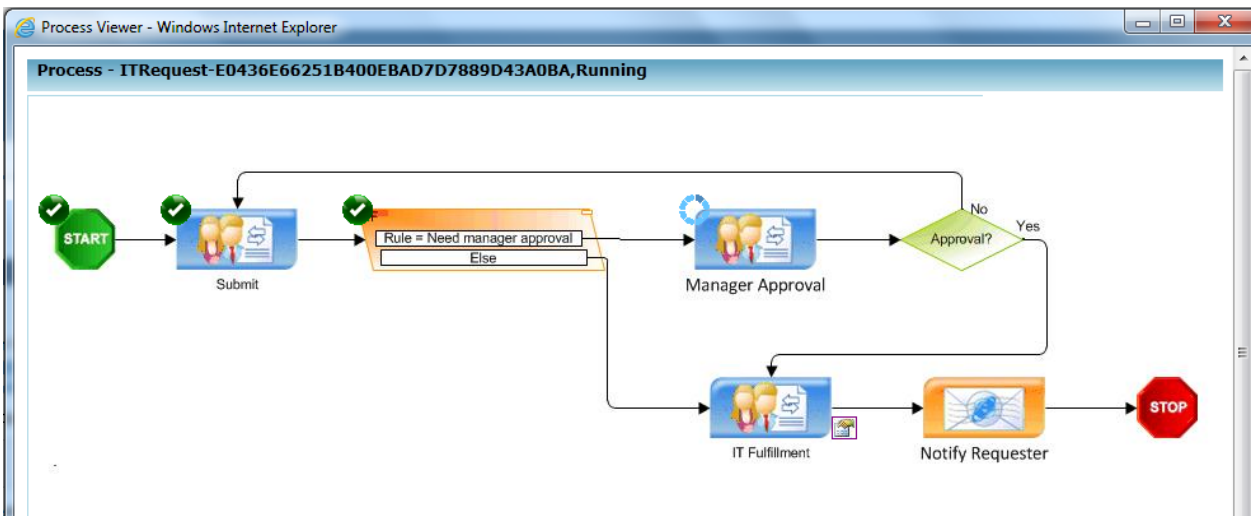
6. This will take you to your task list page.

AgilePoint Task List						
Global actions						
Drag a column header here to group by that column						
Priority ▾	Template Name ▾	Process Name ▾	Task ▾	Status ▾	Due Date ▾	
	ITRequest	ITRequest-E0436E66251B400EBAD7D7889D43A08A	ManagerApproval	Assigned	3/22/2013 3:45:00 PM	
					1 of 1	Prev Next

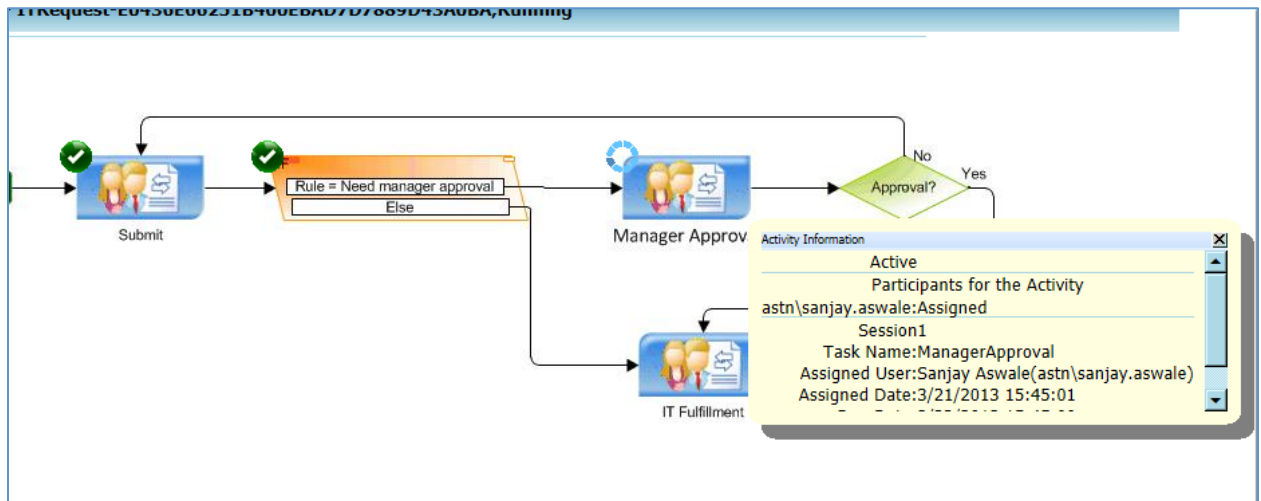
7. Click on the task link to show up available actions you can perform on the task.



8. Click on View Process to see the runtime visual



9. Click on the shape to see more information about it.



10. Next we will complete the manager approval task. Since in the initial process design we selected same user to be the participant of all 3 manual steps you do not need to log off. This can be changed to unique users or roles after you execute simple demo successfully.

11. Click on “**Open ManagerApproval**” to open the task. You can see all information entered while submitting new request is automatically available in approval screen. This is because of the Binding Name property on each control that ties the data between process and UI.

User ID	Date
121	3/21/2013
Full Name	Title
don	sales manager
Department	Email Address
Sales	don@tusca.com
☒ Computer Hardware	
Computer Hardware Required	
Monitor	
☐ Server	
Technical Specification	
☒ Mobile Devices	
Mobile Equipment Required	
Cell Phone	
☐ Land Line Phone	
Requester's Comments	
☒ Approve ☐ Reject	

12. Select Approve radio button and click on Submit.
13. Navigate back to the task list and open the ITFulfillment task, and click Complete to finish the process.

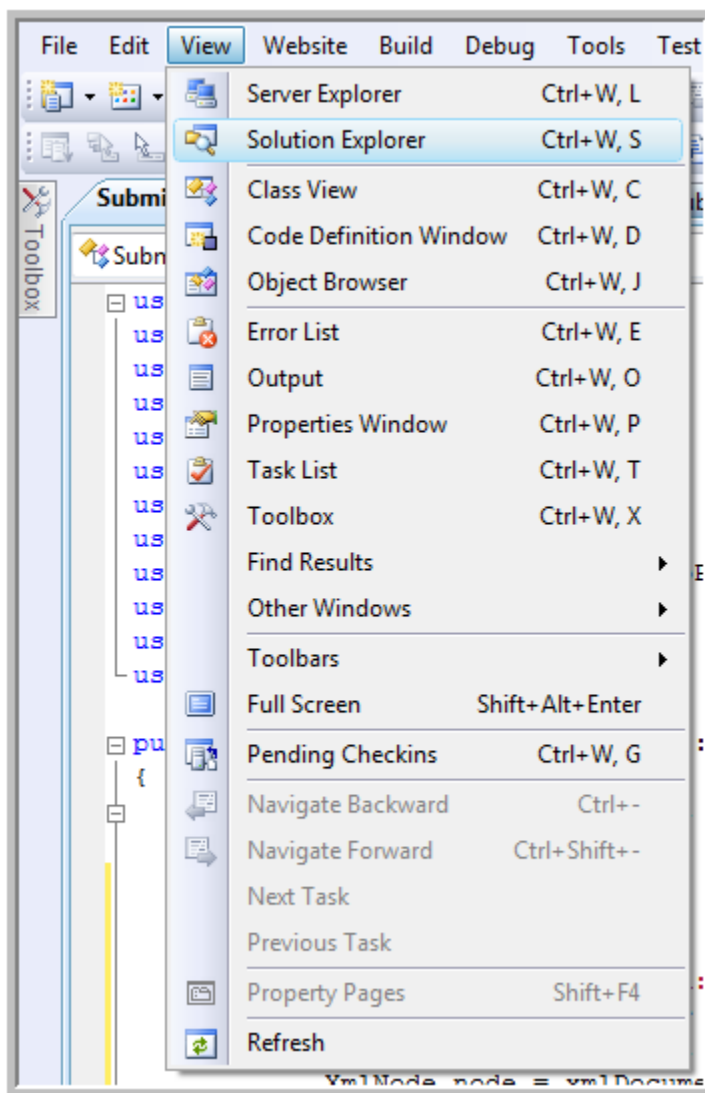
Full Name	Title
don	sales manager
Department	Email Address
Sales	don@tusca.com
☒ Computer Hardware	
Computer Hardware Required	
Monitor	
☐ Server	
Technical Specification	
☒ Mobile Devices	
Mobile Equipment Required	
Cell Phone	
☐ Land Line Phone	
Requester's Comments	
☒ Approve ☐ Reject	
Complete	

Advanced Scenarios

Include a repeating section using the WFGridView Control

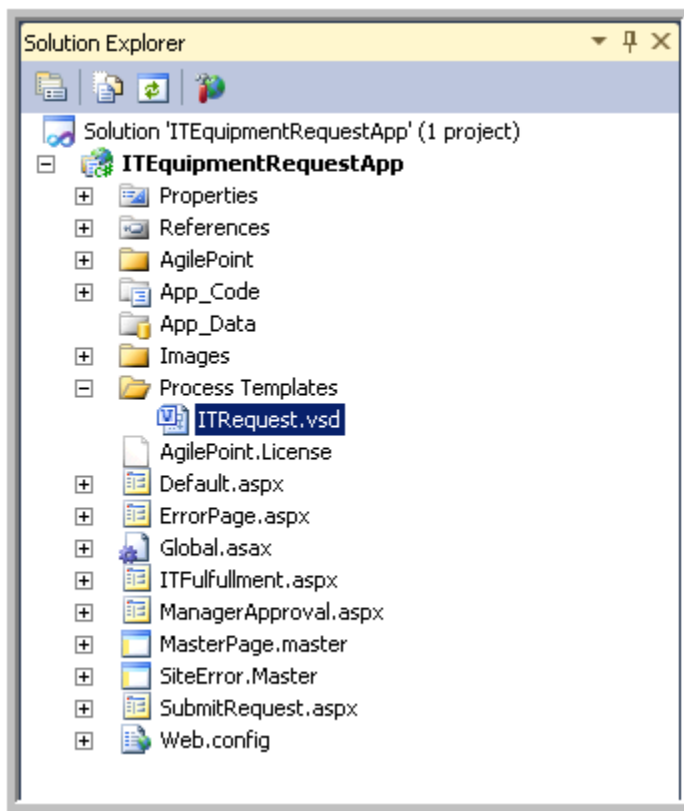
Now that the simple ASP.Net web application exercise covered in previous section is successfully completed and executed, we will try to accomplish some advance scenario within that where we will introduce a repeating node in the schema and bind it to WFGridView control.

1. Click on **View->Solution Explorer** from the menu bar.

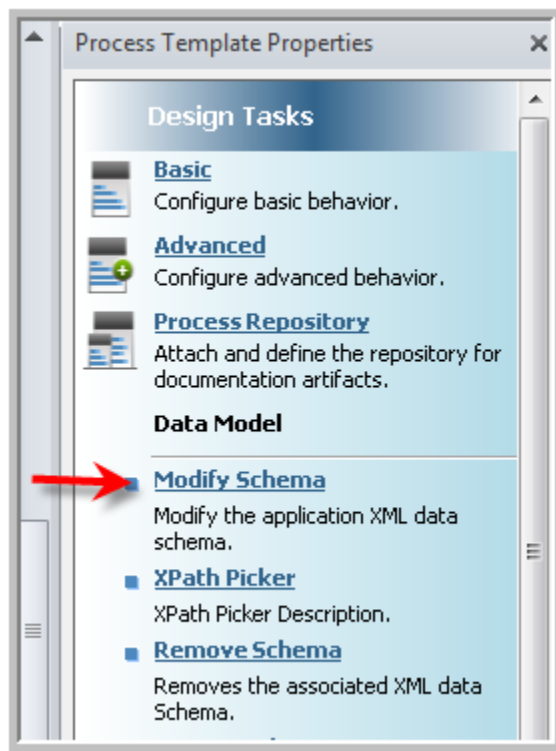


Picture: Open Solution explorer.

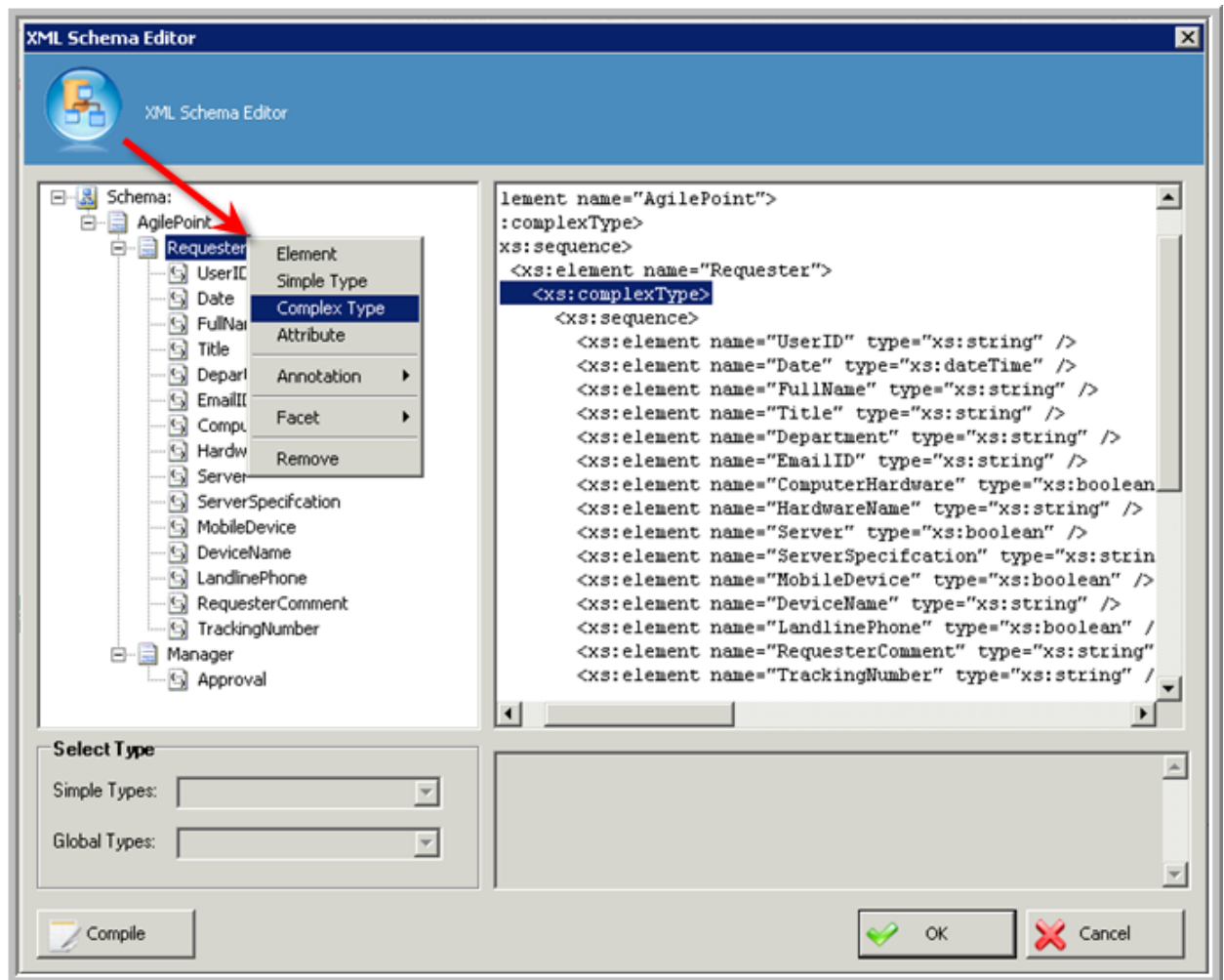
2. Double-click “ITRequest.vsd” from Solution Explorer. It will take you to AgilePoint Envision where you can modify your process design and schema.



3. Click on “**Modify Schema**” link in the property panel on right hand side.

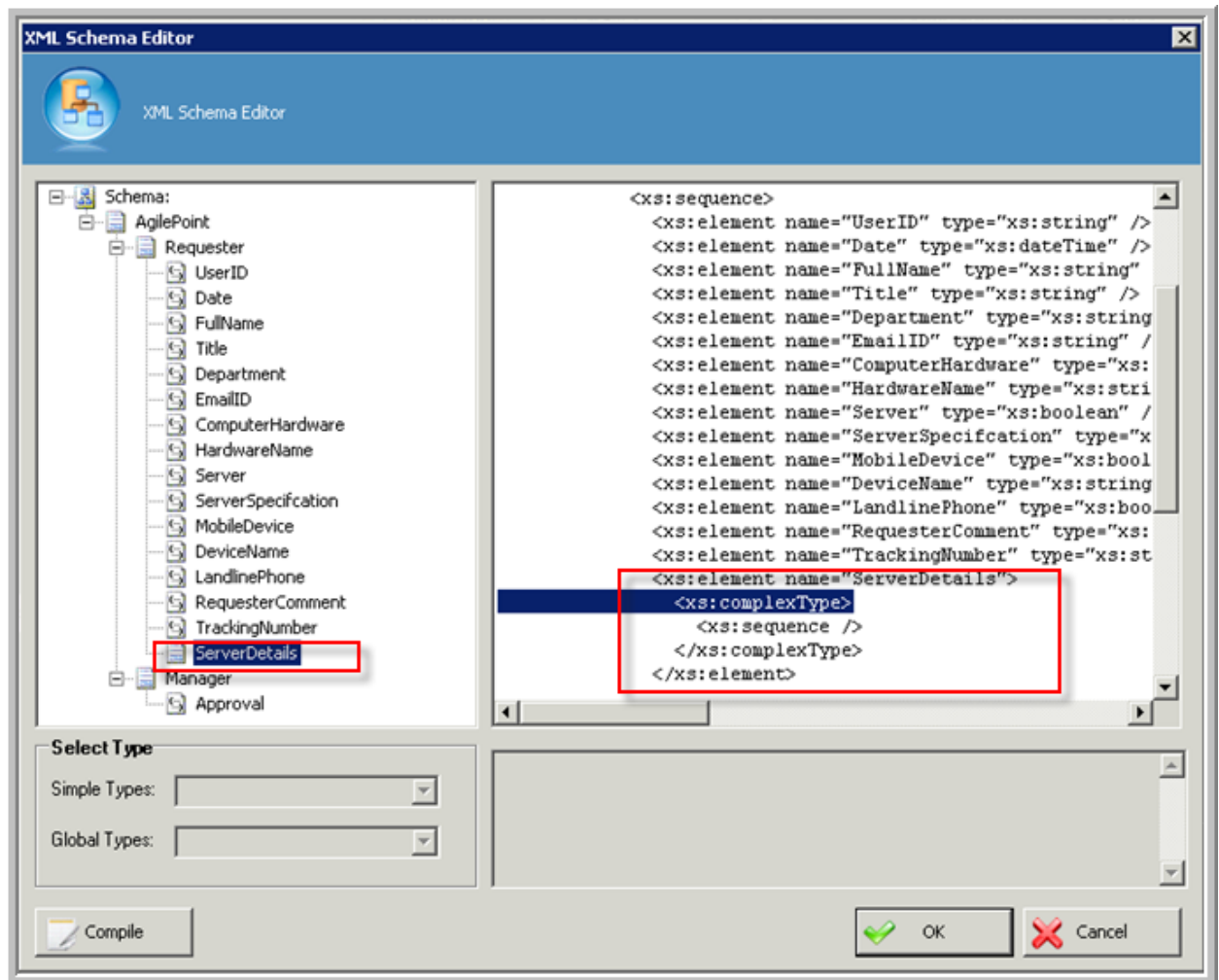


4. Select “**Requester**” node by left click (do not select any node by right click). Then right click on selected node and click “Complex Type” option to add a category for ServerDetails information where user can add multiple server requirements. Name the newly created complex type as “ServerDetails”.



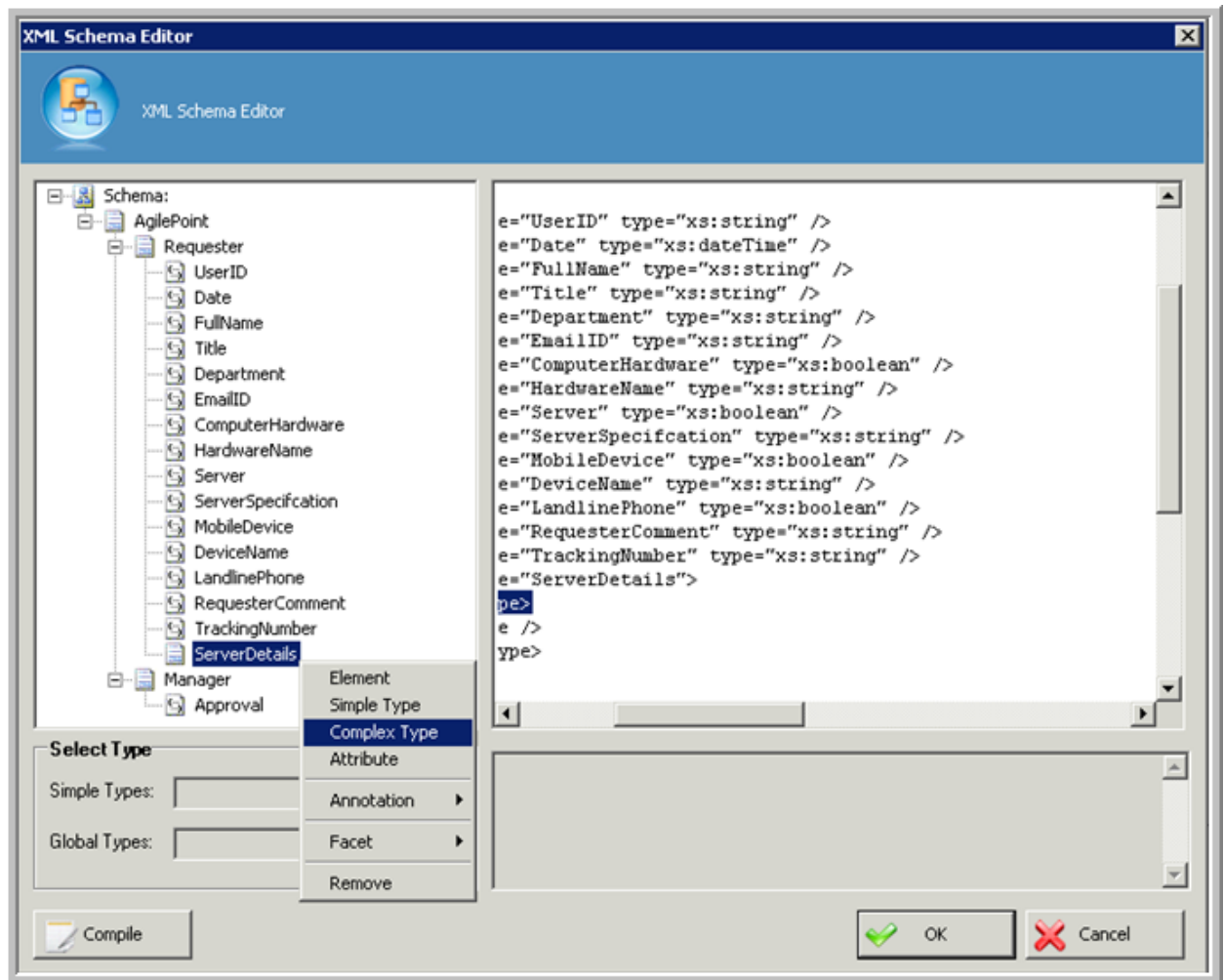
Picture: "Create Schema" dialog box to add "ServerDetails" node

5. Once the node is added to the schema, it should appear in the schema tree structure as shown below



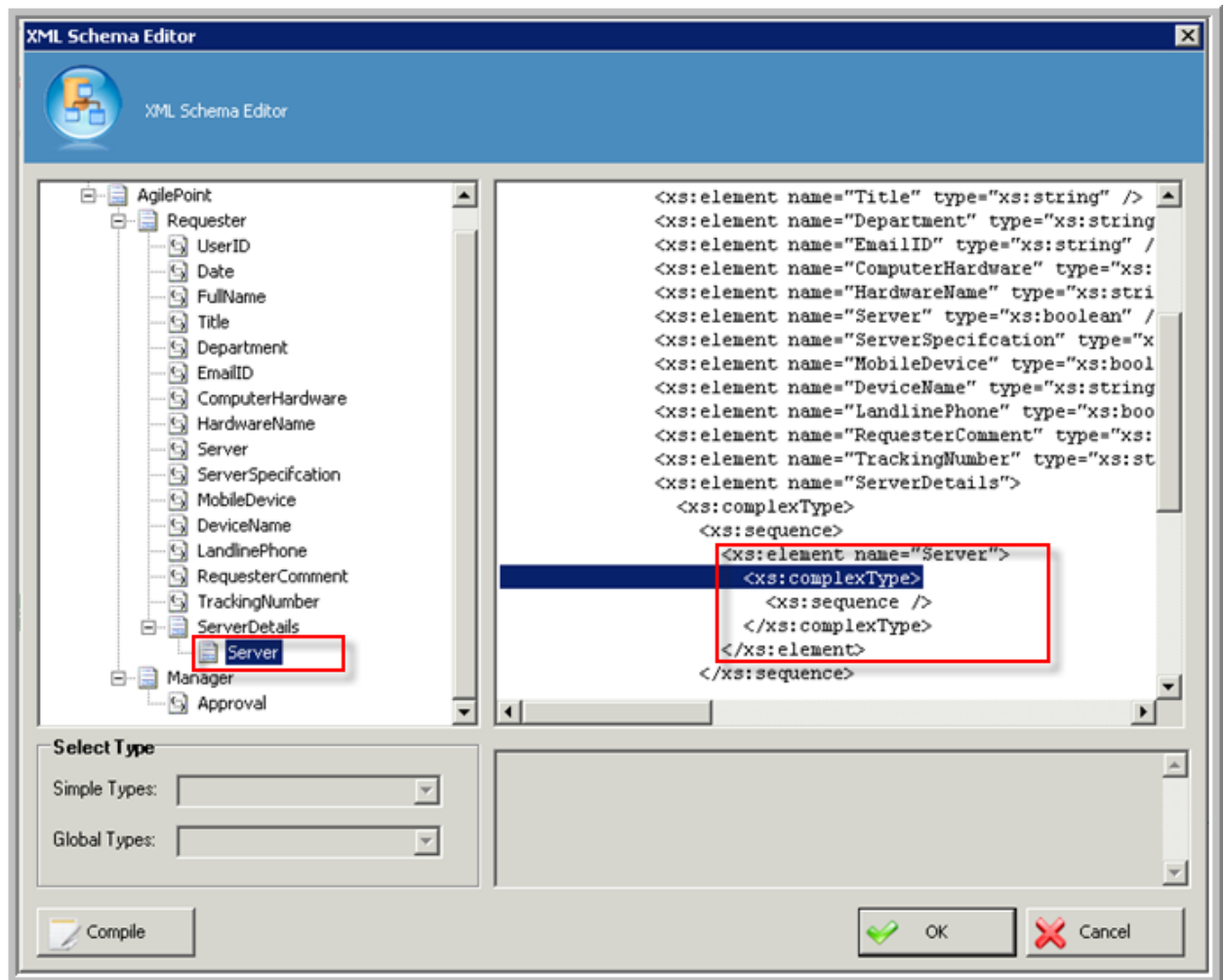
Picture: "ServerDetails" element after creation.

6. Next select the newly added "ServerDetails" node by left click to make it the active node (do not select any node by right click). Then right click on selected node and click "Complex Type" option to add a node for individual Server information. Name the newly created complex type "Server".



Picture: "Create Schema" dialog box to add "Server" node

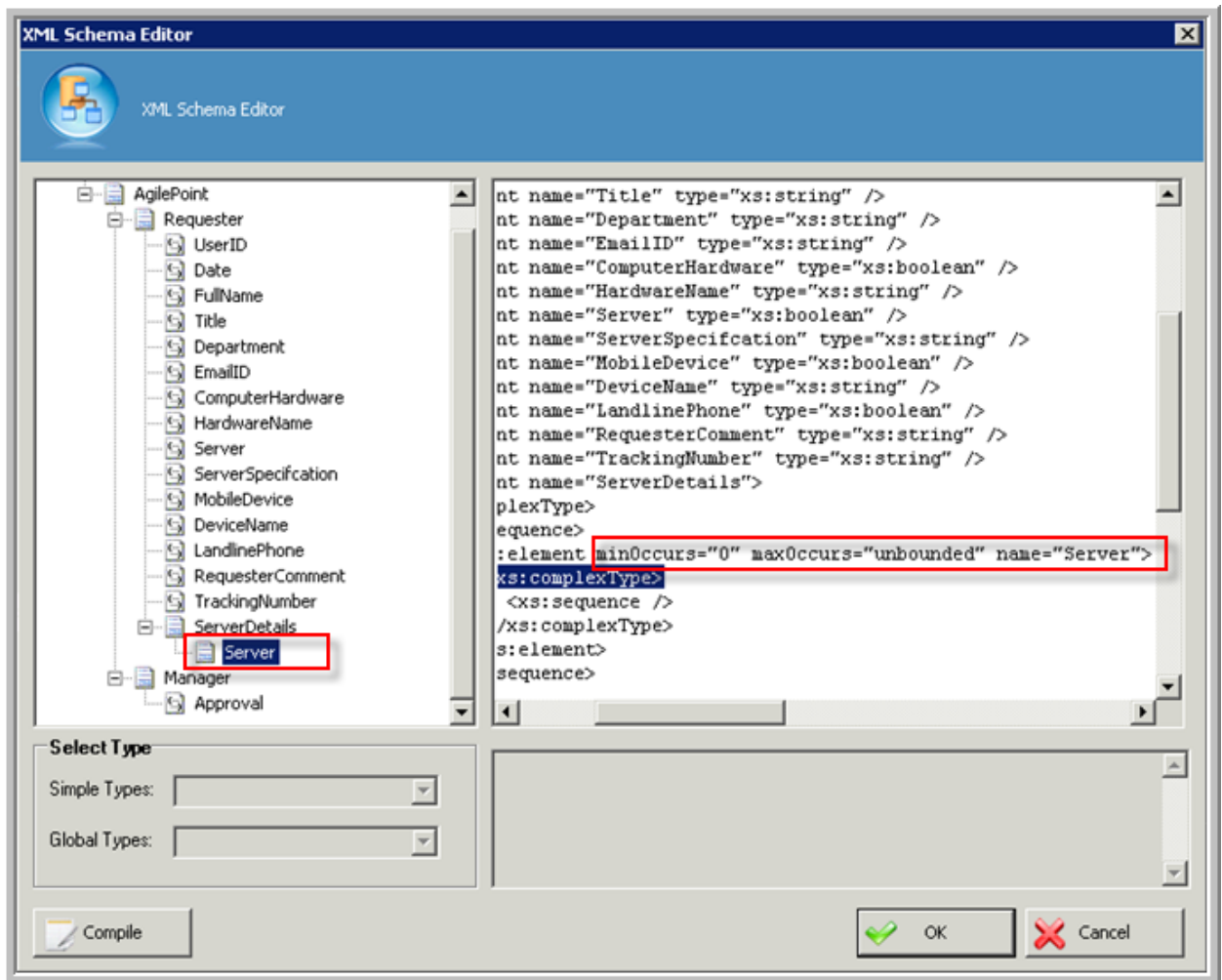
7. Once the node is added to the schema, it should appear in the schema tree structure as shown below.



Picture: "Server" element after creation.

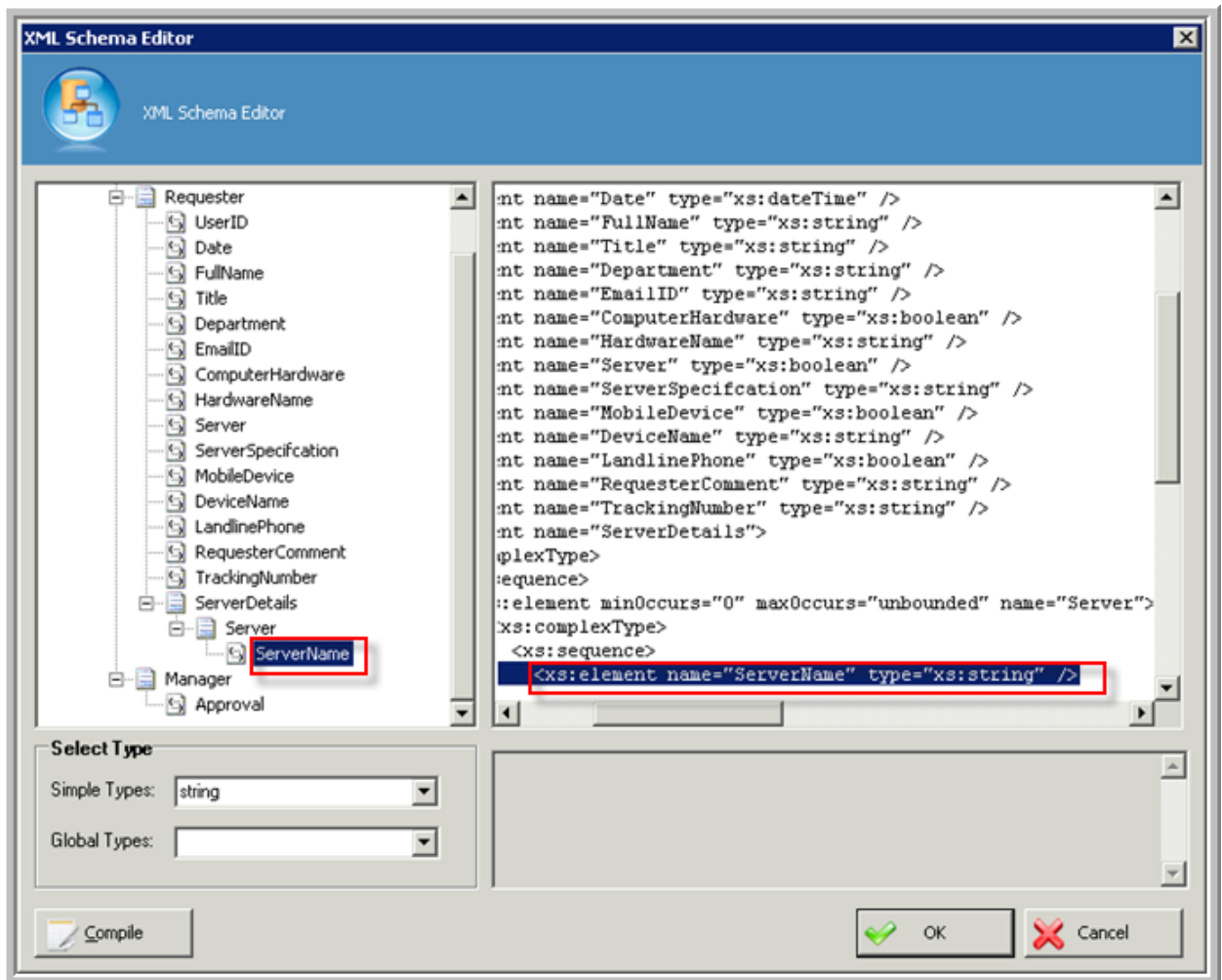
8. Make the "Server" node repeatable by adding `minOccurs` and `maxOccurs` XML properties in right hand side XML editor inside "Server" element. Have a look on below code.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Server">
```

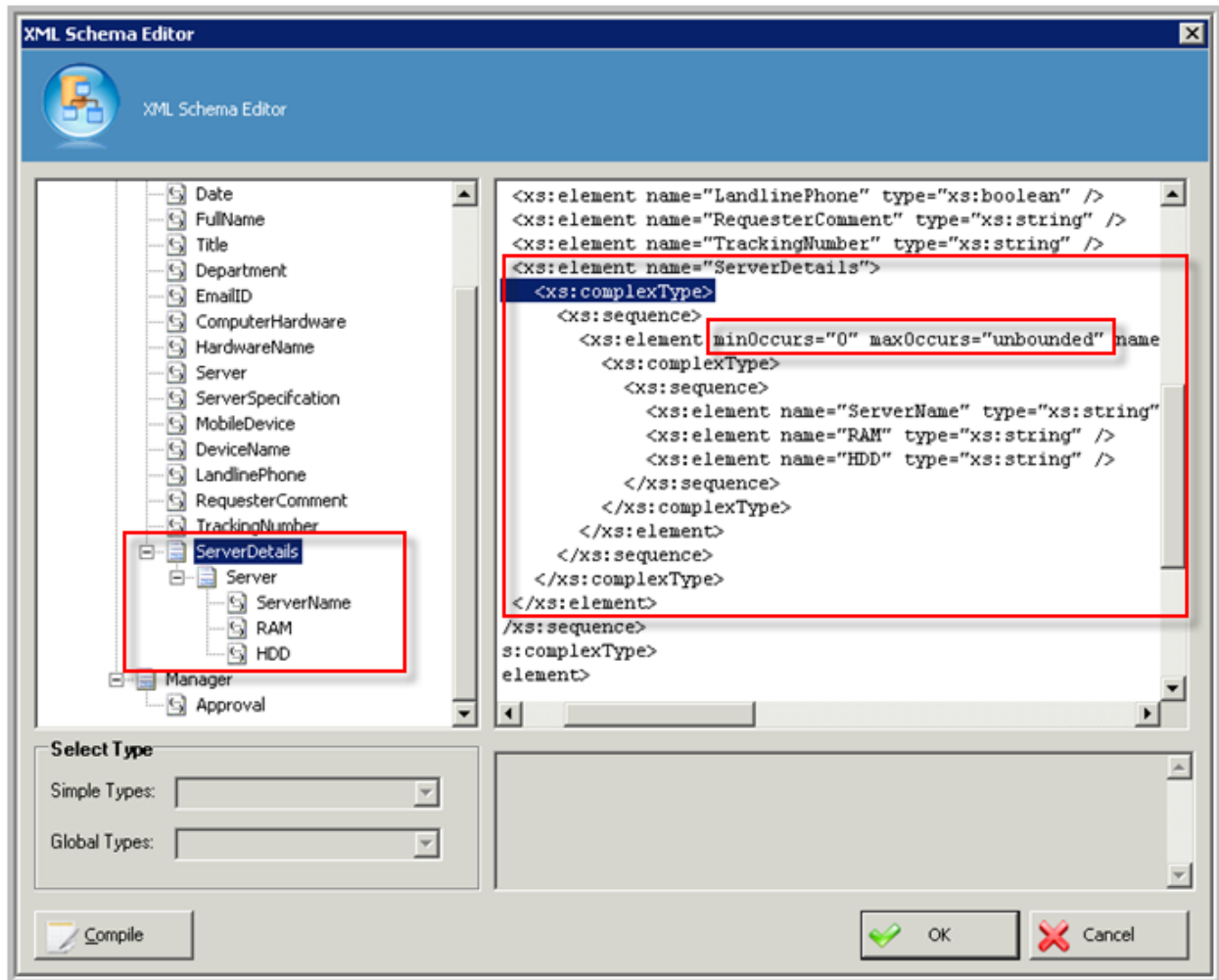
Picture: "Create Schema" dialog box to add "Server" node and make repeatable.

9. Create an element called "ServerName". The default data type of an element is string. If required, the data type can be changed by selecting appropriate data type from the "Simple Types" drop down box at the left hand side panel towards bottom of the screen.



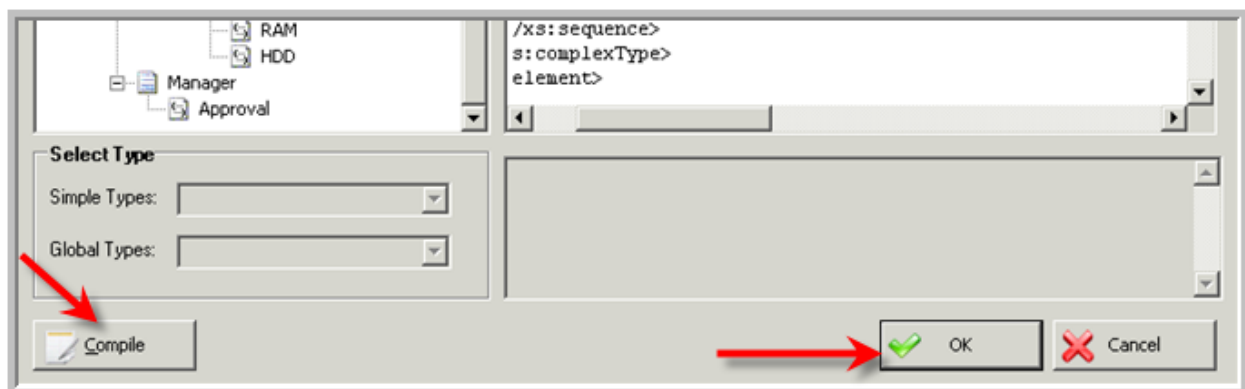
Picture: Creating "ServerName" element in AgilePoint Envision.

10. Repeat the action in previous step to create rest of the fields - RAM (string) and HDD (string) elements under "Server" category. Value in parenthesis indicates data type. Once this step is complete, your schema should show the new nodes as shown below.

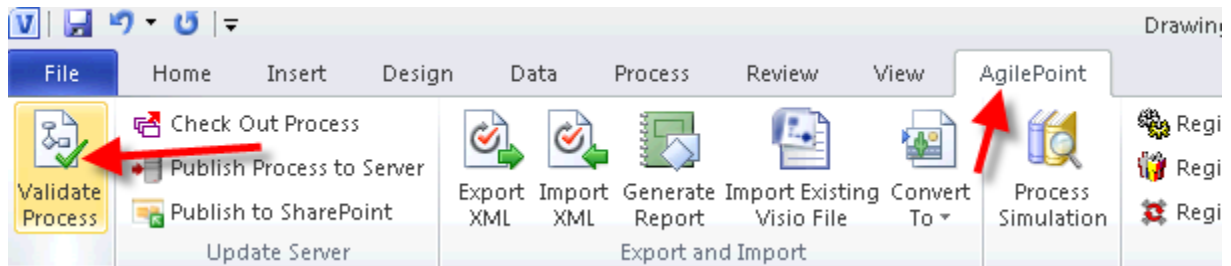


Picture: Repeatable "Server" category to store Servers.

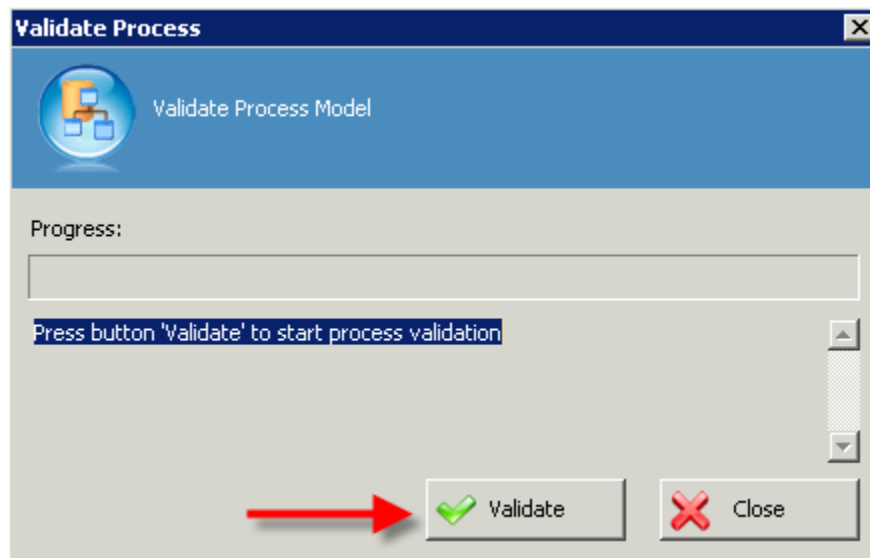
11. After schema creation please click compile button to check if it is a valid schema and then click on OK button as shown below.



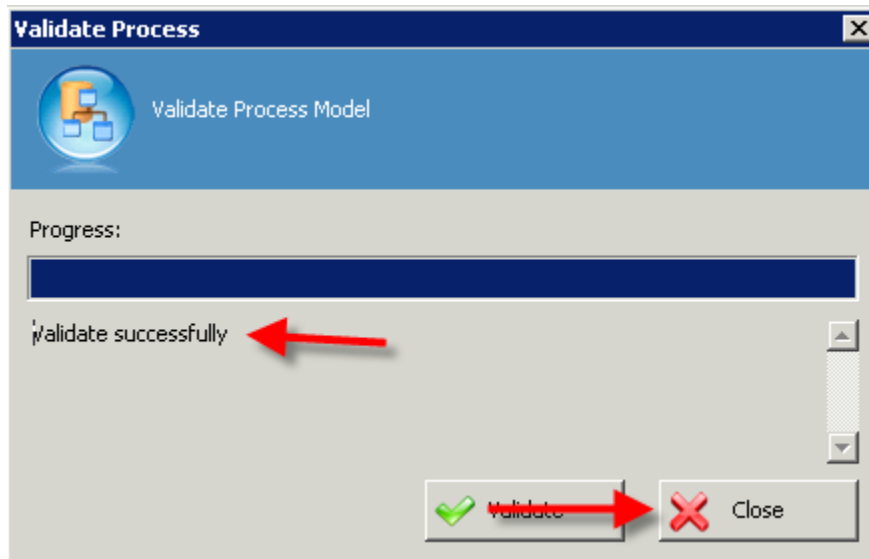
12. Now your updated process is ready to publish to AgilePoint Server. Before that click on “Validate Process” button in AgilePoint ribbon to check if there is no error in the process template and it is good to publish to server.



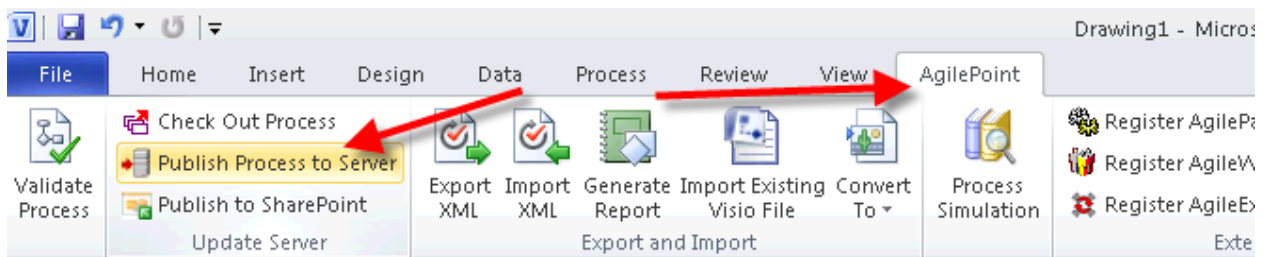
13. Click on Validate button.



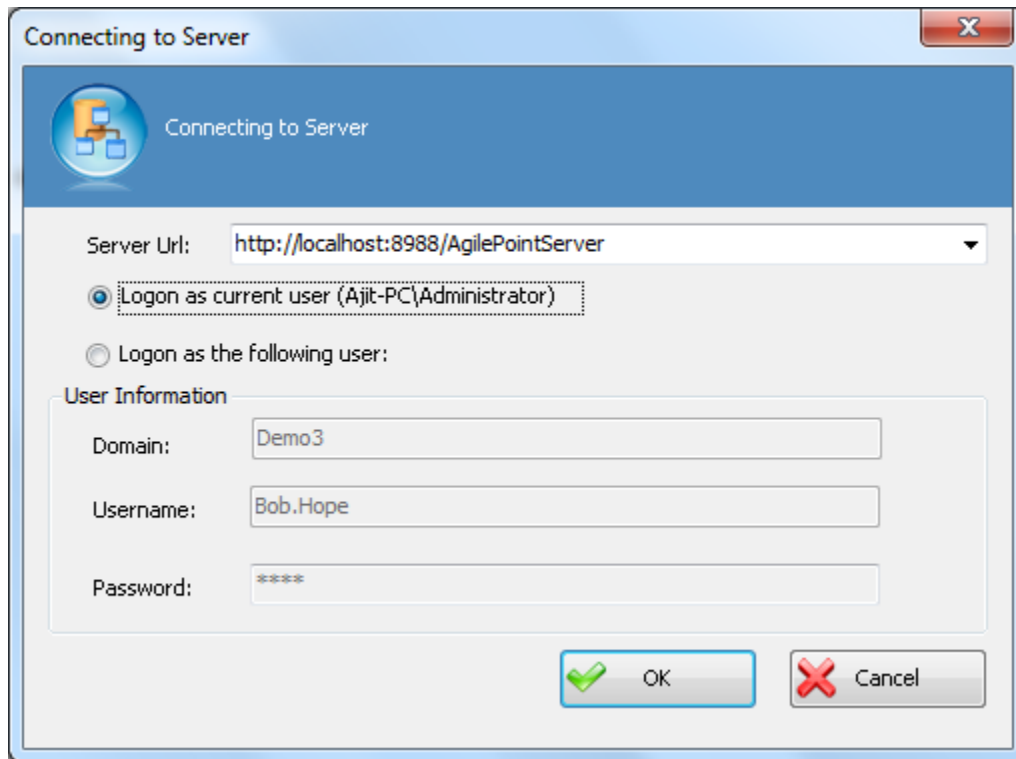
14. Once process is validated successfully click the Close button.



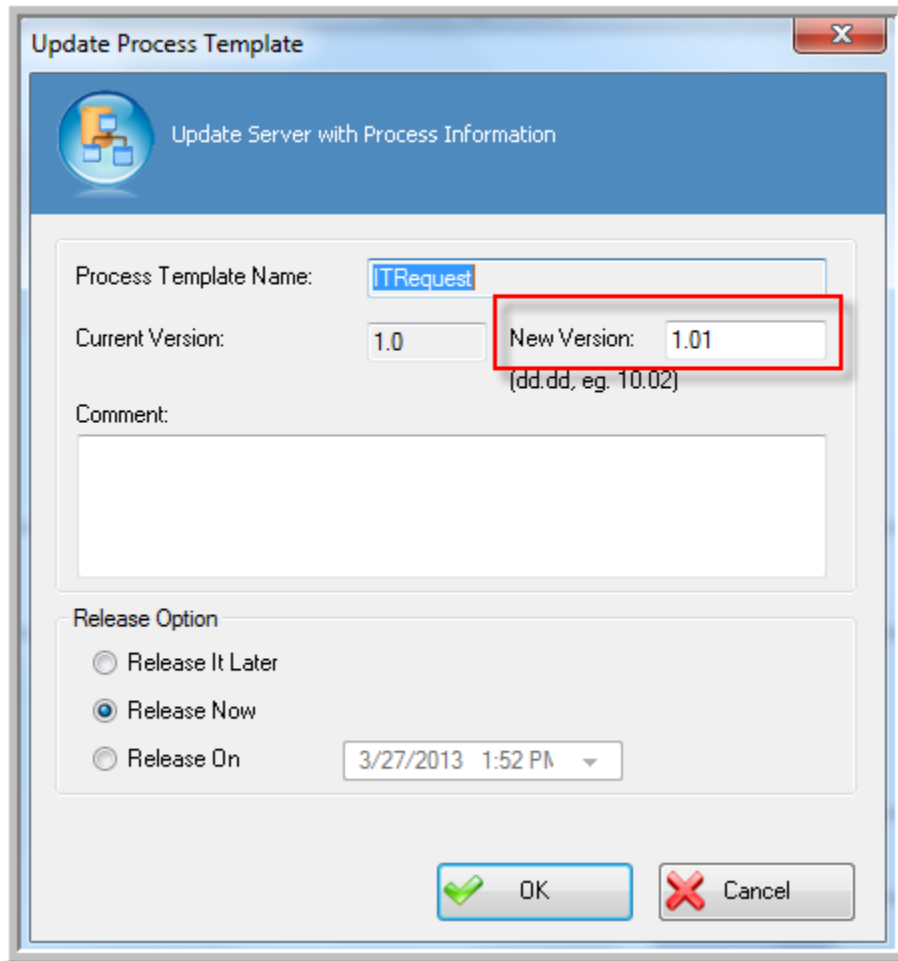
15. Click on "Publish Process to Server" button from "AgilePoint" ribbon menu.



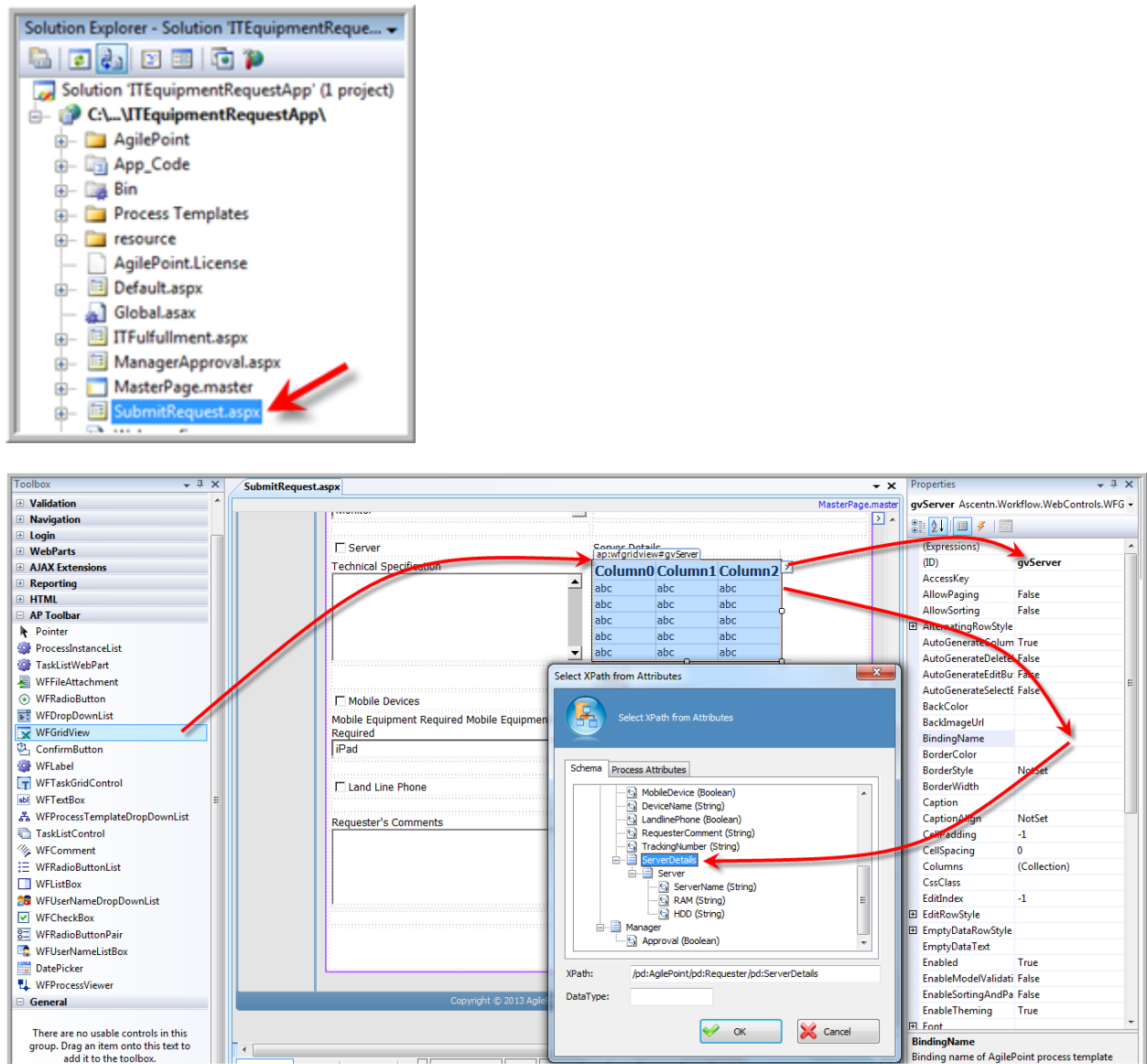
16. Click the "Publish Process to Server" button from the "AgilePoint" ribbon menu.



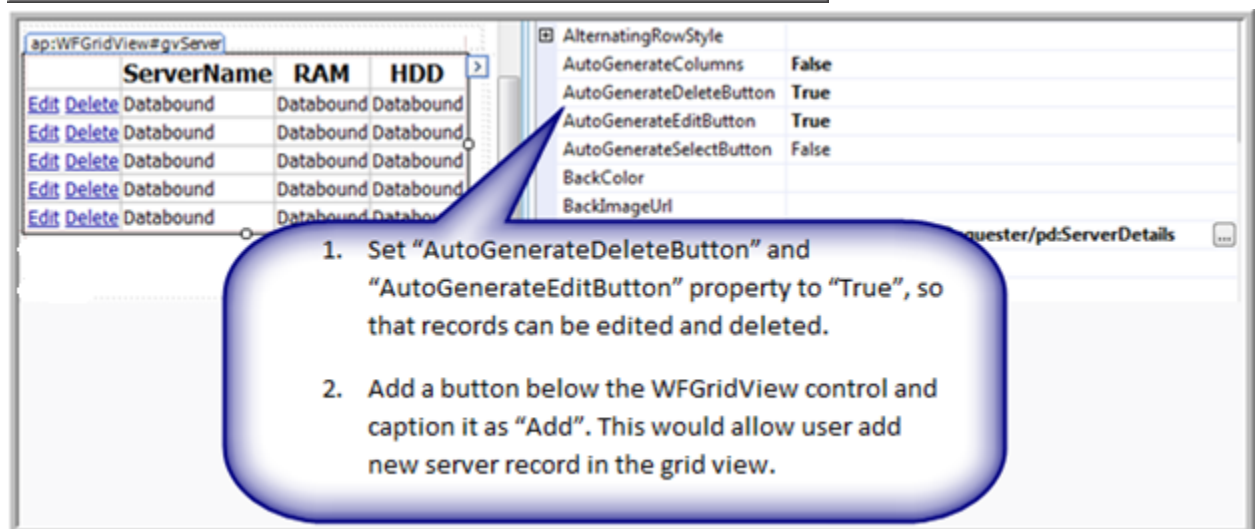
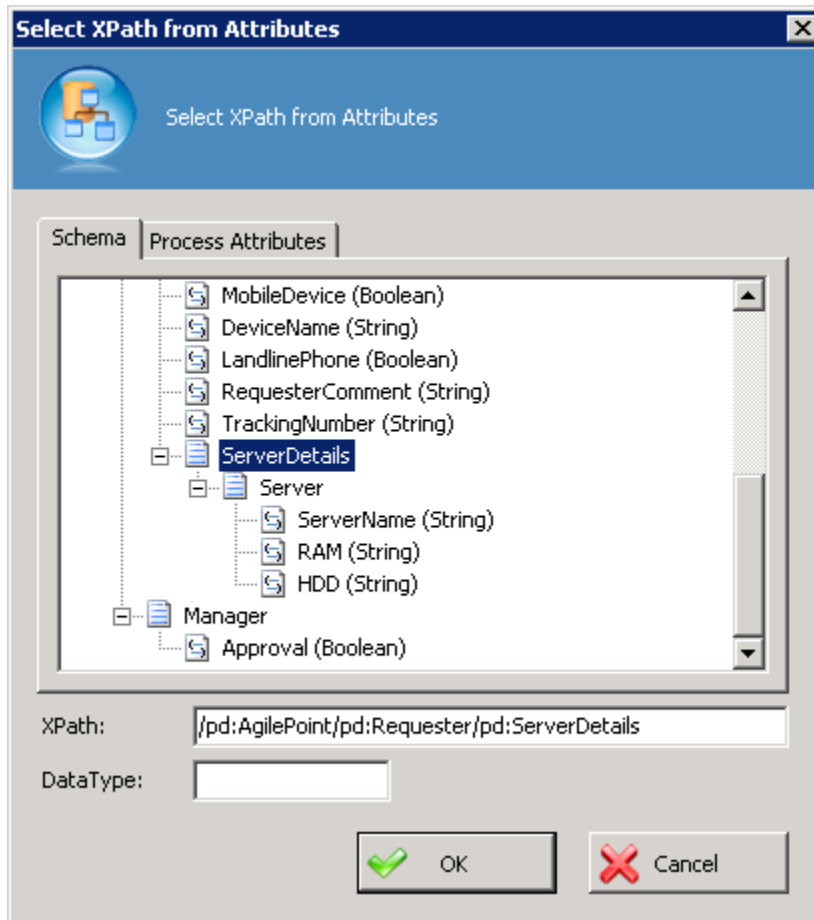
17. This will open up a dialog to provide your AgilePoint Service URL and credentials to connect to the server. Provide the information as shown below, and click OK.



18. Now save the process model in Visio and close AgilePoint Envision then go back to ASP.Net web application.
19. Open **SubmitRequest.aspx**. Drag and drop **WFGridView** control from the AgilePoint toolbox on the page design area next to the technical specification text area. Go to properties panel of the **WFGridView** control and provide ID as "**gvServer**" and then select **BindingName** property. Select the process template, "**ITRequest**" in your case when prompted to select the schema.



20. Select “**ServerDetails**” node from the schema tree. This step should generate GridView layout and html as shown in the picture below.



The auto generated HTML code would look something like this.

```
<ap:WfGridView ID="gvServer" runat="server" AutoGenerateColumns="False"
AutoGenerateDeleteButton="True" AutoGenerateEditButton="True"
BindingName="/pd:AgilePoint/pd:Requester/pd:ServerDetails"
EnableModelValidation="True"
InnerXmlTemplate="PHBk0lN1cnZlcnMgeG1sbnM6cGQ9Imh0dHA6Ly93d3cuYXNjZW50b
```

```

i5jb20vYnBtL1hNFTFNjaGVtYSI+PHBkO1NlcnZlck5hbWU+U3RyaW5nPC9wZDpTZXJ2ZXJO
YW1lPjxwZDpSQU0+U3RyaW5nPC9wZDpSQU0+PHBkOkhERD5TdHJpbmc8L3BkOkhERD48L3B
kO1NlcnZlcnM+"">
<Columns>
<asp:BoundField DataField="pd:ServerName" HeaderText="ServerName" />
<asp:BoundField DataField="pd:RAM" HeaderText="RAM" />
<asp:BoundField DataField="pd:HDD" HeaderText="HDD" />
</Columns>
</ap:WFGGridView>

```

21. Once **WFGgridview** control has been added to the page and all settings are configured then provide title of the control as “**Server Details**” on top of the **WFGgridview** control.

The screenshot shows a web form titled "Web Form for activity 'Submit'". On the left is a sidebar with links: Home, New IT Request, and My Tasks. The main form contains several input fields: User ID, Date (4/5/2013), Full Name, Title, Department (Finanace), Email Address, Computer Hardware (unchecked), Computer Hardware Required (Monitor), and Server (unchecked). Below these is a "Technical Specification" section with a scrollable area. To the right of the scrollable area is a table with the header "Server Details" (highlighted with a red box). The table has columns: Edit, Delete, ServerName, RAM, and HDD. It contains five rows of data, each with "Databound" values for the last three columns.

Server Details				
	ServerName	RAM	HDD	
Edit Delete	Databound	Databound	Databound	
Edit Delete	Databound	Databound	Databound	
Edit Delete	Databound	Databound	Databound	
Edit Delete	Databound	Databound	Databound	
Edit Delete	Databound	Databound	Databound	

22. Next add one ASP.Net button below the **WFGGridView** control for generating an empty row in the **WFGgridview** control. After adding the **WFGGridView** Control, **Title** and **ASP.Net button** on page. The page would appear as below.

Web Form for activity 'Submit'

User ID: _____ Date: 4/5/2013 ...

Full Name: _____ Title: _____

Department: Finance Email Address: _____

☐ Computer Hardware
Computer Hardware Required: Monitor

☐ Server
Technical Specification: _____

☐ Mobile Devices

Server Details

	ServerName	RAM	HDD
Edit Delete	Databound	Databound	Databound
Edit Delete	Databound	Databound	Databound
Edit Delete	Databound	Databound	Databound
Edit Delete	Databound	Databound	Databound
Edit Delete	Databound	Databound	Databound

Add

23. When user clicks the “Add” button, it should create an empty row in the WFGGridView control, so that user can add new Server record. Double-click the “Add” button to generate button click event method. Write the following code in the “Add” button click event method.

```

DataTable dt = gvServer.RepeatingSectionDataSource;
if (dt != null)
{
    DataRow child = dt.NewRow();
    child[0] = "updates";
    child["pd:ServerName"] = "";
    child["pd:RAM"] = "";
    child["pd:HDD"] = "";
    dt.Rows.Add(child);
    gvServer.RepeatingSectionDataSource = dt;
    gvServer.BindRepeatingSectionDataSource();
}

```

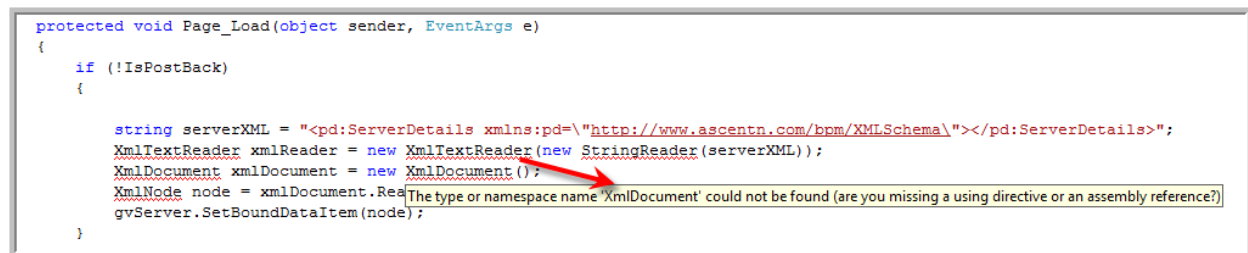
Note: The above code for WFGGridView control is NOT using DataSource property and DataBind method to bind with data source, instead it is using AgilePoint custom property “RepeatingSectionDataSource” and custom method “BindRepeatingSectionDataSource”. Making a call to this property/method would handle the entire data binding operation. Also, while creating a new row in data table, it’s loading existing data source which is bound to the control and it automatically figures out the structure for the new record.

24. Once the design part is done, switch to the code behind file "SubmitRequest.aspx.cs". The Page_Load event method has some auto generated code provided by AgilePoint component to save some coding effort. Add the code below after the existing auto generated code in Page_Load method.

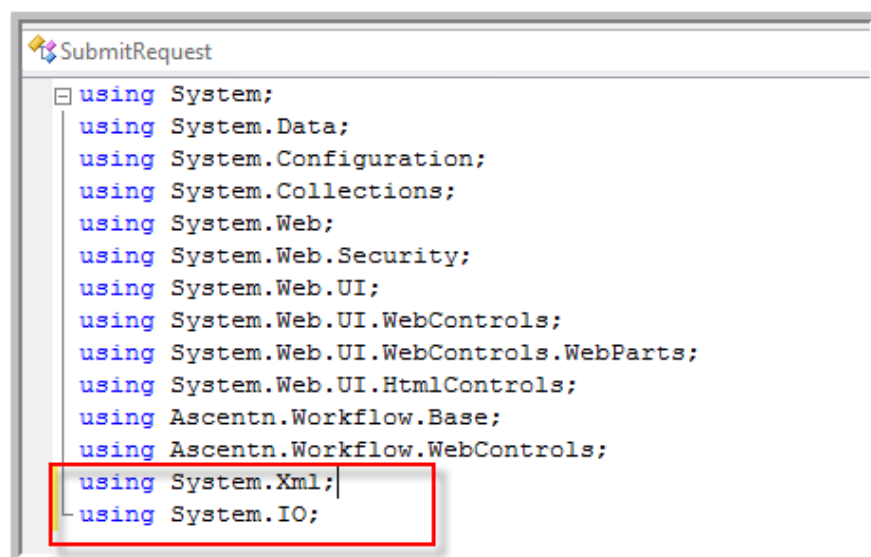
This step is needed only if the WFGGridView control is used on a web form, which starts the AgilePoint process.

```
if (!IsPostBack)
{
    string serverXML = "<pd:ServerDetails
xmlns:pd=\"http://www.ascentn.com/bpm/XMLSchema\"></pd:ServerDetail
s>";
    XmlTextReader xmlReader = new XmlTextReader(new
StringReader(serverXML));
    XmlDocument xmlDocument = new XmlDocument();
    XmlNode node = xmlDocument.ReadNode(xmlReader);
    gvServer.SetBoundDataItem(node);
}
```

If you find any error regarding above mentioned code, like shown below.



Then, please add below mentioned namespace on top the page.



```

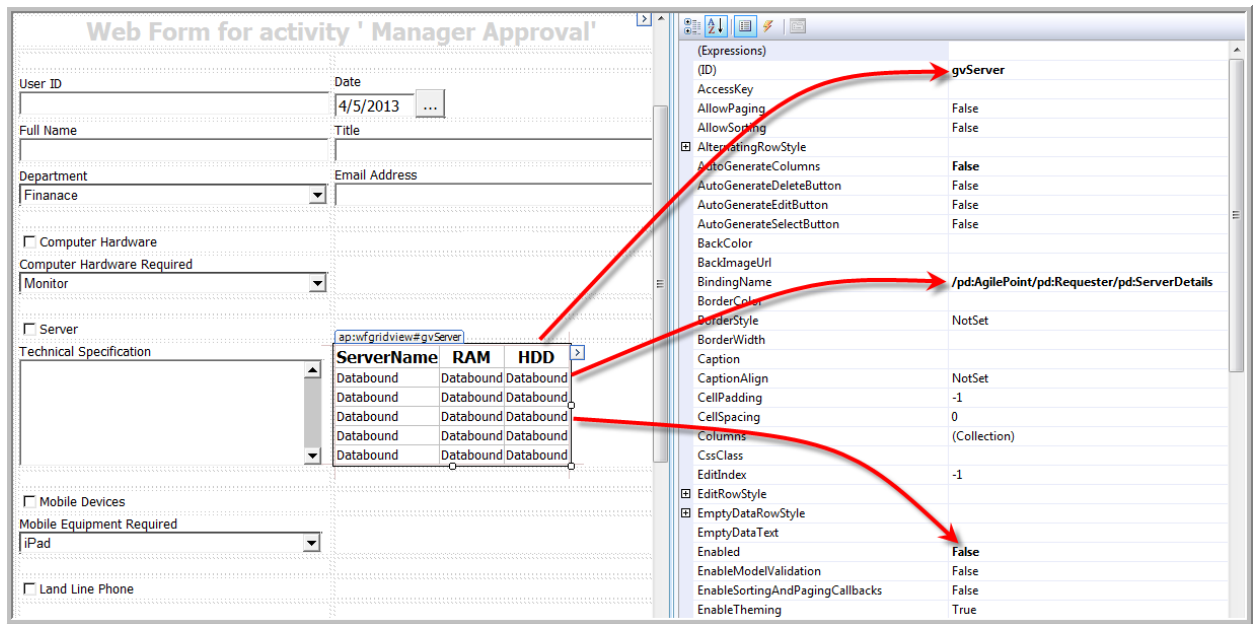
public partial class SubmitRequest : Ascentn.Workflow.WebControls.WFWorksheetPage
{
    protected void btnAdd_Click(object sender, EventArgs e)
    {
        DataTable dt = gvServer.RepeatingSectionDataSource;
        if (dt != null)
        {
            DataRow child = dt.NewRow();
            child[0] = "updates";
            child["pd:ServerName"] = "";
            child["pd:RAM"] = "";
            child["pd:HDD"] = "";
            dt.Rows.Add(child);
            gvServer.RepeatingSectionDataSource = dt;
            gvServer.BindRepeatingSectionDataSoure();
        }
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            string serverXML = "<pd:ServerDetails xmlns:pd=\"http://www.ascentn.com/bpm/XMLSchema\"></pd:ServerDetails>";
            XmlTextReader xmlReader = new XmlTextReader(new StringReader(serverXML));
            XmlDocument xmlDocument = new XmlDocument();
            XmlNode node = xmlDocument.ReadNode(xmlReader);
            gvServer.SetBoundDataItem(node);
        }
    }
}

```

Note – Above code is for “Add” button and “Page_Load” event.

25. Next open “**ManagerApproval.aspx**” page and switch to Design view.
26. Drag and drop **WFGridView** control from the AgilePoint toolbox on the page design area. Go to properties of the control and provide ID as “**gvServer**” and then select **BindingName** property. Select the process template, “**ITRequest**” in your case when prompted to select the schema. Also make **WFGridView** control enabled as “**False**”.



27. Once **WFGGridview** control has been added to the page and all settings are configured then provide title of the control as “**Server Details**” on top of the **WFGGridview** control. The page would be look like below

Web Form for activity ' Manager Approval'

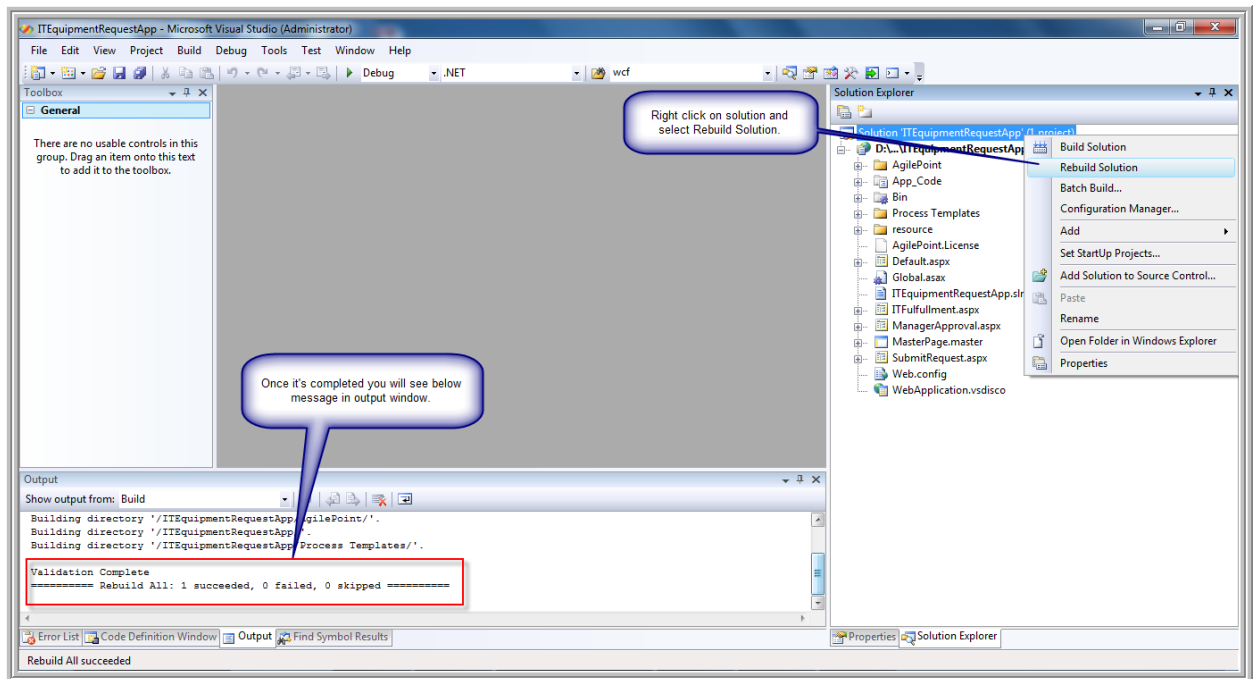
User ID	Date																		
	4/5/2013 ...																		
Full Name	Title																		
Department	Email Address																		
Finanace v																			
☐ Computer Hardware																			
Computer Hardware Required																			
Monitor v																			
☐ Server																			
Technical Specification																			
	Server Details <table border="1" style="width: 100%;"><thead><tr><th>ServerName</th><th>RAM</th><th>HDD</th></tr></thead><tbody><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr></tbody></table>	ServerName	RAM	HDD	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound
ServerName	RAM	HDD																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
☐ Mobile Devices																			
Mobile Equipment Required																			
iPad v																			
☐ Land Line Phone																			

28. Repeat the same steps to add read-only WFGridView to “ITFulfillment.aspx” page.

Web Form for activity 'IT Fulfillment'

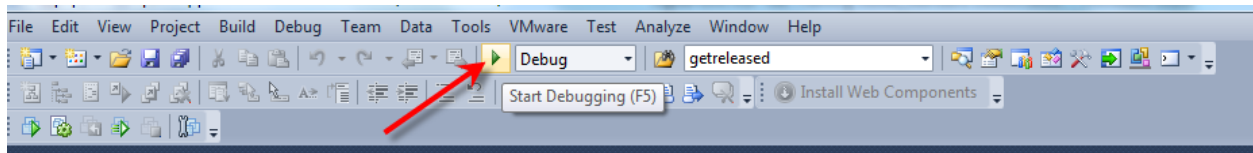
User ID	Date																		
	4/5/2013 ...																		
Full Name	Title																		
Department	Email Address																		
Finanace																			
☐ Computer Hardware																			
Computer Hardware Required																			
Monitor																			
☐ Server																			
Technical Specification																			
Server Details <table border="1" style="width: 100%;"><thead><tr><th>ServerName</th><th>RAM</th><th>HDD</th></tr></thead><tbody><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr><tr><td>Databound</td><td>Databound</td><td>Databound</td></tr></tbody></table>		ServerName	RAM	HDD	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound	Databound
ServerName	RAM	HDD																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
Databound	Databound	Databound																	
☐ Mobile Devices																			
Mobile Equipment Required																			
iPad																			
☐ Land Line Phone																			

29. Build and run the web application. (*Shown below.*)



Running Re-Build application

1. Click the green play button in the Visual Studio to start your application.



2. Provide your credentials to log in to the application and click on Logon Button

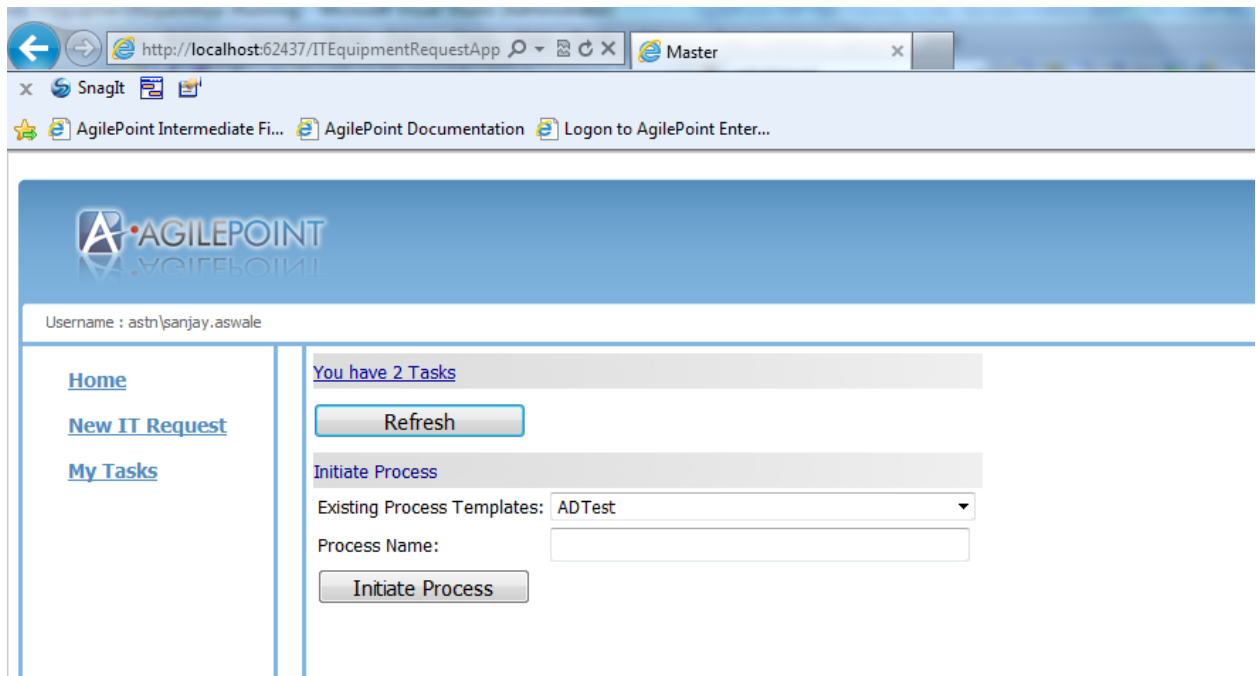
Domain :

User Name :

Password :

Copyright © 2013 AgilePoint Inc. All Rights Reserved

3. After log in you will see a default page and an inbuilt process initiator functionality which we are not going to use it. So you can modify the default.aspx page to change the layout of this page.



4. Click on "New IT Request" to create new request. Provide the information and click submit.

Web Form for activity 'Submit'

User ID
Ajit007

Full Name
Ajit Kumar Singh

Department
Finanace

☒ Computer Hardware
Computer Hardware Required
Monitor

☒ Server
Technical Specification
Server Configuration

☒ Mobile Devices
Mobile Equipment Required Mobile Equipment Required
iPad

☒ Land Line Phone
Requester's Comments
Please have a look.

Date
04/05/2013 ...

Title
Mr

Email Address
ajit.kumar.singh@agilepoint.com

Server Details

	ServerName	RAM	HDD
Edit Delete	Server 1	2 GB	250 GB
Edit Delete	Server 2	4 GB	500 GB
Edit Delete	Server 3	6 GB	750 GB

[Add](#)

Add Servers by clicking on Add button.

[Submit](#)

NOTE: Instruction for adding rows in WFGridView control. Please see below.

Server Details

1. click on "Add" button it will add blank row in WFGGridview control.

	ServerName	RAM	HDD
Edit Delete			

[Add](#)

2. click on Edit button it will take you to modify blank row in WFGGridview control.

3. Please enter server information and then click "Update" button.

Server Details

	ServerName	RAM	HDD
Update Cancel	Server 1	2 GB	250 GB

[Add](#)

4. After update button click, you can see server information has been added to the WFGGridview control.

Server Details

	ServerName	RAM	HDD
Edit Delete	Server 1	2 GB	250 GB

[Add](#)

5. After request submission, it will take you to your task list page where you would see a new task for Manager Approval.

My Tasks

Global actions

Drag a column header here to group by that column

	Priority	Template Name	Process Name	Task	Status	Due Date
	📌	ITRequest	ITRequest-E969F040FAEB44C4B32F07AAAAA7AFDA	ManagerApproval	Assigned	4/5/2013 7:51:00 PM

1 of 1 Prev Next

6. Click on the task link to show up available actions you can perform on the task.

My Tasks

Global actions

Drag a column header here to group by that column

	Priority	Template Name	Process Name	Task	Status	Due Date
		ITRequest	ITRequest-E969F040FAEB44C4B32F07AAAAA7AFDA	ManagerApproval	Assigned	4/5/2013 7:51:00 PM

1 of 1 Prev Next

- View Process
- Open ManagerApproval
- Cancel Task
- Reassign Task
- Cancel Process
- Task Rework
- Create Linked Work Item
- View Description

- Next we will complete the manager approval task. Since in the initial process design we selected same user to be the participant of all 3 manual steps, you do not need to log off. This can be changed to unique users or roles after you execute simple demo successfully.
- Click on **“Open ManagerApproval”** to open the task. You can see all information entered while submitting new request is automatically available in approval screen. This is because of the Binding Name property on each control that ties the data between process and UI.

Web Form for activity ' Manager Approval'

Computer Hardware Required

User ID **Date** ...

Full Name **Title**

Department **Email Address**

☒ Computer Hardware

☒ Server

Technical Specification

Server Details

ServerName	RAM	HDD
Server 1	2 GB	250 GB
Server 2	4 GB	500 GB
Server 3	6 GB	750 GB

☒ Mobile Devices

Mobile Equipment Required

☒ Land Line Phone

Requester's Comments

☒ Approve ☐ Reject

9. Select the Approve radio button, and click on Submit.
10. Navigate back to task list, open the ITFulfillment task, and from ITFulfillment page click the Complete button to finish the process.

AgilePoint Task List

Global actions

Drag a column header here to group by that column

Priority	Template Name	Process Name	Task	Status	Due Date
	ITRequest	ITRequest-9D71349E3A4D4EB59C09F97F67EE3D6A	ITFulfillment	Assigned	4/6/2013 11:25:00 AM

Prev Next

- View Process
- Open ITFulfillment
- Cancel Task
- Reassign Task
- Cancel Process
- Task Rework
- Create Linked Work Item
- View Description

Copyright © 2013 AgilePoint, Inc.

Web Form for activity 'IT Fulfillment'

Computer Hardware Required

User ID

Date

Full Name

Title

Department

Email Address

☒ Computer Hardware

☒ Server

Technical Specification

Server Details

ServerName	RAM	HDD
Server 1	2 GB	250 GB
Server 2	4 GB	500 GB
Server 3	6 GB	750 GB

☒ Mobile Devices

Mobile Equipment Required

☒ Land Line Phone

Requester's Comments

Complete

Summary

ASP.NET development with AgilePoint process is similar to the traditional web development except lot less coding efforts. AgilePoint Envision has provided “Generic” process template for ASP.Net and any other third party form technology. The Generic process template has schema to be defined, which can be used seamlessly for integration with ASP.Net web site using BindingName property on AgilePoint controls. This minimizes lots of plumbing required to make process workflow work. Also, concept of sub process makes cross form technology work together and provides the reusability of process across the server/farm.