

A SharePoint Developer Introduction

Hands-On Lab

Lab Manual

SPCHOL302 – Using LINQ-to-SharePoint – C#

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft. All rights reserved.

**Contents**

[SPCHOL302 – Using Linq-to-SharePoint 2](#_Toc259797292)

[Lab Objective 2](#_Toc259797293)

[Additional Resources 2](#_Toc259797294)

[Getting Started 2](#_Toc259797295)

[Logging in to the Virtual Machine 2](#_Toc259797296)

[Locations 2](#_Toc259797297)

[Lab Pre-requisites 3](#_Toc259797298)

[Copying code samples from Word document 4](#_Toc259797299)

[Code Snippets 4](#_Toc259797300)

[Exercise 1 – Extending the Visual Web Part using Linq-to-SharePoint 5](#_Toc259797301)

[Task 1 – Using Linq-to-SharePoint 5](#_Toc259797302)

[Task 2 – Generate Linq-to-SharePoint proxy class to access list data 8](#_Toc259797303)

[Task 3 – Linq Queries 14](#_Toc259797304)

[Exercise 2 – Programming with Content Types 18](#_Toc259797305)

[Task 1 – Create a new SharePoint Project 18](#_Toc259797306)

[Task 2 – Create a new Content Type 21](#_Toc259797307)

[Task 3 – Verify the new Content Type 25](#_Toc259797308)

[Task 4 – Create a new List based on the new Content Type 28](#_Toc259797309)

[Task 5 – Disassociate a List from a Content Type 33](#_Toc259797310)

[Task 6 – Delete the List 34](#_Toc259797311)

[Lab Summary 35](#_Toc259797312)

# SPCHOL302 – Using LINQ-to-SharePoint

Estimated time to complete this lab: **30 minutes**

Visual Studio 2010 and SharePoint Foundation 2010 are required for these exercises. These are installed on the Virtual Machine used in this lab.

## Lab Objective

Linq is a new data access paradigm that allows users to express SQL like syntax against a variety of data sources. Linq can improve performance by allowing the back-end data source to decide the best way to solve the query. SharePoint now fully supports Linq for querying lists, so you can query information from the platform in a more condensed, easier to understand format.

The objective of this lab is to learn about how to use the new Linq-to-SharePoint data access paradigm:

* Create a basic SharePoint Visual Web Part that displays information from within the SharePoint site.
* Create a new Content Type using the SharePoint Object Model

## Additional Resources

This lab includes the following additional resources:

|  |  |  |
| --- | --- | --- |
| This Lab Manual | SPCHOL302\_Manual\_CS.docx | This document. |
| Source Code | Completed\CS\Ex1  Completed\CS\Ex2 | Completed lab source code in C#. |
| Resources | *Resources\CS* | Various other resources used throughout this lab. |

## Getting Started

### Logging in to the Virtual Machine

Please log into the virtual machine as the following user:

**Username**: Administrator

**Password**: pass@word1

### Locations

This Hands-On Lab contains a number of additional resources in fixed locations. By default, it is assumed that the base HOL directory is **C:\Content Packs\Packs\SharePoint 2010 Developer Labs 1.0\SUPPORTING FILES\*SPCHOL302\Resources****.*

The default working folder for this lab is C**:\*SPHOLS\SPCHOL302***

### Lab Pre-requisites

1. Browse to base HOL directory ***Supporting Files\SPCHOL302\Resources***and execute the **optimize.ps1** PowerShell script:
   * Right-click on **optimize.ps1** and select **Run with PowerShell**:

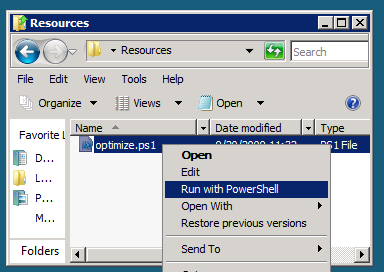


Figure 1 - Execute the PowerShell script

* + This will open a PowerShell window to execute the script. Please wait until the PowerShell script completes executing the script and closes the PowerShell window (this may take a while):

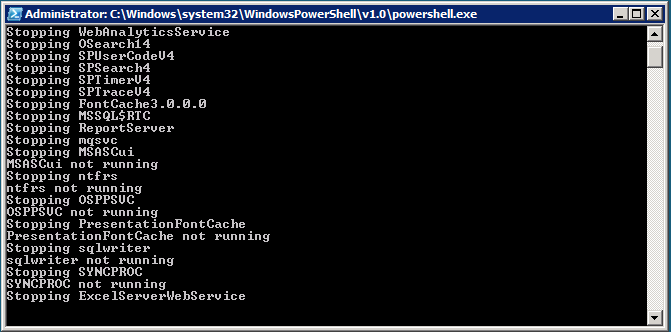


Figure 2 - PowerShell Window executing the script

1. Browse to base HOL directory ***Supporting Files\SPCHOL302\Resources***and execute the **add\_bugs.ps1** PowerShell script:
   * Right-click on **add\_bugs.ps1** and select **Run with PowerShell:**
   * This will open a PowerShell window to execute the script. Please wait until the PowerShell script completes executing the script and closes the PowerShell window.

### Copying code samples from Word document

Copying and pasting code from this Word document to Visual Studio is only safe for the sections of formatted code, e.g.:

Console.WriteLine("This is safe code!");

Code not in these sections may contain Unicode or other invisible characters that are not valid XML or C#/VB code, e.g.:

**Console.WriteLine(“This is NOT safe code !!”);**

### Code Snippets

You can also use Code Snippets to insert the appropriate code in the lab.

To use the required code snippet for this lab:

* Right-click on the code file where you want to insert the code snippet.
* Select **Insert Snippet:**

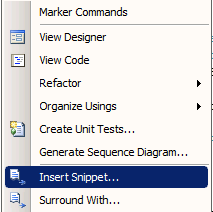
****

Figure 3 - Insert Code Snippet

* Select the required code snippet(s) from **My Code Snippets** gallery.

## Exercise 1 – Extending the Visual Web Part using Linq-to-SharePoint

In this exercise, in a Visual Web Part, you will write Linq queries across 3 lists – Projects, Employees, and Bugs..This Web Part will:

* Display the employees in a Visual Web Part according to the Linq queries performed across 3 different lists – Projects, Employees, and Bugs.

### Task 1 – Using Linq-to-SharePoint

In this task you will open an existing Visual Studio SharePoint project that consists of a Visual Web Part.

1. Open **Visual Studio 2010** by going to **the Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | Open | Project/Solution..**.
3. In the **Open Project** dialog, browse to the ***SPHOLS\SPCHOL302\CS\Ex1*** directory and open the file **Begin.sln**.

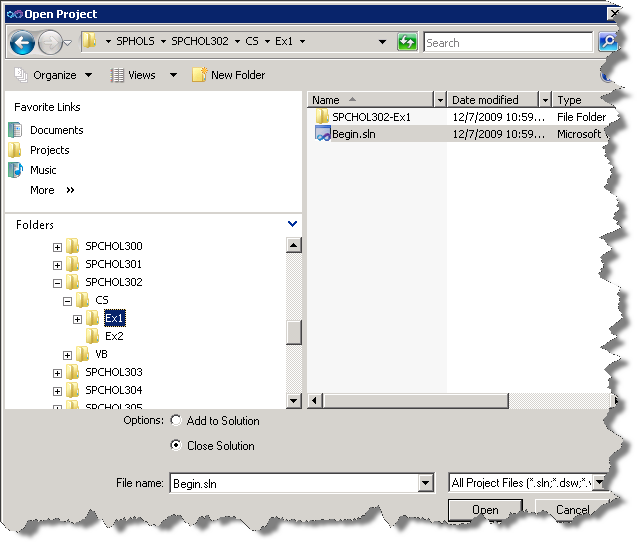


Figure 4 - Open Begin Solution

1. Select **Begin.sln**.
2. Click **Open**.
3. A Visual C# VisualWebPart project SPCHOL302-Ex1 can now be seen in the Solution Explorer.**.**

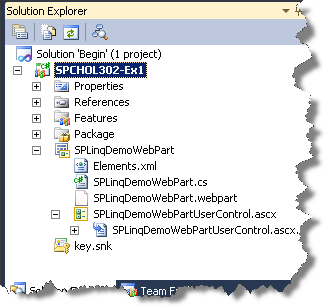


Figure 5 – SPCHOL302-Ex1 Project

1. In the Solution Explorer double-click on **SPLinqDemoWebPart**.
2. The Visual Web Part User Control **SPLinqDemoWebPartUserControl.ascx** contains an **SPGridView** and an ASP.NET **DropDownList**.

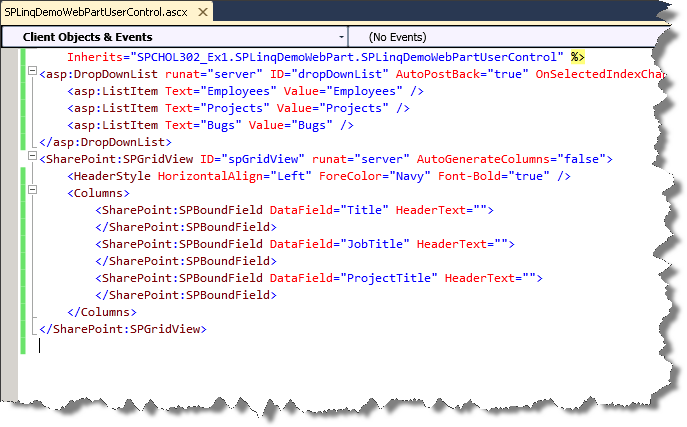


Figure 6 - Visual WebPart User Control

1. In the Solution Explorer, right-click on References and select **Add Reference.**
2. Switch to **Browse** tab and in the File Name textbox enter **C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\ISAPI**. Press **Enter**.
3. Select **Microsoft.SharePoint.Linq.dll**, and press **OK.**

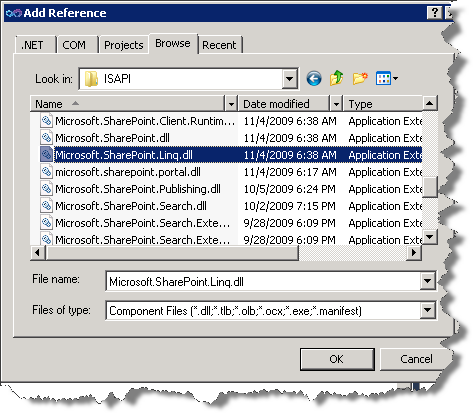


Figure 7 - Add Reference

### Task 2 – Generate Linq-to-SharePoint proxy class to access list data

In this task, you will use the new **spmetal.exe** to generate the necessary Linq-to-SharePoint proxy code.

1. In the Solution Explorer, right-click on **SPCHOL302-Ex1** and select **Open Folder in Windows Explorer.**
2. Hold **Shift key, right-click** anywhere in the **Explorer Window,** and select **Open Command Window Here** to open the command prompt window in the current project directory.

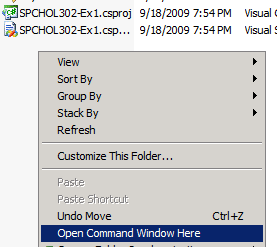


Figure 8 - Open Command Window Here

1. Type the following command in the command prompt and press **Enter** key to set the path to the SharePoint 2010 folder:

**set path=%path%;c:\program files\common files\microsoft shared\web server extensions\14\bin**

1. Type the following command in the command prompt and press **Enter** key to generate the Linq proxy code:

**spmetal.exe /web:http://intranet.contoso.com /namespace:SPCHOL302\_Ex1.SPLinqDemoWebPart /code:SPLinq.cs**

1. Switch to Visual Studio.
2. Right-click on **SPCHOL302-Ex1** and select **Add | Existing Item.**

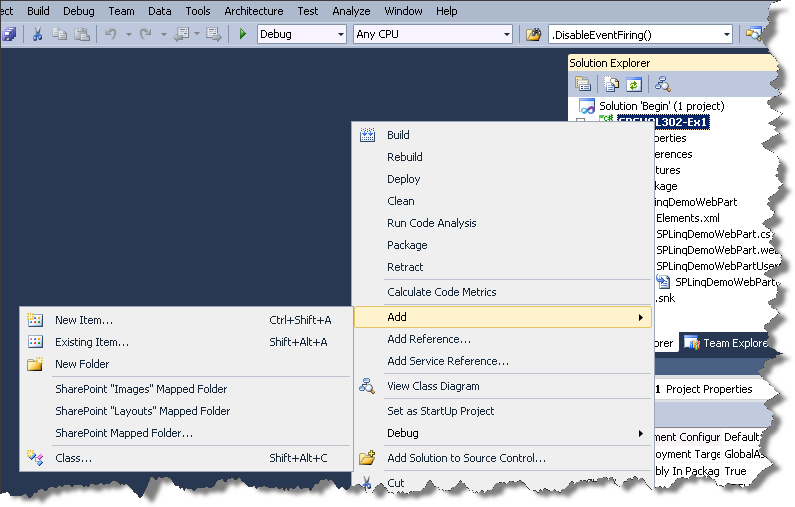


Figure 9 - Add Existing Item

1. Select **SPLinq.cs** from the **Add Existing Item** dialog window and click **Add.**

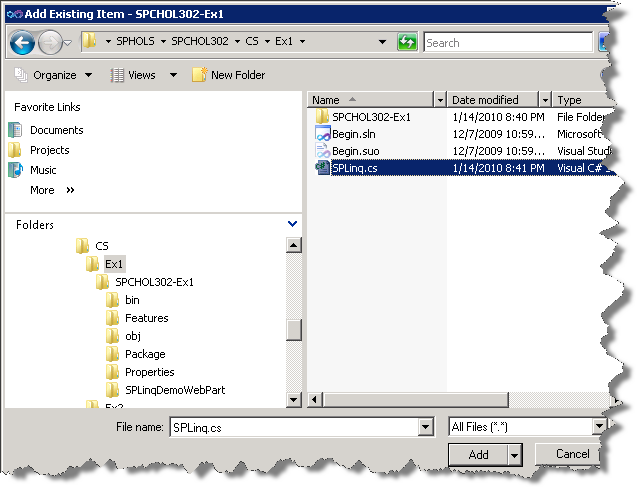


Figure 10 - Adding SPLinq.cs

1. Expand **SPLinqDemoWebPart** in the Solution Explorer.
2. Right-click on **SPLinqDemoWebPartUserControl.ascx** and select **View Code.**

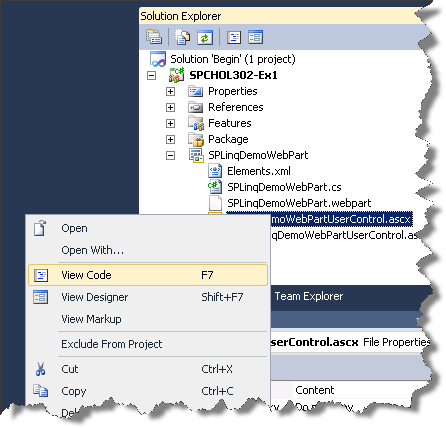


Figure 11 - View Code

1. Add the following **using statements**:

using Microsoft.SharePoint.Linq;

using Microsoft.SharePoint;

using System.Linq;

using System.Collections.Generic;

**Code Snippet**: *My Code Snippets | spchol302\_ex1\_namespaces*

1. Declare the following variables below the **SPLinqDemoWebPartUserControl** class:

EntityList<ProjectsItem> Projects;

EntityList<EmployeesItem> Employees;

EntityList<BugItem> Bugs;

**Code Snippet:** *My Code Snippets | spchol302\_ex1\_variables*

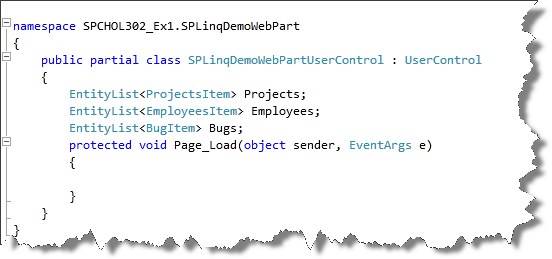


Figure 12 - VisualWebPart1UserControl class

1. Insert the following code in the **Page\_Load** method:

string spWebUrl = SPContext.Current.Web.Url;

SPLinqDataContext dc = new SPLinqDataContext(spWebUrl);

Projects = dc.GetList<ProjectsItem>("Projects");

Employees = dc.GetList<EmployeesItem>("Employees");

Bugs = dc.GetList<BugItem>("Bugs");

dc.Dispose();

**Code Snippet:** *My Code Snippets | spchol302\_ex1\_pageload*

### Task 3 – Linq Queries

In this task, you will write Linq queries and query across 3 SharePoint lists – Projects, Employees, and Bugs.

1. Add the Linq Query methods below the **Page\_Load** method using the **spchol302\_ex1\_queries** code snippet:

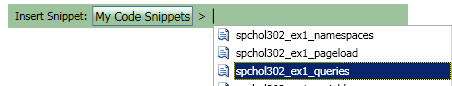


Figure 13 - Insert Linq Query Methods

This will insert 4 methods:

* ChangeQuery
* GetProjects
* GetEmployees
* GetBugs

By changing the value of the DropDown, you will change the query that is run, and then the data in the GridView will change.

1. In **Solution Explorer**, right-click the **SPCHOL302-Ex1** project and select **Deploy**. This will deploy the web part to the <http://intranet.contoso.com> SharePoint site.

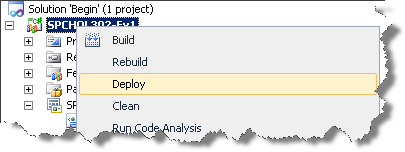


Figure 14 - Deploy SPCHOL302

1. Open Internet Explorer and browse to the following site:

<http://intranet.contoso.com>

1. If prompted for authentication, enter the following details.

**Username**: Administrator

**Password**: pass@word1

1. Click the **Edit** icon in the ribbon.

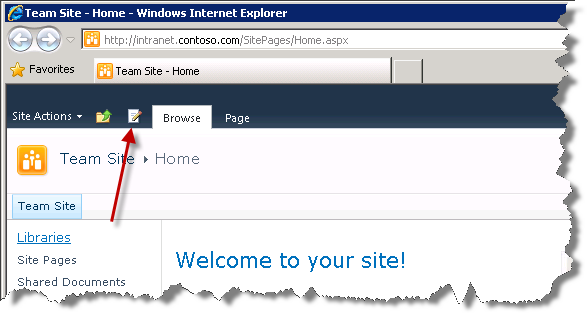


Figure 15 - SharePoint Ribbon - Edit Icon

1. Click the **Insert** tab under **Editing Tools** in the Ribbon

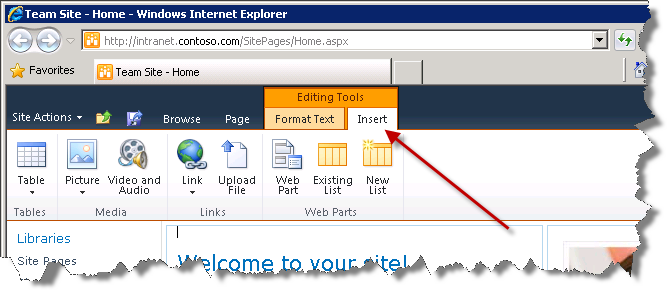


Figure 16 - SharePoint Ribbon - Insert WebPart

1. Click the **Web Part** button. Then, under **Categories,** select **Custom. SPLinqDemoWebPart**should appear in the list of Web Parts. In the page, place your cursor below the Welcome Web Part (and above the Shared Documents area), then select **Add** on the right-hand side of the page

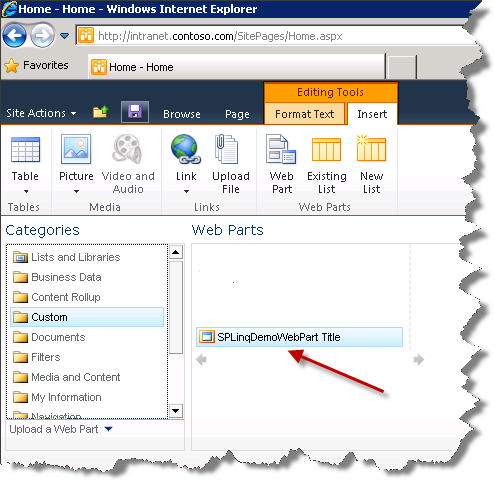


Figure 17 - Selecting *SPLinqDemoWebPart Title* Web Part

1. Click on the **Save & Close** button in the Ribbon and select **Stop Editing**. Select **Yes** if the **Save Changes** dialog appears.

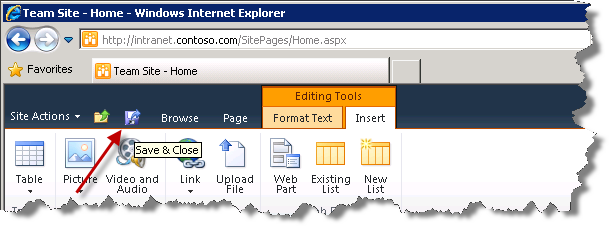


Figure 18 - SharePoint Ribbon - Stop Editing

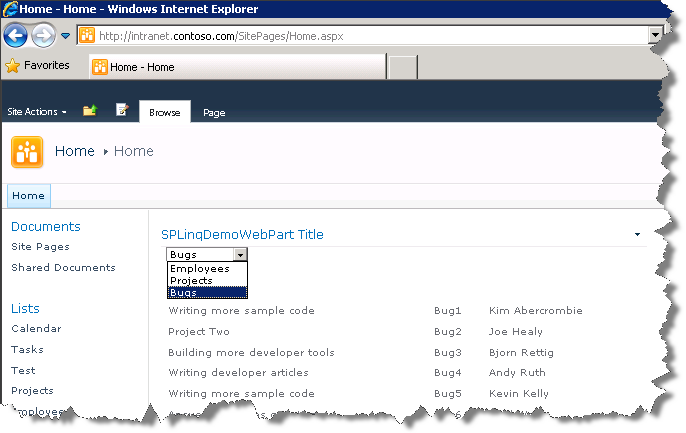


Figure 19 - Web Part in page

In the past few minutes you saw how to use the new Linq-to-SharePoint code generation utility **spmetal.exe** to generate Linq class mappings for SharePoint and write Linq queries across 3 lists – **Projects**, **Employees,** and **Bugs**.

## Exercise 2 – Programming with Content Types

In this exercise, you will create a new Content Type and add two fields to the new content type:

* A new Text field
* An existing field

You will also create new SharePoint List and associate the list with the new content type.

### Task 1 – Create a new SharePoint Project

In this task you will use the SharePoint object model to create the new content type and deploy the solution to the local SharePoint site <http://intranet.contoso.com>.

1. Open **Visual Studio 2010** by going to **the Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | New | Project**.
3. In the **New Project** dialog window, choose **Visual C# | SharePoint | 2010** from the **Installed Templates**.
4. Select **Empty SharePoint Project** from the Project Items**.**

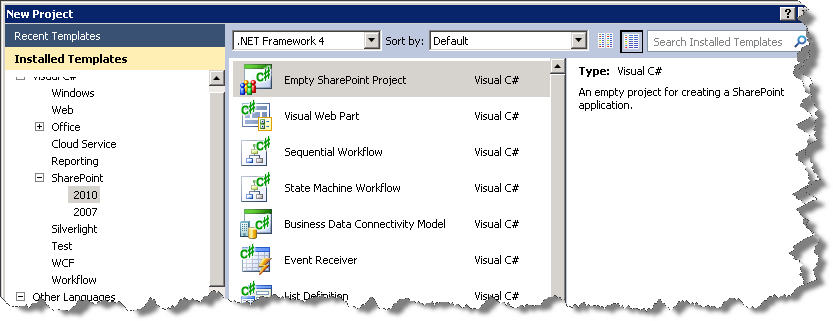


Figure 20 – Empty SharePoint 2010 Project

1. In the Nametextbox, enter **SPHOL302-Ex2**.
2. In the Location textbox, enter **C:\SPHOLS\SPCHOL302\CS\Ex2**.

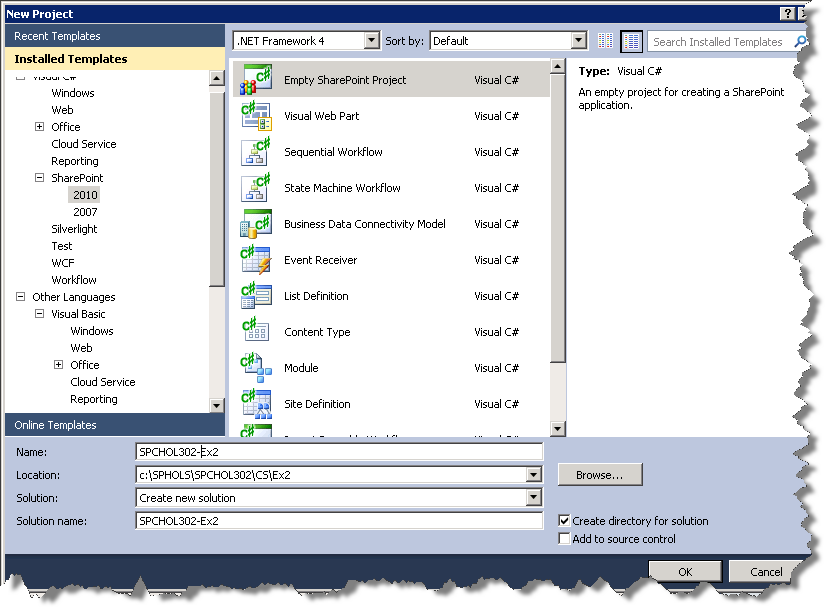


Figure 21 - Visual Studio New Project Dialog Window

1. Uncheck **Create directory for solution**.
2. Click **OK**.
3. In the **SharePoint Customization Wizard**:
   * Enter [**http://intranet.contoso.com**](http://intranet.contoso.com) for the local site.
   * Set the trust level as **Deploy as a farm solution.**
   * Click **Finish.**

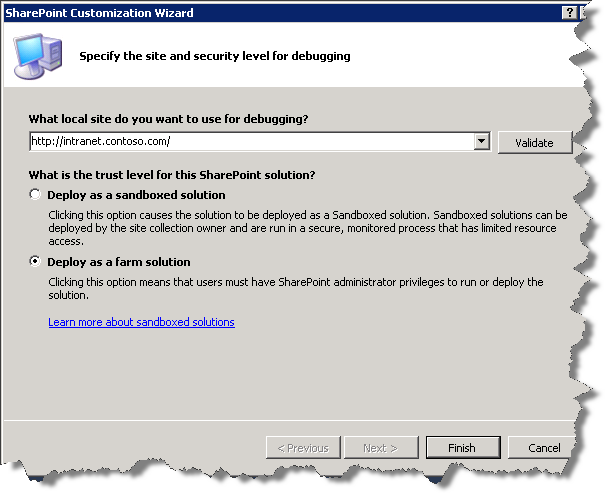


Figure 22 - SharePoint Customization Wizard

1. Visual Studio will create the new **SPCHOL302-Ex2** project and add the necessary files.

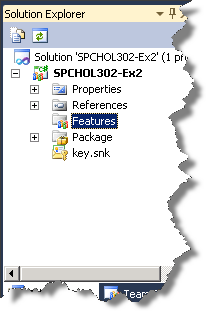


Figure 23 – SPCHOL302-Ex2 Project

### Task 2 – Create a new Content Type

In this task, you will create the new content type.

1. Right-click on the **Features** Folder in the **Solution Explorer** and select **Add Feature**.
2. Right-click on **Feature1** and select **Add Event Receiver.**

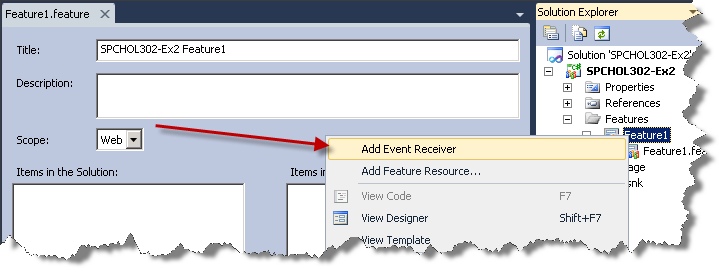


Figure 24 - Add Feature Receiver

1. Visual Studio will add a feature event receiver to **Feature1**.

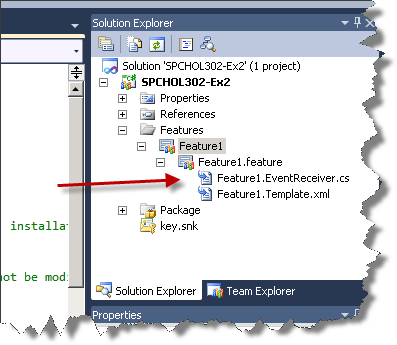


Figure 25 - Feature1 Event Receiver

1. Right-click on **Feature1.EventReceiver.cs** and select **View Code.**
2. Add the following **using statements**:

using System.Linq;

**Code Snippet**: *My Code Snippets | spchol302\_ex2\_namespaces*

1. Uncomment the **FeatureActivated** method in the **Feature1EventReceiver** class.

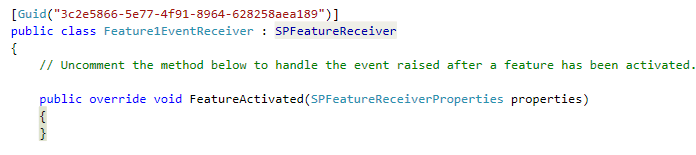


Figure 26 - FeatureActivated Method

1. Insert the following code in the **FeatureActivated** method:

using (SPWeb spWeb = properties.Feature.Parent as SPWeb)

{

SPContentType newAnnouncement = spWeb

.ContentTypes

.Cast<SPContentType>()

.FirstOrDefault(c => c.Name == "New Announcements");

if (newAnnouncement != null)

{

newAnnouncement.Delete();

}

SPField newField = spWeb.Fields

.Cast<SPField>()

.FirstOrDefault(f => f.StaticName == "Team Project");

if (newField != null)

{

newField.Delete();

}

SPContentType myContentType =

new SPContentType(spWeb.ContentTypes["Announcement"],

spWeb.ContentTypes, "New Announcements");

myContentType.Group = "Custom Content Types";

spWeb.Fields.Add("Team Project", SPFieldType.Text, true);

SPFieldLink projFeldLink = new SPFieldLink(spWeb.Fields["Team Project"]);

myContentType.FieldLinks.Add(projFeldLink);

SPFieldLink companyFieldLink = new SPFieldLink(spWeb.Fields["Company"]);

myContentType.FieldLinks.Add(companyFieldLink);

spWeb.ContentTypes.Add(myContentType);

myContentType.Update();

}

**Code Snippet:** *My Code Snippets | spchol302\_ex2\_feature\_activated*

The **FeatureActivated** method will get executed whenever the **Feature1** is activated. This code block:

* Deletes the content type **New Announcements** and the field **Team Project** if they exist.
* Creates a new Content Type **New Announcements**-based parent Content type **Announcement**.
* Creates a new Text field **Team Project** and adds it to the Content Type
* Adds an existing field **Company** to the Content Type

1. Uncomment the **FeatureDeactivating** method.
2. Insert the following code in the **FeatureDeactivating** method:

using (SPWeb spWeb = properties.Feature.Parent as SPWeb)

{

SPContentType myContentType = spWeb.ContentTypes["New Announcements"];

spWeb.ContentTypes.Delete(myContentType.Id);

spWeb.Fields["Team Project"].Delete();

}

**Code Snippet:** *My Code Snippets | spchol302\_ex2\_feature\_deactivating*

The **FeatureDeactivating** method gets executed whenever the **Feature1** is being deactivated. This code block:

* Deletes the new Content Type **New Announcements** created.
* Deletes the new Text field **Team Project** created.

1. In Solution Explorer, right-click on **SPCHOL302-Ex2** and select **Deploy.** This will deploy the solution to the SharePoint site – <http://intranet.contoso.com>

### Task 3 – Verify the new Content Type

In this task, you will verify the existence of the new content type and the new field created in **Task 2.**

1. Open Internet Explorer and browse to the following site:

<http://intranet.contoso.com>

1. If prompted for authentication, enter the following details.

Username: Administrator

Password: pass@word1

1. Click **Site Actions | Site Settings** from the SharePoint Ribbon.

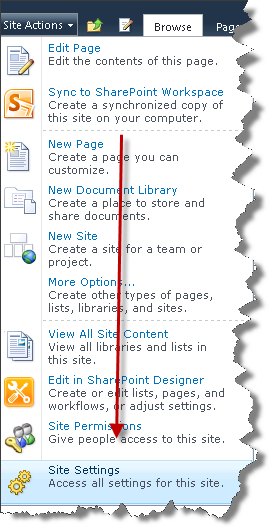


Figure 27 - Site Actions Menu

1. Click on **Site Columns** under **Galleries**
2. Select **Custom Columns** from the **Show Group** options.

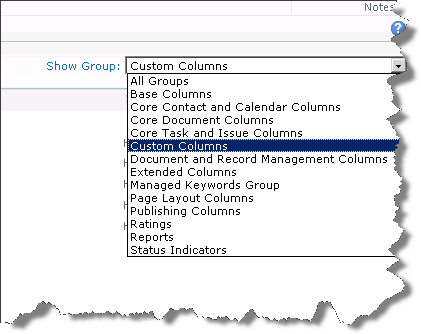


Figure 28 - Custom Columns

1. You should see the new field **Team Project** created in our solution

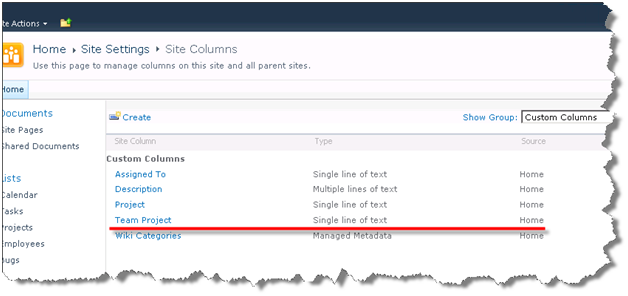


Figure 29 - Team Project Field

1. Click **Site Actions | Site Settings** from the SharePoint Ribbon.
2. Click on **Site content types** under **Galleries.**
3. Select **Custom Content Types** from the **Show Group** options.

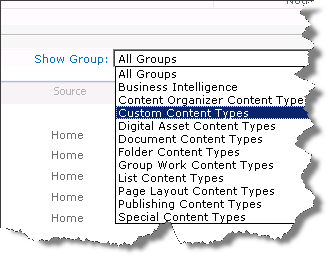


Figure 30 - Custom Content Types

1. You should see the new Content Type **New Announcements** created in our solution.

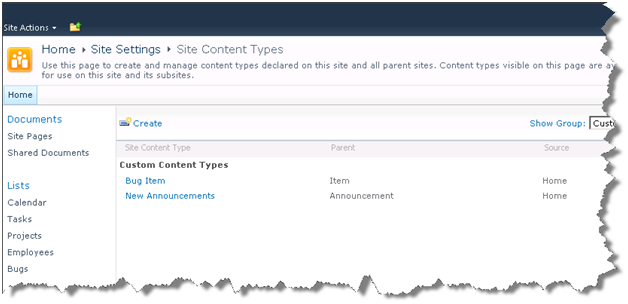


Figure 31 - New Announcements Content Type

### Task 4 – Create a new List based on the new Content Type

In this task, you will create a new List and associate the List with the new content type **New Announcement.**

1. Click **Site Actions | More Options…**

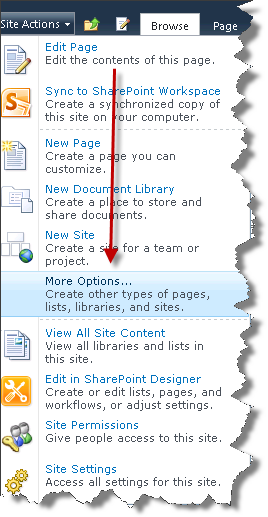


Figure 32 - Site Actions | More Options

1. Click on **List** in the **Filter By** menuin the **Create** dialog.

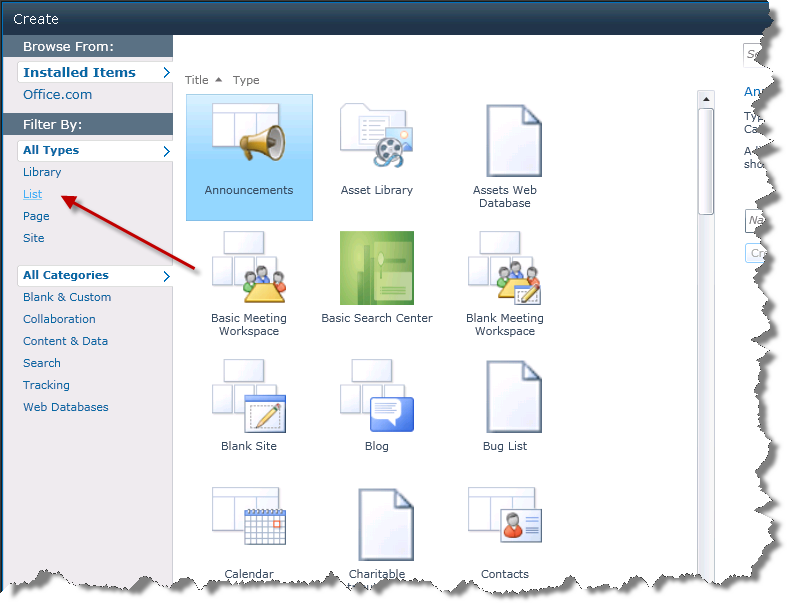


Figure 33 - Create *New Announcements* List

1. Select **Custom List**

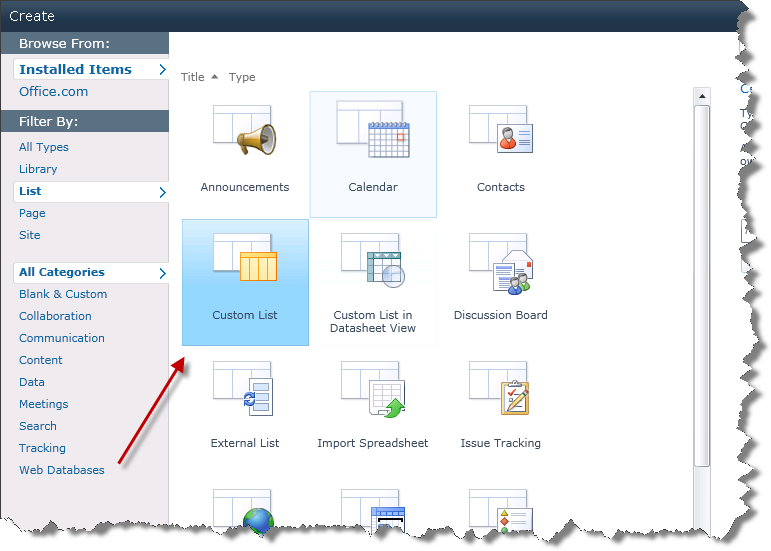


Figure 34 - Custom List

1. Enter **New Announcements** as the name and click **Create.**

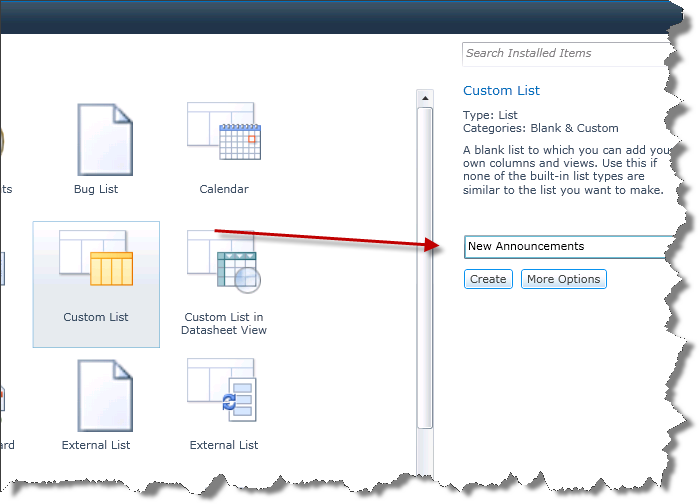


Figure 35 - Create *New Announcements* List

1. Click on **List Settings** in the SharePoint Ribbon.

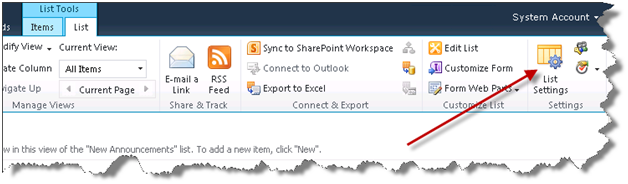


Figure 36 - List Settings

1. Click on **Advanced settings** under **General Settings.**
2. Select **Yes** for **Allow management of content types?.** Click **OK** (you may have to scroll in the browser window to see the **OK** button).

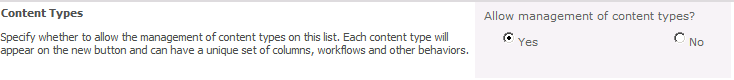


Figure 37 - Allow management of content types

1. Click on **Item** content type under the **Content Types** section.
2. Click on **Delete this content type** under the **Settings.**

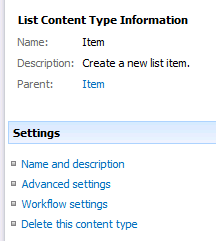


Figure 38 - Item Content Type Information

1. If prompted for confirmation, select **OK.**
2. Click on **Add from existing site content types** under the **Content Types** section.
3. Select **New Announcements** from the available site content types list and click **Add >**

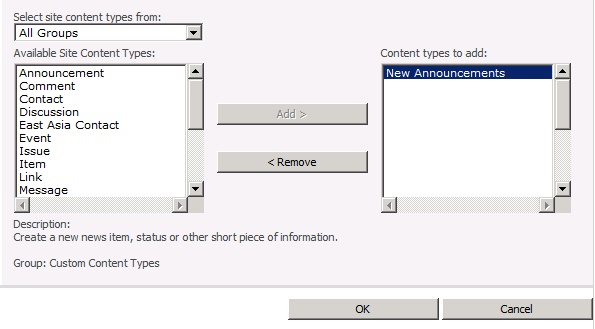


Figure 39 - Add *New Annoucements* Content Type

1. Click **OK.**
2. You should now see the **New Announcements** content type associated with the **New Announcements** list.

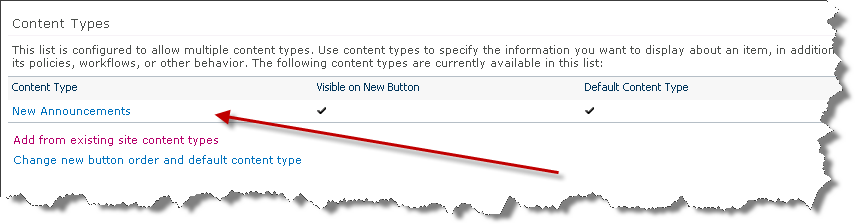


Figure 40 - *List* Content Types

### Task 5 – Disassociate a List from a Content Type

In this task, you will disassociate the list **New Announcements** created in **Task 2** from the new content type **New Announcements.**

1. Click **Site Actions | View All Site Content**
2. Click on **New Announcements** under **Lists**

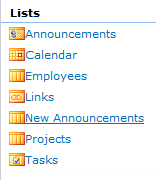


Figure 41 - Site Lists

1. Click on **List |** **List Settings** in the SharePoint Ribbon.
2. Click on **New Announcements** under the **Content Types** section.
3. Click on **Delete this content type**

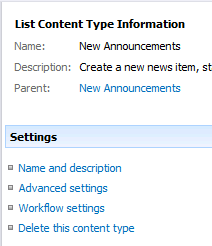


Figure 42 - Delete New Announcements Content Type from the list

1. If prompted for confirmation, select **OK.**

### Task 6 – Delete the List

1. Click on **Delete this list** under the **Permissions and Management** section

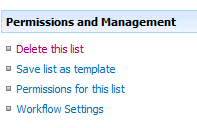


Figure 43 - Delete the New Announcements List

1. If prompted for confirmation, click **OK**

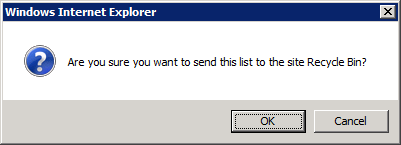


Figure 44 - Confirm Deletion

In the past few minutes you saw how to:

* Programmatically create a new Content Type.
* Programmatically create a new Text field.
* Add the new Text field to the Content Type.
* Add an existing field to the Content Type.
* Deploy the solution and activate the new Content Type feature from Visual Studio 2010.

## Lab Summary

In this lab you performed the following exercises.

Created Linq queries across 3 lists using the new Linq-to-SharePoint model

Programmatically created a new Content Type, created a new List, and associated the new Content Type to that list.

In this lab, you learned how to write simple and complex Linq queries that spanned across 3 different SharePoint lists – Projects, Employees, and Bugs. Queries were performed to get the employees based on specific conditions in the Linq query. You also learned how to create Content Types programmatically and associate a List to the content type.