# Demo 1: Using Social Features

In this Demo, you will utilize some of the social features of SharePoint 2013.

## Step 1 - Sharing Activities and Documents

In this Step, you will share activities and documents with other users.

1. Log in to the SharePoint portal using the account **CORP\patriciad**.
   * + - * Patricia Doyle is the President and CEO. So she is likely to have many followers interested in the information she shares.
   1. On the portal home page, click **SkyDrive**.
   2. If prompted, allow SharePoint to change your settings to support the Social features.
   3. Click the **Shared with Everyone** folder.
   4. Click **New Document.**
   5. In the Add a Document dialog, **Browse** to the file **Contoso Limited.docx** and select **Open**.
   6. Click **OK**.
   7. Click the ellipsis associated with the document and select **Share** from the menu.
   8. In the Share dialog, enter **CORP\PortalUsers** and click **Share**.
   9. Click **Newsfeed**.
   10. Enter the following message in the **Share something with everyone** field and click **Post**.

I’m pleased to announce the merger with Fabrikam, Inc is complete. Many thanks to every one of you who worked so hard to make it happen! I have shared a memo outlining the merger so the entire staff can read the details.

## Step 2 - Follow People, Sites, and Documents

In this Step, you will people, sites, and documents.

1. Log in to the SharePoint portal using the account **CORP\brianc**.
   * + - * Brian Cox is a Solutions Developer at Contoso. He wants to follow the Activities of the President.
   1. Click **Newsfeed**.
   2. Click **Follow**.
   3. Click **Follow** again.
   4. In the Follow People dialog, enter **CORP\patriciad** and click **Follow**.
   5. Click the ellipsis next to Patricia Doyle to see her activities.
   6. Click **See Conversation**.
   7. In the thread, click **Like**.
   8. Click **Reply**, enter the following message, and click **Post**.

Great news for the company! Congratulations to all!

* 1. Click **Patricia Doyle**.
  2. Click **Documents**.
  3. Click the ellipsis associated with the **Contoso Limited** document.
  4. Click **Follow**.
  5. Click **Newsfeed**.
  6. Click the elipses to see **Activities**..
  7. **Navigate** to the home page of the Site Collection you are using for the Demo.
  8. **Click** the star near the Site menu to start following the site.
  9. Click **Newsfeed**.
  10. Click the elipses to see **Activities**..

# Demo 2: Programming the Microfeed

In this Demo, you will create a custom solution that uses the Microfeed API to read and write activities.

## Step 1 – Create the Windows Application

In this Step, you will create the windows application project.

1. Open Microsoft Visual Studio 2012 and create a new Console Application
   1. **Open** Microsoft Visual Studio 2012
   2. Select **File⮚New Project** from the main menu
   3. Click the **Templates⮚Visual C#⮚** node and select the **Windows Form Application** template
   4. Name the new project **FeedDeck**
   5. Click the **OK** button
2. Open the Toolbox and add controls to the form.
   1. Select **View⮚Toolbox** from the main menu in Visual Studio.
   2. Add 3 **TextBox** controls, 1 **NumericUpDown** control, and 2 **Button** controls to **Form1**.
   3. Name the **TextBox** controls **accountName**, **feedThreads**, and **responsePost**.
   4. Set the **MultiLine** property for the **feedThreads** control to **True**.
   5. Name the **NumericUpDown** control **threadCount**.
   6. Name the buttons **getActivities** and **postReply**.
   7. Set the **Text** property of the **getActivities** control to **Get Activities**.
   8. Set the **Text** property of the **postReply** control to **Post Reply**.
   9. **Arrange** the controls on the form as shown in the following figure.

## Step 2 – Code the Application

In this Step, you will set references and add code to the application.

1. In the Solution Explorer, right click the **References** node and select **Add Reference**.
2. Click **Browse** and Navigate to **\Program Files\Common Files\Microsoft Shared\web server extensions\15\ISAPI**
3. **Select** the following assemblies and click **Add**.

* Microsoft.SharePoint.Client.dll
* Microsoft.SharePoint.Client.Runtime.dll
* Microsoft.SharePoint.Client.UserProfiles.dll

1. In the **Reference Manager** dialog, click **OK**.
2. Open **Form1.cs** for editing.
3. **Add** the following statements to the top of the code file.

using Microsoft.SharePoint.Client;

using Microsoft.SharePoint.Client.Microfeed;

using Microsoft.SharePoint.Client.UserProfiles;

* + - * + After the above statements are added, you will notice that the dev environment detects ambiguity in the reference to the Form class. The next step will clear up this error.

1. **Fully qualify** the reference to the Form class by **altering** the class definition to appear as follows:

public partial class Form1 : System.Window.Forms.Form

1. Add the following member variables to the Form1 class.

ClientContext clientContext;

MicrofeedManager microfeedMgr;

Dictionary<int, string> idDictionary;

1. **Add** the following code to establish a client context for the application.
   * + - * Be sure to alter the URL to match the Site Collection for you lab environment.

private void Form1\_Load(object sender, EventArgs e)

{

clientContext = new ClientContext([URL for Site Collection]);

}

1. **Add** the following code to retrieve the activities for a given user account.

private void LoadThreads()

{

try

{

feedThreads.Text = string.Empty;

string targetUser = accountName.Text;

// Get the MicrofeedManager object.

microfeedMgr = new MicrofeedManager(clientContext);

// Get the properties for the target user

PersonProperties personProps =

new PeopleManager(clientContext).GetPropertiesFor(targetUser);

// Get Display Name and Account Name for the target user

clientContext.Load(personProps, o => o.DisplayName, o => o.AccountName);

clientContext.Load(microfeedMgr);

clientContext.ExecuteQuery();

// Specify the feed content that you want to retrieve.

MicrofeedRetrievalOptions retrievalOptions =

new MicrofeedRetrievalOptions();

retrievalOptions.IncludedTypes = MicroBlogType.RootPost;

retrievalOptions.ThreadCount = 5;

// Get all of the target owner's posts and activities

ClientResult<MicrofeedThreadCollection> threads =

microfeedMgr.GetPublishedFeed(

personProps.AccountName,

retrievalOptions,

MicrofeedPublishedFeedType.Full);

clientContext.ExecuteQuery();

//Create a dictionary to store the thread identifiers

idDictionary = new Dictionary<int, string>();

for (int i = 0; i < threads.Value.Count; i++)

{

MicrofeedThread thread = threads.Value[i];

// Keep only user-sourced threads (not events).

if (thread.DefinitionName ==

"Microsoft.SharePoint.Microfeed.UserPost")

{

//Save thread identifier

idDictionary.Add(i, thread.Identifier);

// Write out the text of the post.

feedThreads.Text += (

personProps.DisplayName + ": " +

(i + 1) + ". " + thread.RootPost.Content + "\r\n");

}

}

threadCount.Maximum = threads.Value.Count;

}

catch (Exception x)

{

MessageBox.Show(x.Message);

}

}

1. **Add** the following code to initiate the retrieval of activities.

private void getActivities\_Click(object sender, EventArgs e)

{

LoadThreads();

}

1. Press **F5** to debug the application.
2. Enter **CORP\patriciad** in the **accountName** TextBox and click **Get Activities**.
3. Select **Debug⮚Stop Debugging** from the main menu in Visual Studio.
4. **Add** the following code to post a reply to an activity thread.

private void postReply\_Click(object sender, EventArgs e)

{

try

{

string threadId = string.Empty;

//Get the thread identifier

idDictionary.TryGetValue(

(int)(threadCount.Value - 1), out threadId);

// Define properties for the reply.

MicrofeedPostOptions postOptions = new MicrofeedPostOptions();

postOptions.Content = responsePost.Text;

// Register the reply.

microfeedMgr.PostReply(threadId, postOptions);

// Make the changes on the server.

clientContext.ExecuteQuery();

MessageBox.Show("Reply Posted!");

}

catch (Exception x)

{

MessageBox.Show(x.Message);

}

}

1. Press **F5** to debug the application.
2. Enter **CORP\patriciad** in the **accountName** TextBox and click **Get Activities**.
3. After the activities load, select **1** in the **threadCount** control to indicate that you want to reply to the first thread.
4. **Enter** the following in the **responsePost** TextBox and click **Post Reply**.

@brianc did a fantastic job on the web site

1. **Review** Patricia Doyle’s activity feed in the browser to verify the post was made and that Brian Cox is properly mentioned.

# Demo 3: Setting up a Community

In this Demo, you will create a new Community site in SharePoint 2013.

## Step 1 - Create the Community Site

In this Step, you will create the Community site using the new template.

1. **Navigate** to the home page of the Site Collection you are using for the Demo.
2. Click the **Site Contents link** and click **New Site**.
   1. On the **New Site** page, enter **SharePoint Support** in the **Title** field.
   2. Give the site a **Description**.
   3. Enter **spsupport** in the **URL Name** field.
   4. Select **Community Site** as the template.
   5. Click **Create**.

## Step 2 - Configure the Community Site

In this Step, you will configure the various options in the Community site.

1. On the home page of the SharePoint Support site, click **Manage Categories**.
   1. In the Categories list, click **New Item**.
   2. Enter **Setup, Upgrade, Administration, and Operation** in the **Category Name** field.
   3. Enter **Discuss setup, upgrade, administration and operation for SharePoint 2010 and SharePoint services such as Access Services, BCS, Excel Services and Visio Services** in the **Description** field.
   4. Click **Save**.
   5. In the Categories list, click **New item**.
   6. Enter **Using SharePoint Designer, InfoPath, and Other** field.
   7. Enter **Discuss using SharePoint Designer, SharePoint Gallery Solutions, templates & other customization for SharePoint 2010 and SP services such as Access Services, BCS, Excel Services, & Visio Services** in the **Description** field.
   8. Click **Save**.
   9. In the Categories list, click **New Item**.
   10. Enter **Using Visual Studio with SharePoint and other programming** field.
   11. Enter **Discuss using Visual Studio and other programming with SharePoint 2010 and SharePoint services such as Access Services, BCS, Excel Services and Visio Services** in the **Description** field.
   12. Click **Save**.
2. On the home page of the SharePoint Support site, click **Reputation Settings**.
   1. Click **Manage the List of Gifted Badges**.
   2. Click **Add**.
   3. Enter **Moderator** in the **Badge Name** field.
   4. Click **Save**.
3. On the home page of the SharePoint Support site, click **Assign Badges to members**.
   1. In the Members list, select your record and click **Moderation⮚Give Badge** in the ribbon.
   2. Select **Moderator** as the Gifted Badge.
   3. Click **Save**.
4. On the home page of the SharePoint Support site, click **Community Settings**.
   1. Check **Enable Reporting of Offensive Content**.
   2. Click **OK**.

## Step 3 - Using the Community Site

In this Step, you will post content to the community site.

1. Log into the SharePoint portal as **CORP\brianc**.
2. **Navigate** to the home page of the SharePoint Support site.
3. Click **Start a New Discussion**.
   1. Enter **Using SharePoint Search in an App** in the **Subject** field.
   2. Enter **What permissions are required to use SharePoint Search from an App?** In the **Body** field.
   3. Check the **Question** box.
   4. Click **Save**.
4. Log into the SharePoint portal as **CORP\administrator**.
5. **Navigate** to the home page of the SharePoint Support site.
   1. Click **Manage Discussions**.
   2. In the Discussions list, select the discussion started by Brian Cox and choose **Items⮚Edit Item** from the ribbon.
   3. Change the Category to **Using Visual Studio with SharePoint and other programming**.
   4. Click **Save**.
   5. **Return** to the home page of the SharePoint Support site.
   6. Click **Using Search in a SharePoint App**.
   7. Enter **Moved to programming category** and click **Reply**.
6. Log into the SharePoint portal as **CORP\davidy**.
7. **Navigate** to the home page of the SharePoint Support site.
   1. Click **Using Search in a SharePoint App**.
   2. Enter **In Beta 1 the permission is:**  
      **Scope=http://search/query  
      Right=QueryAsUserIgnoreAppPrincipal  
      but in Beta 2 it will be:  
      Scope=http://sharepoint/search Right=QueryAsUserIgnoreAppPrincipal**
   3. Click **Reply**.