

Lab 02: Code Sharing Existing Projects

Prerequisites

You will need a development environment, either a Mac or Windows PC with the Android SDK, iOS SDK and Xamarin tools installed. We will be using the Android and iOS emulators to test the code we are building, so make sure to have a virtual device already configured and ready to run. See the [Cross-Platform Installation Guild](#) if you need help getting your environment setup.

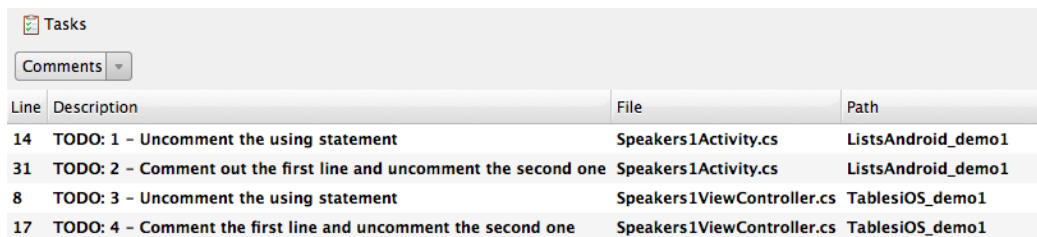
Downloads

Included with this lab document is a folder with resources that you will need in order to complete the lab. The folder name is **Fundamentals Introduction to Cross Platform Resources**. Make sure you have this folder before you begin.

Lab Goals

The goal of this lab is to take an existing Android and iOS mobile project, combine them into a single solution and move the shared code into a Portable Class Library (PCL) to maximize code re-use while still delivering high-quality native experience on all of the platforms.

The lab has been provided as a starter solution with most of the code already filled in for you – as you following along with the instructor you will make small changes for each step, either writing a little code or uncommenting a block of code. Most of these steps are clearly marked in the supplied solution with `// TODO:` comments. These comments are picked up by Xamarin Studio and shown in the Task Pad, which you can make visible either by clicking the Tasks button in the status bar of the application, or through the **View > Pads > Tasks** menu item. When the Tasks Pad is open, it will look like this:



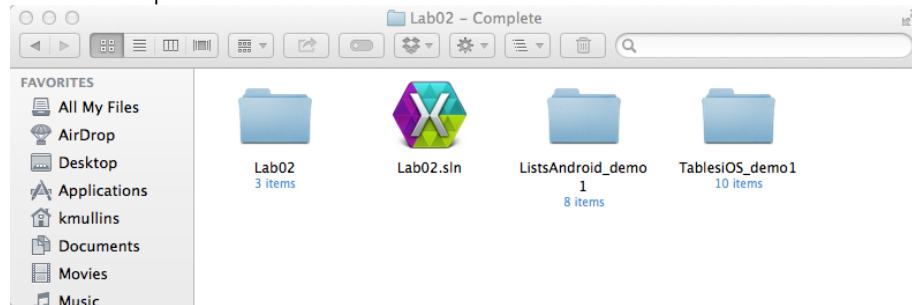
Tasks			
Comments ▾			
Line	Description	File	Path
14	TODO: 1 - Uncomment the using statement	Speakers1Activity.cs	ListsAndroid_demo1
31	TODO: 2 - Comment out the first line and uncomment the second one	Speakers1Activity.cs	ListsAndroid_demo1
8	TODO: 3 - Uncomment the using statement	Speakers1ViewController.cs	TablesiOS_demo1
17	TODO: 4 - Comment the first line and uncomment the second one	Speakers1ViewController.cs	TablesiOS_demo1

Steps

Open the Begin Solution

1. Launch **Xamarin Studio** using Spotlight or the application icon

2. Click **Open...** on the Xamarin Studio Welcome Screen and navigate to the **Fundamentals Introduction to Cross Platform Resources** folder included with this document
3. Locate the **Lab 02 - Begin** folder – make sure it's the starter and not the completed folder.
4. Inside the **Lab 02 - Begin** folder you will find a `Lab02.sln` file – double click on this file to open the starter solution:

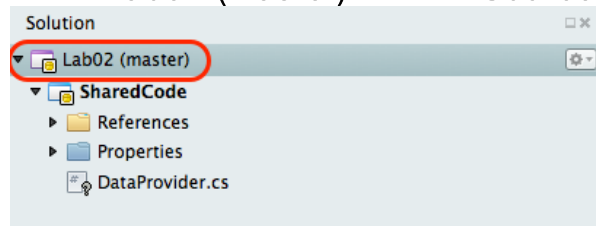


5. Go ahead and build and run the solution to make sure it compiles and your environment is ready. Let the instructor know if you have any trouble.

Adding the Android Project

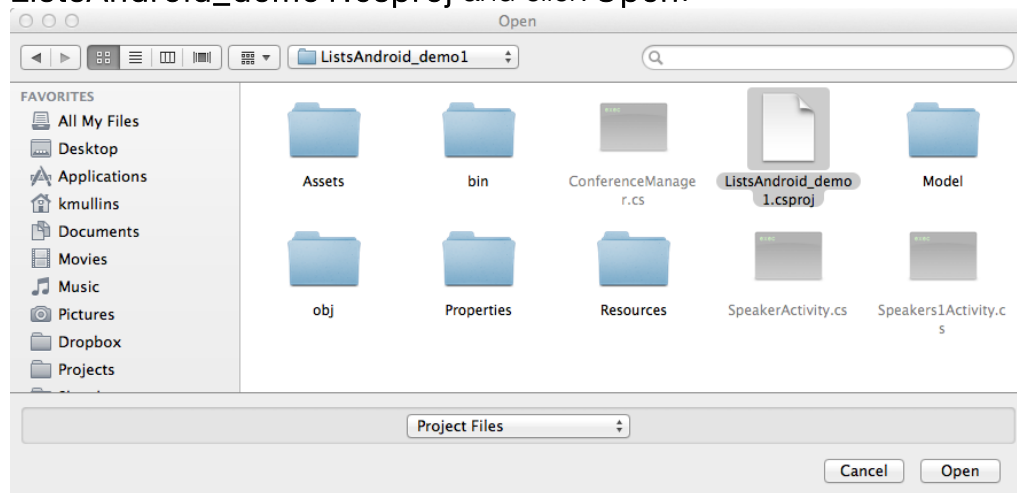
We are going to add an existing Android Project to our solution so we can move the shared code to a common PCL project.

1. Select **Lab02 (master)** from the **Source Tree**:



2. Right click on **Lab02 (master)** and select **Add > An Existing Project...**

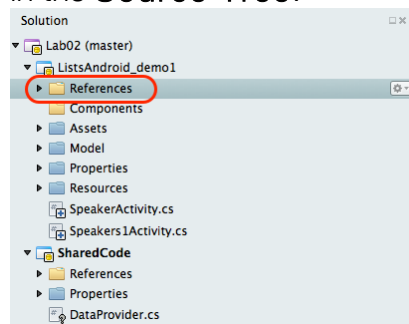
3. Navigate to the **ListsAndroid_demo1** folder, select **ListsAndroid_demo1.csproj** and click **Open**:



Adding a Reference to the PCL Project

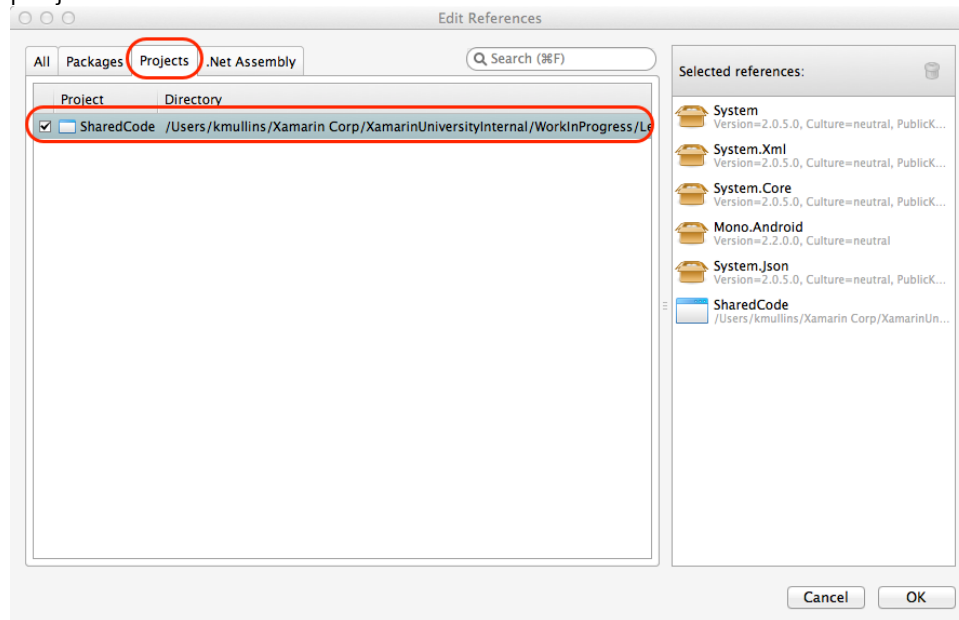
Next we need to add a reference to the PCL project included in this solution so that the Android project can access the shared data provider it contains.

4. Select the **References** folder under the **ListsAndroid_demo1** project in the **Source Tree**:



5. Right click on **References** and select **Edit References...**

- Click on the **Projects** tab and place a check by the **SharedCode** project:

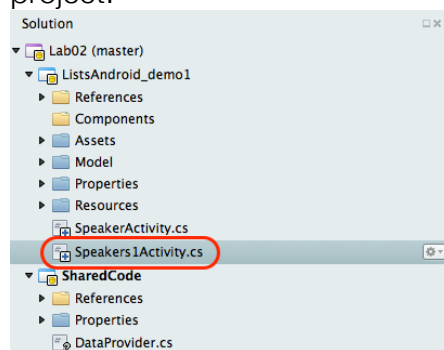


- Click the **OK** button to add the reference and close the dialog box.

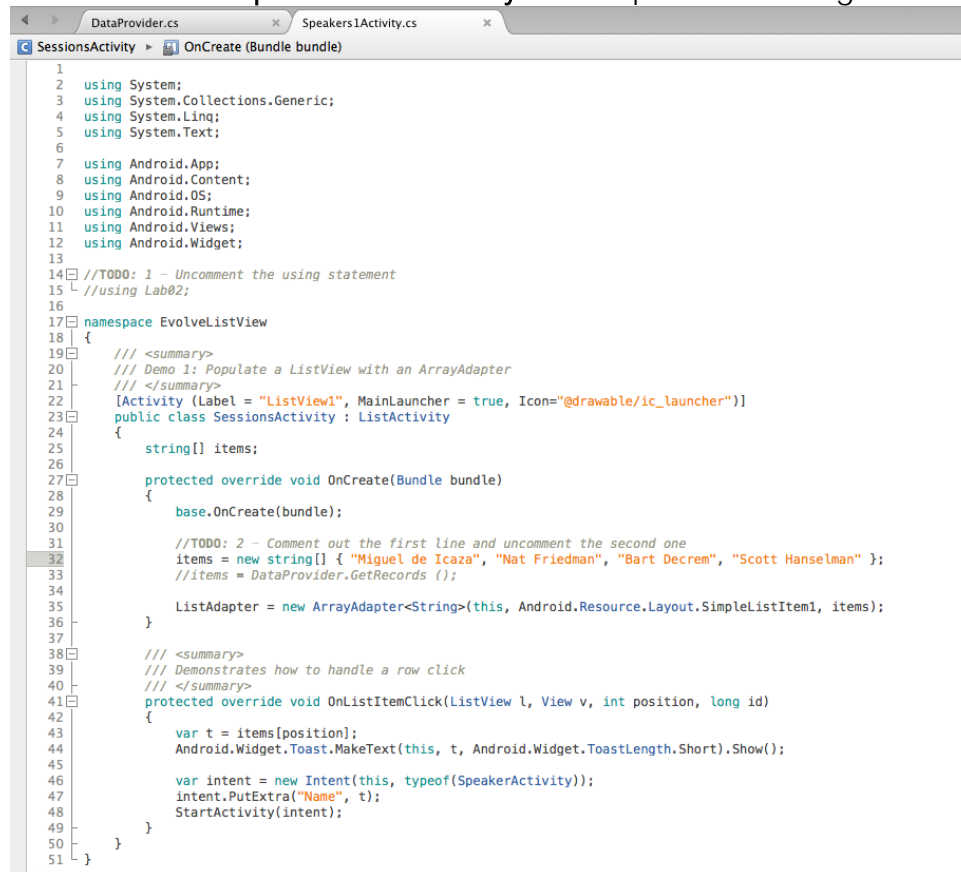
Sharing Common Code

Next we are going to remove the common code from our Android project and replace it with the data provide from the PCL included with the solution.

- Select **Speakers1Activity.cs** file under the **ListsAndroid_demo1** project:



2. Double click on **Speakers1Activity.cs** to open it for editing:



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Text;
6
7  using Android.App;
8  using Android.Content;
9  using Android.OS;
10 using Android.Runtime;
11 using Android.Views;
12 using Android.Widget;
13
14 //TODO: 1 - Uncomment the using statement
15 //using Lab02;
16
17 namespace EvolveListView
18 {
19     /// <summary>
20     /// Demo 1: Populate a ListView with an ArrayAdapter
21     /// </summary>
22     [Activity (Label = "ListView1", MainLauncher = true, Icon="@drawable/ic_launcher")]
23     public class SessionsActivity : ListActivity
24     {
25         string[] items;
26
27         protected override void OnCreate(Bundle bundle)
28         {
29             base.OnCreate(bundle);
30
31             //TODO: 2 - Comment out the first line and uncomment the second one
32             items = new string[] { "Miguel de Icaza", "Nat Friedman", "Bart Decrem", "Scott Hanselman" };
33             //items = DataProvider.GetRecords ();
34
35             ListAdapter = new ArrayAdapter<String>(this, Android.Resource.Layout.SimpleListItem1, items);
36
37
38             /// <summary>
39             /// Demonstrates how to handle a row click
40             /// </summary>
41             protected override void OnItemClickListener(ListView l, View v, int position, long id)
42             {
43                 var t = items[position];
44                 Android.Widget.Toast.MakeText(this, t, Android.Widget.ToastLength.Short).Show();
45
46                 var intent = new Intent(this, typeof(SpeakerActivity));
47                 intent.PutExtra("Name", t);
48                 StartActivity(intent);
49             }
50         }
51     }

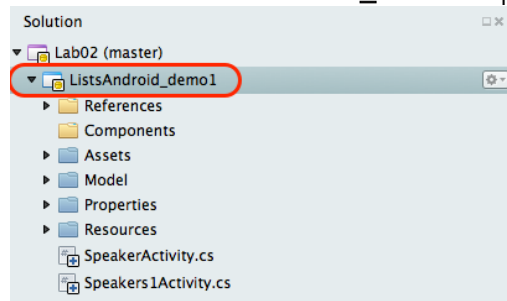
```

3. Locate the *//TODO: 1 - Uncomment the using statement* and uncomment the using statement:
using Lab02;
4. Locate the *//TODO: 2 -*
Comment out the first line and uncomment the second one and comment the first line and uncomment the second one:
//items = new string[] { "Miguel de Icaza", "Nat Friedman", "Bart Decrem", "Scott Hanselman" };
items = DataProvider.GetRecords ();
5. **Save** the project.
6. **Build** the project and ensure that there are no errors.

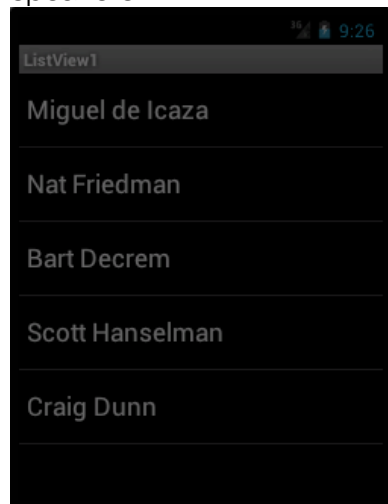
Testing the Android Application

Next we will test our Android application and assure that it is consuming the shared data provided by the data provider in the PCL.

1. Select the **ListsAndroid_demo1** project from the **Source Tree**:



2. Right click on **ListsAndroid_demo1** and select **Set As Startup Project**.
3. Click on the **Run** menu and select **Start Debugging**.
4. The application will run in the Android emulator and display a list of speakers:

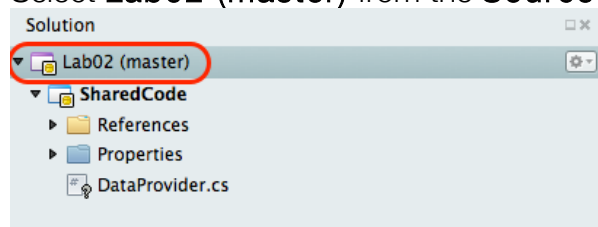


5. Stop debugging.

Adding the iOS Project

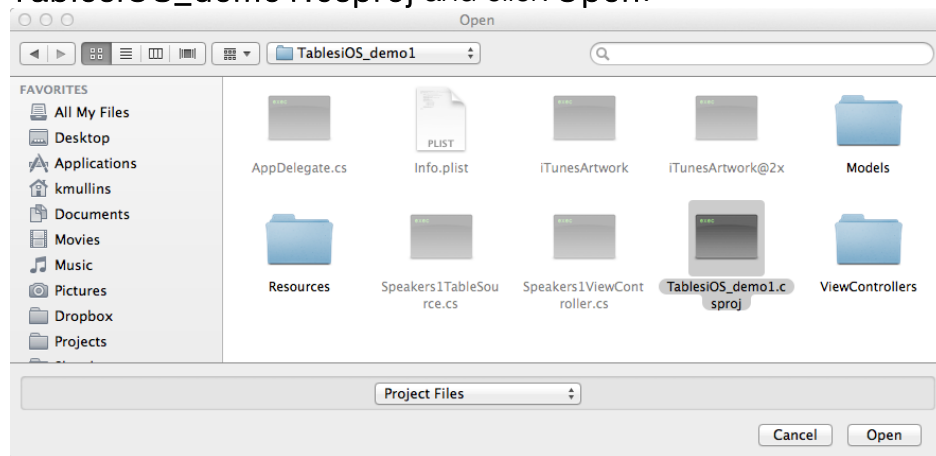
Like the Android project above, we are going to add an existing iOS Project to our solution and reference the shared code from the common PCL project.

1. Select **Lab02 (master)** from the **Source Tree**:



2. Right click on **Lab02 (master)** and select **Add > An Existing Project...**

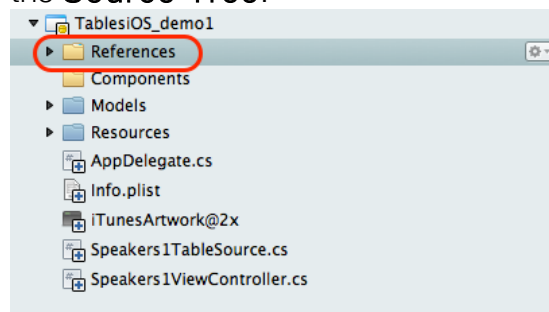
3. Navigate to the **TablesiOS_demo1** folder, select **TablesiOS_demo1.csproj** and click **Open**:



Adding a Reference to the PCL Project

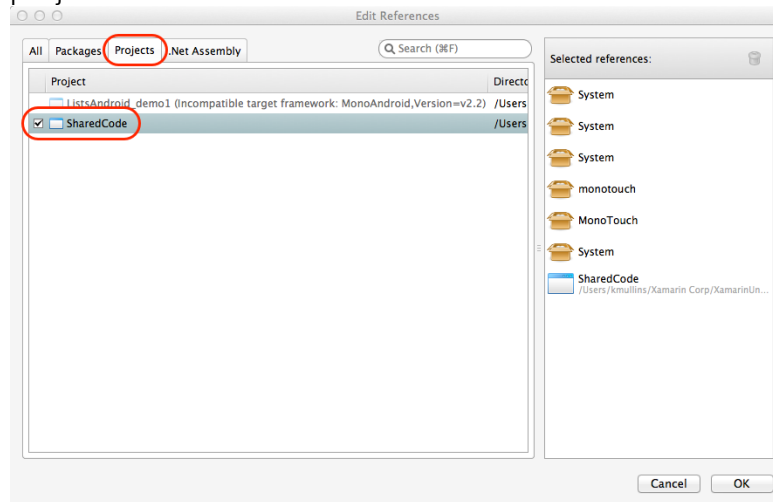
Next we need to add a reference to the PCL solution so that the iOS project can access the shared data provider.

1. Select the **References** folder under the **TablesiOS_demo1** project in the **Source Tree**:



2. Right click on **References** and select **Edit References...**

- Click on the **Projects** tab and place a check by the **SharedCode** project:

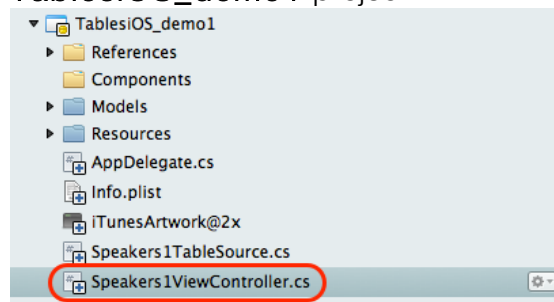


- Click the **OK** button to add the reference and close the dialog box.

Sharing Common Code

Next we are going to remove the common code from our iOS project and replace it with the data provide from the PCL included with the solution.

- Select **Speakers1ViewController.cs** file under the **TablesiOS_demo1** project:



- Double click on **Speakers1ViewController.cs** to open it for editing:

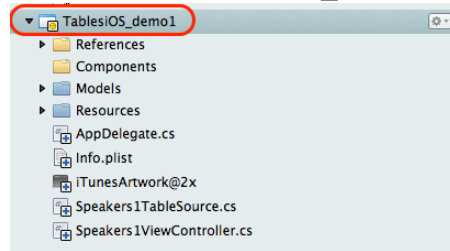


3. Locate the **//TODO: 3** - *Uncomment the using statement* and uncomment the using statement:
`using Lab02;`
4. Locate the **//TODO: 4** -
Comment out the first line and uncomment the second one and comment the first line and uncomment the second one:
`//string[] items = new string[] { "Miguel de Icaza", "Nat Friedman", "Bart Decrem", "Scott Hanselman" };
string[] items = DataProvider.GetRecords ();`
5. **Save** the project.
6. **Build** the project and ensure that there are no errors.

Testing the iOS Application

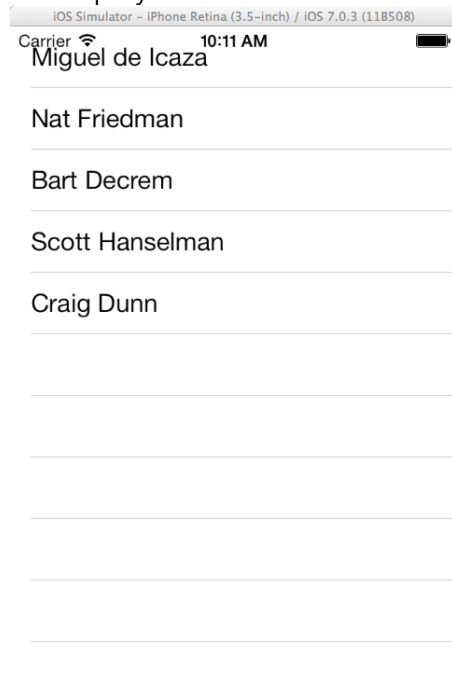
Next we will test our iOS mobile application and assure that it is consuming the shared data provided by the data provider in our PCL.

1. Select the **TablesiOS_demo1** project from the **Source Tree**:



2. Right click on **TablesiOS_demo1** and select **Set As Startup Project**.
3. Click on the **Run** menu and select **Start Debugging**.

4. The application will run in the iOS Simulator and the list of speakers will be displayed:



5. Stop debugging.

Summary

In this lab we learned how to take existing platform specific mobile projects, add them to a single solution and move the shared code into a single PCL to maximize code reuse, while maintaining high quality native applications on each platform supported.