Portable Visual Basic.NET

Visual Basic.NET for Xamarin using Portable Class Libraries

Overview

In this guide we're going to walk through creating a new Visual Basic class library in Visual Studio as a Portable Class Library (PCL) project that can be referenced from Xamarin.iOS and Xamarin.Android projects.

Xamarin iOS and Android projects do not natively support Visual Basic; however developers can use Portable Class Libraries to migrate existing Visual Basic code to iOS and Android, or to write significant portion of their application logic in Visual Basic.

Requirements

Portable Class Library Support was added in Xamarin.Android 4.10.1, Xamarin.iOS 7.0.4 and Xamarin Studio 4.2, meaning any Xamarin projects created with those tools can incorporate Visual Basic PCL assemblies.

To create and compile Visual Basic Portable Class Libraries you should use Visual Studio 2012 or newer on Windows.

NOTE: Visual Basic libraries can only be created and compiled using Visual Studio. Xamarin.iOS and Xamarin.Android do not support the Visual Basic language.

If you work solely in Visual Studio you can reference the Visual Basic project from Xamarin.iOS and Xamarin.Android projects.

If your iOS and Android projects must also be loaded in Xamarin Studio you should reference the output assembly from the Visual Basic PCL.

Visual Basic.NET in Visual Studio

This section walks through how to create a Visual Basic Portable Class Library using Visual Studio. Later sections describe how to reference the PCL in other projects, including Xamarin.iOS and Xamarin.Android apps.

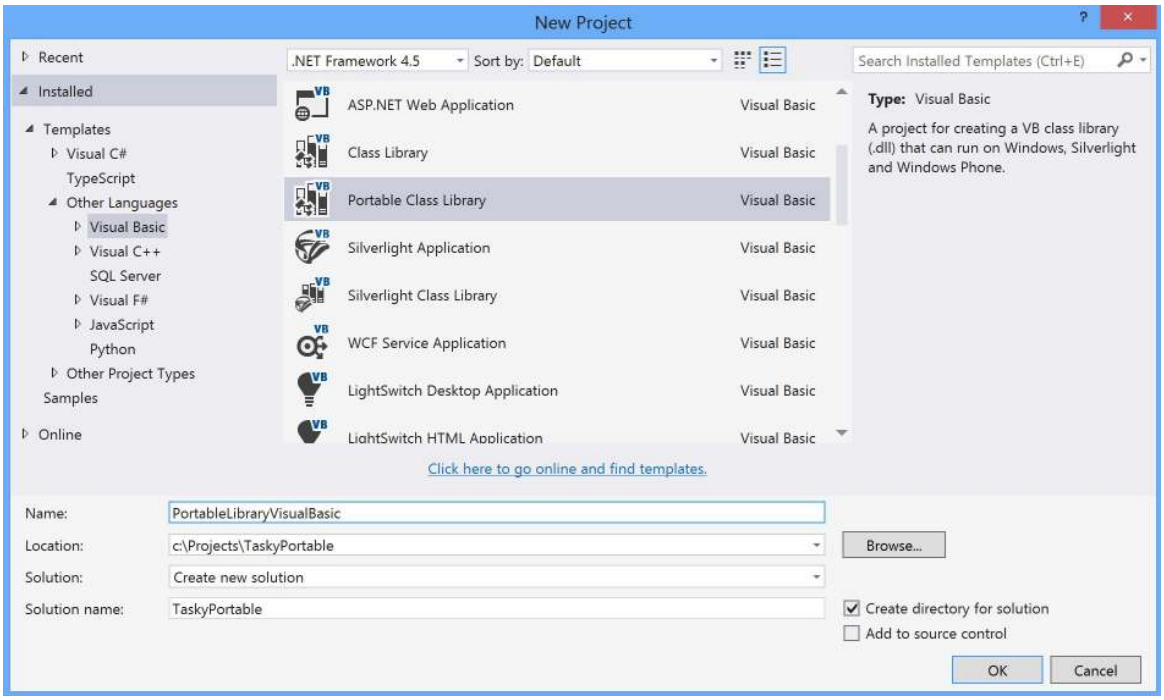
Creating a PCL

When adding a Visual Basic PCL in Visual Studio you must choose a Profile that describes what platforms your

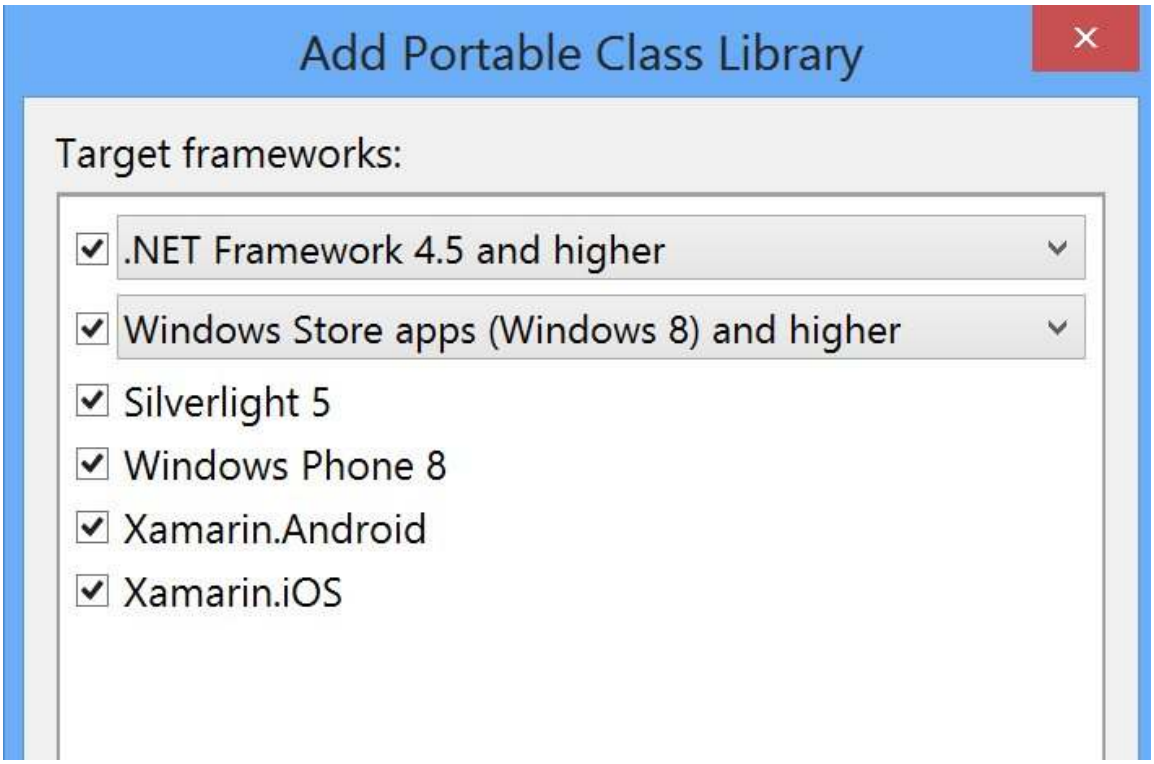
library should be compatible with. Profiles are explained in the Introduction to PCL document.

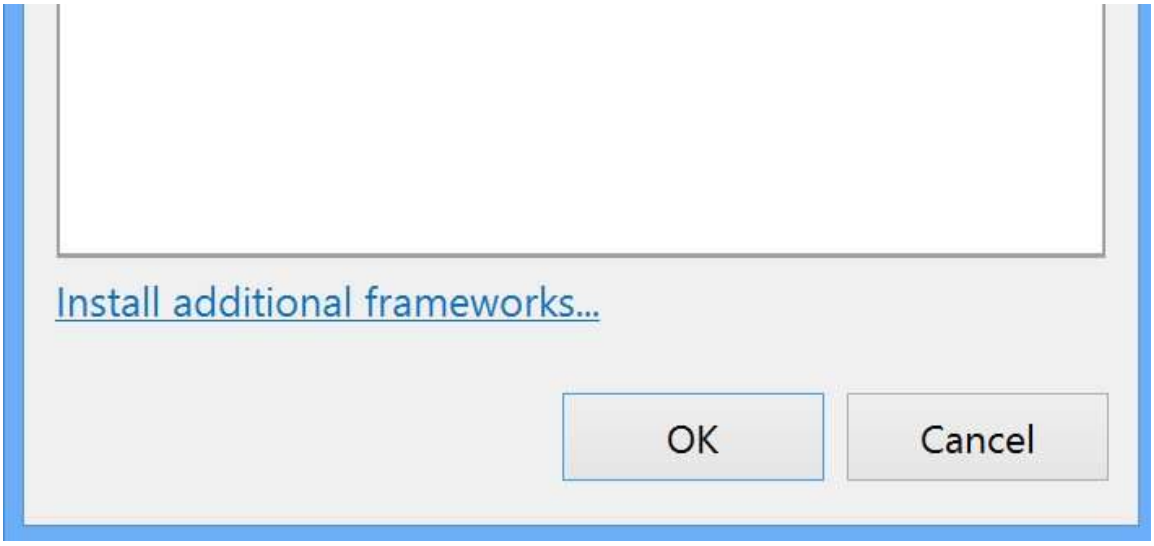
The steps to create a PCL and choose its Profile are:

1. In the `New Project` screen, select the `Visual Basic > Portable Class Library` option:

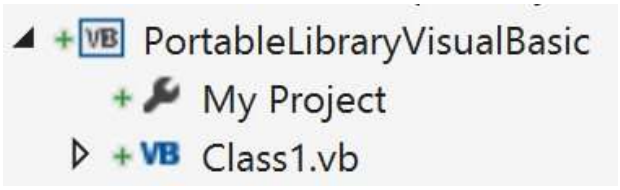


2. Visual Studio will immediately prompt with the following `Add Portable Class Library` dialog so that the Profile can be configured. Tick the platforms you need to support and press `OK`.





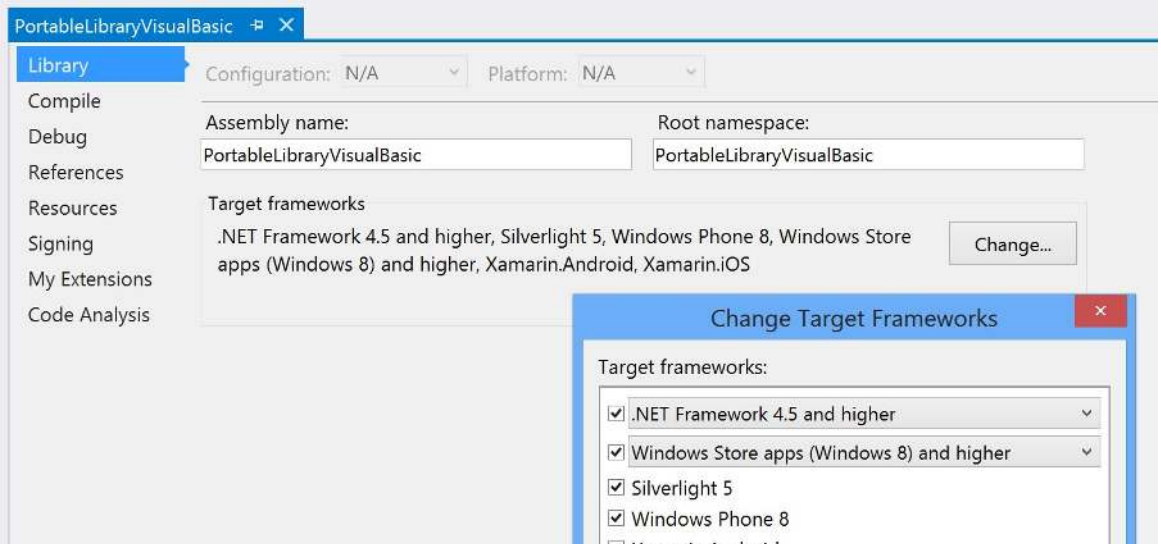
3. The Visual Basic PCL project will appear as shown in the `Solution Explorer` like this:

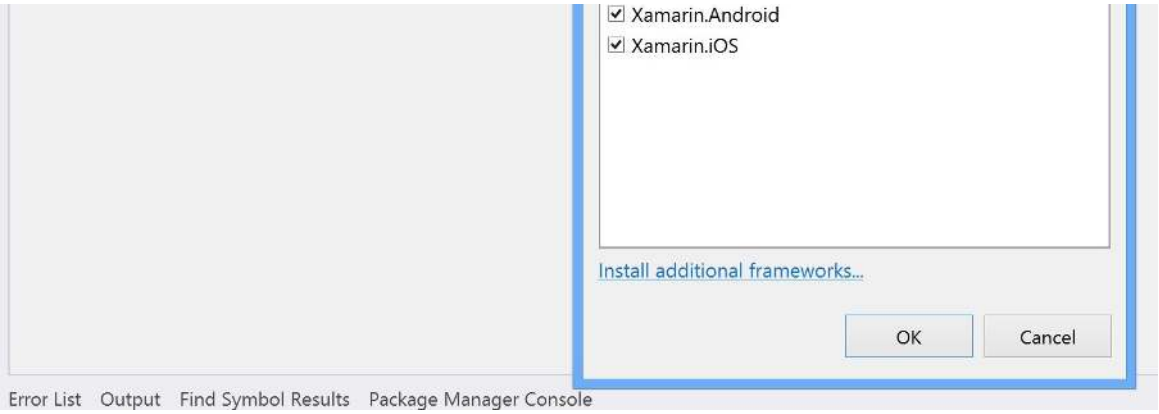


The PCL is now ready for Visual Basic code to be added. PCL projects can be referenced by other projects (Application projects, Library projects and even other PCL projects).

Editing the PCL Profile

The PCL Profile (that controls which platforms the PCL is compatible with) can be viewed and changed by right-clicking on the project and choosing `Properties > Library > Change...`. The resulting dialog is shown in this screenshot:

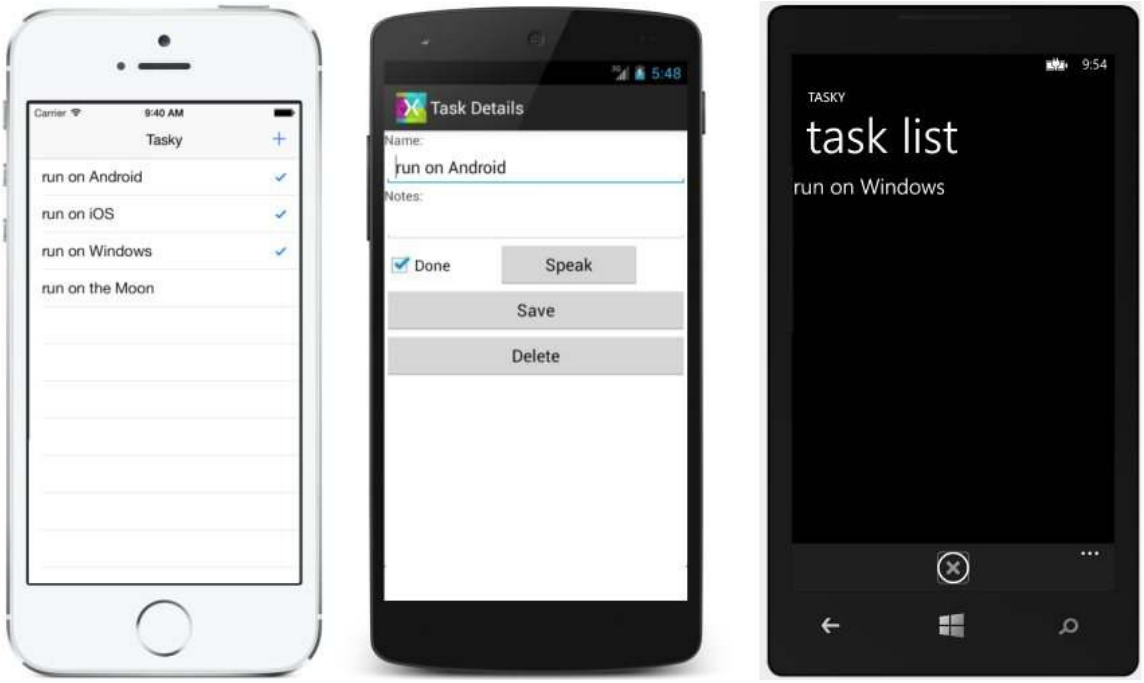




If the profile is changed after code has already been added to the PCL, it's possible that the library will no longer compile if the code references features that are not part of the newly selected profile.

Visual Basic.NET PCL Example

The TaskyPortable sample application demonstrates how Visual Basic code compiled into a Portable Class Library can be used with Xamarin. Here are some screenshots of the resulting apps running on iOS, Android and Windows Phone:



The iOS, Android and Windows Phone projects in the example are all written in C#. The user interface for each application is built with native technologies (Storyboards, Xml and Xaml respectively), while the TodoItem management is provided by the Visual Basic Portable Class Library using an `IXmlStorage` implementation provided by the native project.

VisualBasicPortableLibrary

Visual Basic Portable Class Libraries can only be created in Visual Studio. The example library contains the basics of our application in four Visual Basic files:

- IXmlStorage.vb
- TodoItem.vb
- TodoItemManager.vb
- TodoItemRepositoryXML.vb

IXmlStorage.vb

Because file access behaviors vary so greatly between platforms, Portable Class Libraries do not provide `System.IO` file storage APIs in any profile. This means that if we want to interact directly with the filesystem in our portable code, we need to call back to our native projects on each platform. By writing our Visual Basic code against a simple interface that can be implemented in C# on each platform, we can have shareable Visual Basic code that still has access to the file system.

The sample code uses this very simple interface that contains just two methods: to read and write a serialized Xml file.

```
Public Interface IXmlStorage
    Function ReadXml(filename As String) As List(Of TodoItem)
    Sub WriteXml(tasks As List(Of TodoItem), filename As String)
End Interface
```

iOS, Android and Windows Phone implementations for this interface will be shown later in the guide.

TodoItem.vb

This class contains the business object to be used throughout the application. It will be defined in Visual Basic and shared with the iOS, Android and Windows Phone projects that are written in C#.

The class definition is shown here:

```
Public Class TodoItem
    Property ID() As Integer
    Property Name() As String
    Property Notes() As String
    Property Done() As Boolean
End Class
```