PCL vs SAP

By: Mosh Hamedani

Sharing Code

We have two options for sharing code across different platforms in a Xamarin Forms app:

- **Portable Class Library** (PCL)
- **Shared Assets Project** (SAP)

The recommended approach is to use the PCL as it results in cleaner and more readable code.

Portable Class Library (PCL)

A PCL is a class library that can be ported to multiple platforms. And for this portability, it uses a subset of .NET that is available on all these platforms. More specifically, the version of .NET that is available to Android and iOS projects is different from the one used by Windows projects. You'll see this difference in the lecture called "Working with File System" in "Data Access" section.

The code in this portable class library is bundled into a dynamically-linked library (DLL) that each application project (e.g. Android, iOS, etc) references and binds to at run time. If you've built regular desktop or web applications with C#, chances are you've referenced one or more class libraries in the same solution. When you run your application, it binds to these class libraries at runtime.

Now, there are times that we need to write platform-specific code in the PCL. For example, on iOS, we often set 20 units padding on top of the page. We want the code for setting the padding to run only on an iOS device. To achieve this, we use the **Device** class in Xamarin Forms:

```
if (Device.OS == TargetPlatform.iOS) { … }
else if (Device.OS == TargetPlatform.Android) { … }
```

PCL vs SAP

By: Mosh Hamedani

Shared Assets Project (SAP)

With SAP, we don't have a separate class library, even though the Visual Studio solution gives you such an illusion at first. You may think that the folder that identifies the shared assets is a class library, but it's not. The code we write here is included with each of the application projects at build time. Now, if we want to write platform-specific code here, we need to use the old (and super ugly) 70's style C# preprocessor directives:

```
#if __IOS__
    …
#elif __ANDROID__
    …
#endif
```

Why do we have to use these preprocessor directives? Because the same code file (e.g MyPage.cs) is referenced by all application projects and is included at *build* time. Preprocessor directives select a subset of the code for compilation depending on the target platform. So, if we're compiling the Droid project, the following section will be ignored:

```
#if __IOS__
    …
```

It's the same as commenting out the code.

PCL vs SAP

So, the main difference between PCL and SAP is how we deal with platform-specific code:

- **PCL:** Using the Device class
- **SAP:** Using preprocessor directives (#if, #elif, #endif)

The only downside to PCL is that here we have access to a subset of .NET. But this is not a limitation whatsoever. First of all, the version of .NET we have here is sufficient for most scenarios. But even in situations where we need to use parts of .NET that are not available in the PCL, we can use interfaces to solve this problem. In the PCL, we declare and interface, and then implement it in each application project. In the implementation, we have access to the full .NET for that platform. You'll learn about this technique in the Data Access section.

So, this tiny limitation aside, PCL is the preferred way to share code across different platforms.