

A portrait of a man with short brown hair, smiling, wearing a grey zip-up hoodie with a small white logo on the chest. He is standing outdoors with green foliage in the background.

Xamarin Evolve 2014

Cross-Platform  
Development

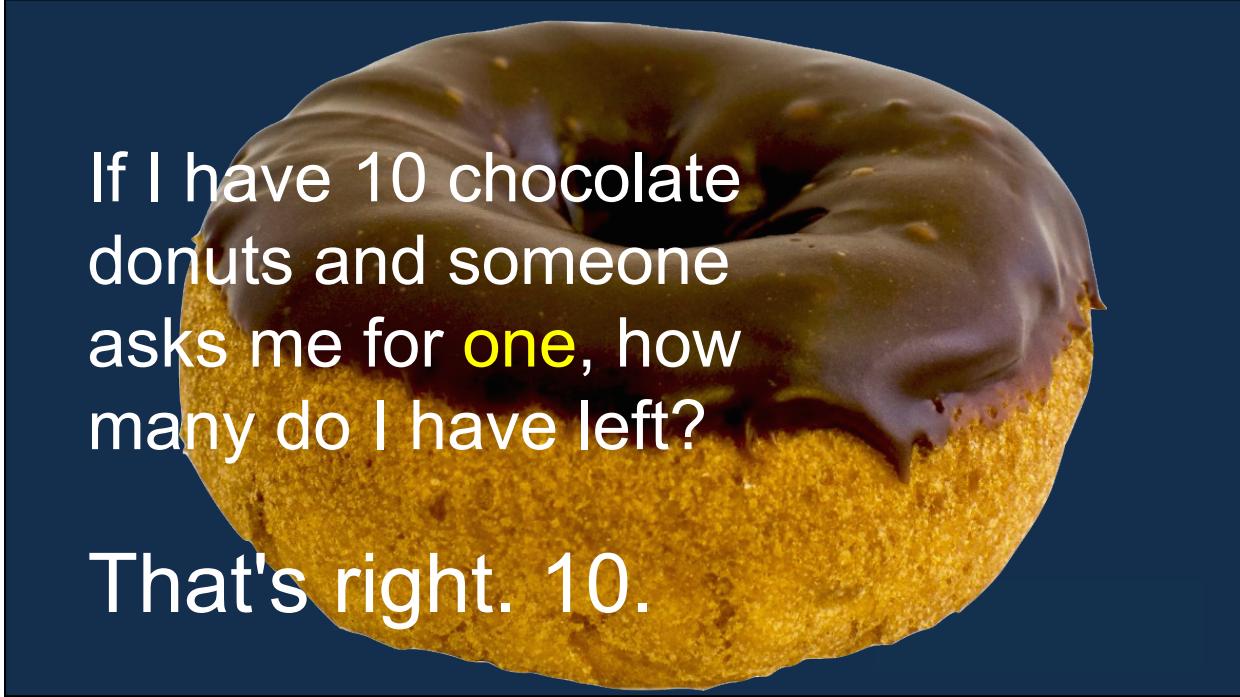
**CODE SHARING**  
**STRATEGIES**

Mark Smith  
[mark.smith@xamarin.com](mailto:mark.smith@xamarin.com)

 **Xamarin**  
University

// Sharing is good. //

Xamarin University



If I have 10 chocolate donuts and someone asks me for **one**, how many do I have left?

That's right. 10.



OK.  
Sharing **code** is good!

Xamarin University

## Agenda

1. What is Sharable Code?
2. Using Shared Projects
3. Using Portable Class Libraries



Xamarin University

## Sharable Code

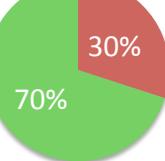
Xamarin University

## Sharable Code

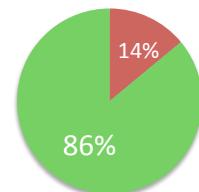
Xamarin apps are *native* and therefore will always include some platform-specific code

Xamarin University

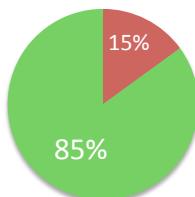
iOS



Android



Windows Phone



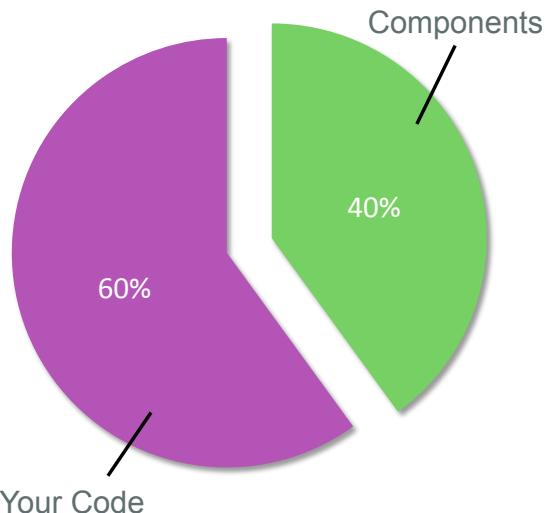
statistics taken from iCircuit

## Sharable Code

Sharable code is split between reusable components and platform-independent code

Xamarin University

## Sharable Code (average)



## NuGet

NuGet is a package manager for .NET that allows you to locate, install, update and remove shared components from your projects right in your IDE



[www.nuget.org](http://www.nuget.org)

Xamarin University

## Xamarin Component Store

Can also get reusable components from the **Xamarin Component Store** which is accessible from the IDE

A screenshot of the Xamarin Component Store website. The left sidebar shows categories like All Components, Cloud Services, User Interface, Libraries, Themes, Game Development, and Prime Components. The main area displays a list of components: Azure Mobile Services (by Microsoft, 29 ratings, Free), SAP Mobile .NET SDK (by Xamarin, 3 ratings, Prime), Signature Pad (by Xamarin, Inc., 9 ratings, Free), SQLCipher (by Zetetic LLC, 11 ratings, \$499.00+), TeeChart Charting Library (by Steema Software, 1 rating, \$339.00+), and Alliance Charts (by Alliance Global Services, 2 ratings, Free). Each component entry includes a small icon, the name, developer, rating, and price.

Xamarin University

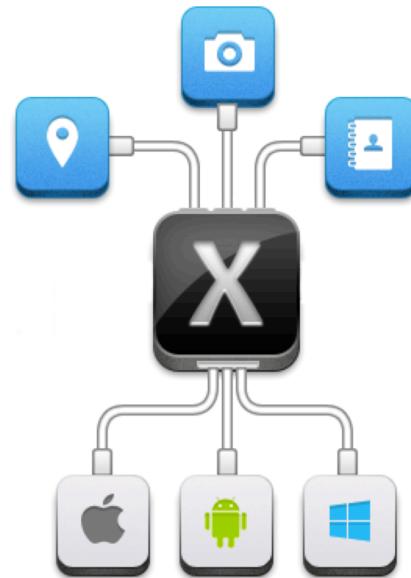
## Xamarin.\* Libraries

Open-Source, Cross-Platform APIs available from [Github.com/Xamarin](https://github.com/Xamarin)

- [Xamarin.Social](#)
- [Xamarin.Auth](#)
- [Xamarin.Mobile](#)

Check out .NET Foundation for more great open source libraries like MailKit and Rx

Xamarin University



# Demo

Xamarin University

## Agenda

1. ~~What is Sharable Code?~~
2. Using Shared Projects
3. Using Portable Class Libraries



Xamarin University

Xamarin University

## Xamarin Forms Project Styles

Two project styles are available for Xamarin Forms

Shared Project

Portable Class  
Library

# Shared Projects

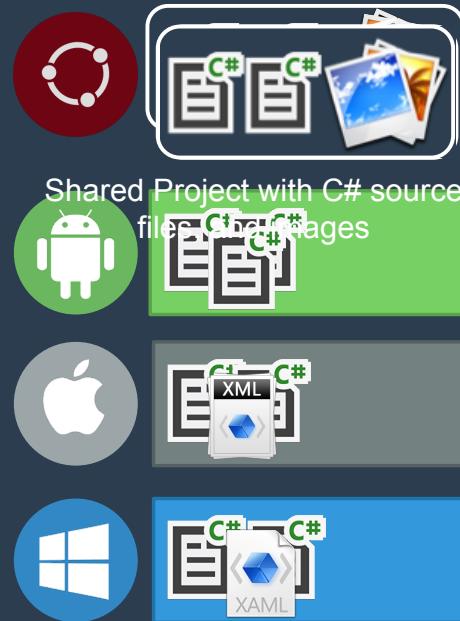
Xamarin University

## What is a Shared Project?

*Shared Projects* enables project-level sharing of **source + assets**

- Single copy of source file (shared)
- Compiled uniquely into project
- Normal refactor + navigation works

Xamarin University



# Shared Project packaging

Project defined by new .shproj type



This project defines the included files as well as the build type (Compile, None, etc.), but does not actually generate any output

no assembly is produced

# Shared Projects Internals

Adding a reference to a shared project adds all the files to the target during the compile process, source files are **compiled for the target**



# Demo

Xamarin University

What if I need platform-specific features?

Conditional  
Compilation

Class  
Mirroring

Partial  
Classes +  
Methods

Xamarin University

# Demo

Xamarin University

## Agenda

- 1. ~~What is Sharable Code?~~
- 2. ~~Using Shared Projects~~
- 3. Using Portable Class Libraries



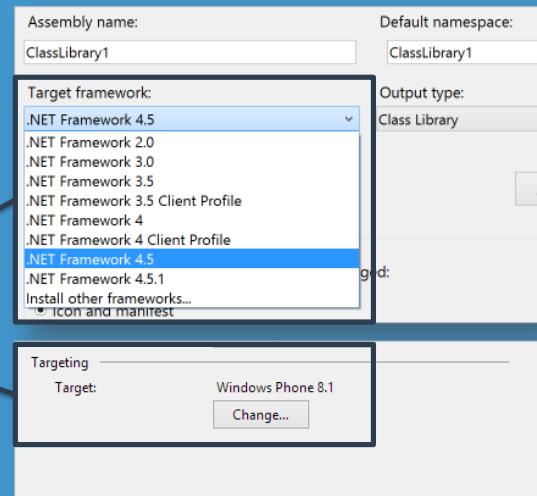
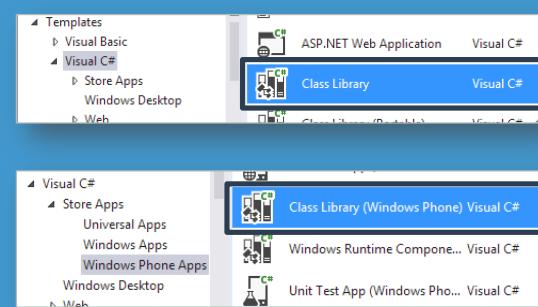
Xamarin University

# Portable Class Libraries

Xamarin University

## Reminder: Class Libraries

Class Library projects are tied to a specific platform + framework



Xamarin University

Xamarin University

## What is a Portable Class Library?

“Portable class libraries are made up of two concepts: an API contract (AKA: reference assembly) and an implementation. Most of the time, those concepts exist together in a single assembly. PCLs, however, have the flexibility to have multiple implementations for a single API contact. This 2<sup>nd</sup> case is useful, when implementations need to be platform-specific in some way (ex: x86 vs. ARM or Android vs. iOS). In this case, a PCL would have a single reference assembly that consumers compile with, and then implementations available to be deployed for each of the supported platforms. In all cases, PCLs expose a surface area that is compatible with multiple .NET platforms.”

– formal definition from Microsoft



## What is a Portable Class Library?

Portable Class Libraries (PCLs) are assemblies that can be used by different flavors of .NET without recompiling

The 'Add Portable Class Library' dialog shows the following targets selected:

- .NET Framework 4.5
- Windows 8
- Windows Phone Silverlight 8
- Windows Phone 8.1
- Xamarin.Android
- Xamarin.iOS

**Xamarin University**

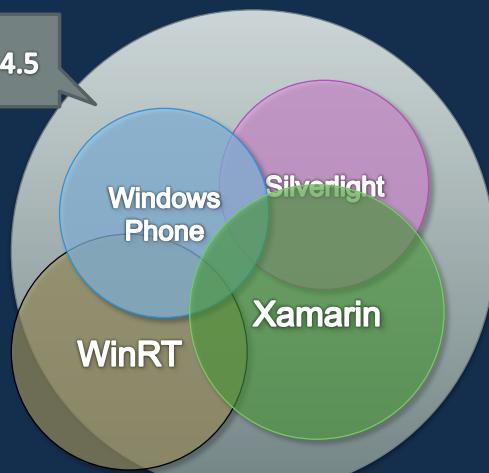
## How does it work?

A PCL is tied to a specific *profile* which defines the specific APIs it can utilize

Feature	.NET Framework	Windows Store	Silverlight	Windows Phone (SL)	Windows Phone (Store)	Xamarin
Core Libraries	✓	✓	✓	✓	✓	✓
LINQ	✓	✓	✓	✓	✓	✓
IQueryable	✓	✓	✓	7.5+	✓	✓
Compression	4.5+	✓	✗	✗	✓	✓
Data Annotations	4.0.3+	✓	✓	✗	✗	✓
System.IO.File	✗	✗	✗	✗	✗	✗

## Configuring Portable Class Libraries

.NET 4.5



You select the platforms the library will be used on – this decides the profile

The available combinations are controlled by the profiles Microsoft has defined

The *more platforms* you choose, the *less APIs* you will be able to use

.NET 4.5

Pick only the framework targets you *need right now* to give you as broad API reach as possible

You select the platforms the library will be used on – this decides the profile

The available combinations are controlled by the profiles Microsoft has defined

The *more platforms* you choose, the *less APIs* you will be able to use

## Adding References to PCLs

Any references, components or Nuget packages you add to your project must be compatible with the platforms you have selected or an error will occur

Demo

## Problems with PCLs

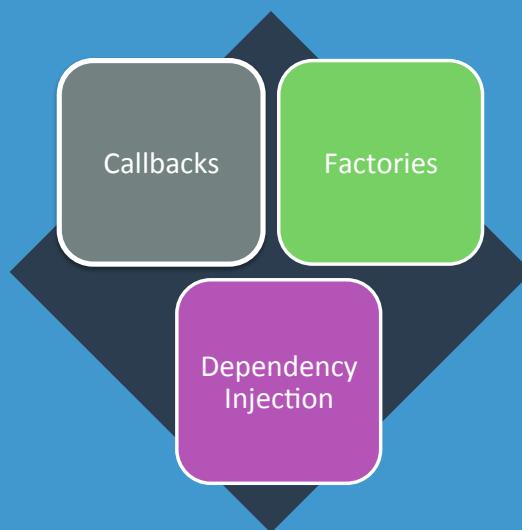
PCLs are restricted to the APIs available on all the target platforms

```
void LoadNotes(string filename) {  
    var reader = new StreamReader(filename);  
    ...  
}
```

Selected profile has no  
constructor on **StreamReader**  
which takes a string

```
public StreamReader(  
    System.IO.Stream stream  
)  
Parameter  
stream: The stream to be read.  
  
Summary  
Initializes a new instance of the StreamReader class  
for the specified stream.
```

## What if I need platform-specific features?



# Demo

Xamarin University

## Shared Project vs. PCL .. FIGHT!

### Shared Projects

PROS	CONS
All APIs available	Can lead to spaghetti code
Platform-specific logic can be added directly	Cannot be unit tested on it's own
All file types can be shared	Must be shipped in source form

### Portable Class Libraries

PROS	CONS
Enforces architectural design	Limited APIs available
Can be unit tested separately	Can be difficult to share non-code files
Can be shipped in binary form (Nuget)	More work to integrate platform-specific code

## Agenda

1. ~~What is Sharable Code?~~
2. ~~Using Shared Projects~~
3. ~~Using Portable Class Libraries~~



Xamarin University

Questions?

Xamarin University



Xamarin Evolve 2014

# Cross Platform Development

Mark Smith  
[mark.smith@xamarin.com](mailto:mark.smith@xamarin.com)

 **Xamarin**  
University