Introduction to Crosslight

Crosslight is an advanced cross-platform mobile development toolset that makes enterprise mobile apps development truly a breeze. Leveraging extensible architecture, MVVM design pattern and integration with Xamarin Platforms, Crosslight lets you easily build powerful iOS, Android and Windows native apps with a common application codebase including domain model, data access, and user interaction logic. And that's nearly 96% of your project's codebase. Built with cutting-edge portable framework and MVVM pattern, Crosslight enables you to build cross-platform mobile apps by leveraging the programming skills and tools you loved – such as C#, MVVM, .NET and Visual Studio.

Watch the Crosslight webinar video below to learn about Crosslight.

If you're new to cross-platform mobile development with Crosslight, start with Getting Started with Crosslight.

To begin building right away, read Walkthrough: Creating Your First Crosslight Application.

To learn what's new in the latest Crosslight release, see Crosslight 2.0 Release Notes.

Why Crosslight

The cross platform mobile development in general exhibits a number of challenges due to the divergence of the mobile platforms. Some of the key challenges are:

- Targets multiple platforms at once (efficiency)
- High code reusability (maintainability)
- Great native user experiences (usability)
- High extensibility (customizability)
- Loosely-coupled components (testability, scalability)

Crosslight was built to address all these challenges, and more importantly, empowering developers to build great enterprise cross-platform apps in dramatically less code and less time.

In addition, Crosslight breaks the barriers commonly found in native cross-platform development such as:

- Difference in the development language.
  iOS apps are mainly written in Objective-C, while Android are in Java, and Windows Phone are in C#.

- Difference in the platform API and operating system
  Since each mobile operating system was designed by different vendor, none of them share common programming API. As the results, apps that have been built for one platform cannot be easily ported to another platform, and mostly will require major code rewrite.

- Difference in the programming design pattern
  Each platform introduces their own programming design pattern that suits to the respective vendor's technology portfolios. For examples, Apple leverages MVC design pattern for both iOS and Mac development, Android is based on more traditional MVP design pattern, while Microsoft built its Windows Phone and Windows RT around MVVM design pattern.

With Crosslight, you can now rapidly build powerful cross-platform mobile apps without worrying the above challenges – thanks to the rock-solid architecture built from the ground-up and integration with latest technology such as Xamarin and Microsoft platforms.

In essence, Crosslight takes out the user interaction logic from each platform-specific code and streamline them to a single codebase in a ViewModel class. As the results, the platform-specific layer no longer contains user interaction logic such as event handling, navigation, input prompt, and much more.

See the simple Hello World code sample below for an overview of Crosslight design pattern.

iOS

```
[ImportBinding(typeof(HelloWorldBindingProvider))]
public class MyViewController : UIViewController<HelloWorldViewModel>
{
}
```

Android

```
[Activity(Label = "AndroidApplication1", MainLauncher = true, Icon =
"@drawable/icon")]
[ImportBinding(typeof(HelloWorldBindingProvider))]
public class Activity1 : Activity<HelloWorldViewModel>
{
}
```

Windows

```
[ImportBinding(typeof(HelloWorldBindingProvider))]
[ViewModelType(typeof(HelloWorldViewModel))]
public partial class MainPage : PhoneApplicationPage
{
}
```

ViewModel in C#

```
public class HelloWorldViewModel : ViewModelBase
{
 public HelloWorldViewModel()
 {
  this.HelloCommand = new DelegateCommand(this.ExecuteHelloCommand);
 }

 public DelegateCommand HelloCommand { get; set; }

 private void ExecuteHelloCommand(object parameter)
 {
  this.MessagePresenter.Show("Hello world from Crosslight mobile app", "My App");
 }
}
```

Compare with the code in platform-specific API

iOS

```
public class MyViewController : UIViewController
{
 public override void ViewDidLoad()
 {
  base.ViewDidLoad();
  ...
  button.SetTitle("Click me", UIControlState.Normal);
  button.TouchUpInside += (object sender, EventArgs e) =>
  {
   UIAlertView alertView = new UIAlertView();
   alertView.Title = "My MobileApp";
   alertView.Message = "Hello world from my mobile app";
   alertView.AddButton("OK");
   alertView.Show();
  };
 }
}
```

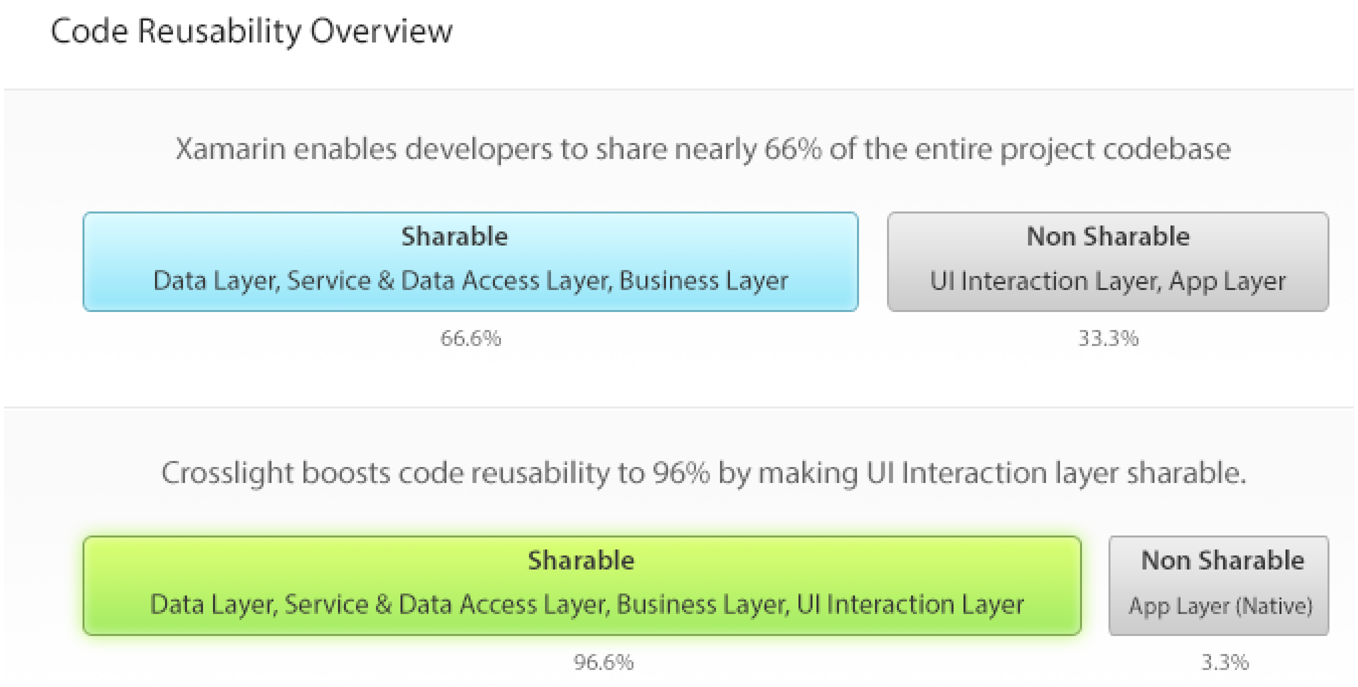
Android

```
[Activity(Label = "AndroidApplication1", MainLauncher = true, Icon =
"@drawable/icon")]
public class Activity1 : Activity
{
 protected override void OnCreate(Bundle bundle)
 {
  base.OnCreate(bundle);
  ...
  Button button = FindViewById<button>(Resource.Id.MyButton);
  button.Click += delegate
  {
   AlertDialog.Builder dlgAlert = new AlertDialog.Builder(this);
   dlgAlert.SetMessage("Hello world from mobile app");
   dlgAlert.SetTitle("My MobileApp");
   dlgAlert.SetPositiveButton("OK", (o, e) => { });
   dlgAlert.SetCancelable(true); dlgAlert.Create().Show();
  };
 }
}
```

Windows

```
public partial class MainPage : PhoneApplicationPage
{
...
 private void MainPage_Loaded(object sender, RoutedEventArgs e)
 {
  this.MyButton.Click += (object s, RoutedEventArgs e2) =>
  {
   MessageBox.Show("Hello world from mobile app");
  };
 }
}
```

The above code should look familiar to Silverlight or WPF developers. It leverages the common MVVM design pattern such as delegate command and view presenter. This allows you to get started quickly with Crosslight application development in minimal learning curves. With the application logic now sharable, it boosted the overall code reusability to nearly 96%, leaving the small percentage on the app-specific layers such as views and controllers.


Code Reusability Overview

Xamarin enables developers to share nearly 66% of the entire project codebase

Sharable
Data Layer, Service & Data Access Layer, Business Layer
66.6%
Non Sharable
UI Interaction Layer, App Layer
33.3%

Crosslight boosts code reusability to 96% by making UI Interaction layer sharable.

Sharable
Data Layer, Service & Data Access Layer, Business Layer, UI Interaction Layer
96.6%
Non Sharable
App Layer (Native)
3.3%

With a host of great capabilities and time-saving features, Crosslight delivers numerous benefits not available in other solutions such as:

- Reuse nearly 96% of codebase
- Rapid cross-platform mobile development with tools you already accustomed to – slash development time in half
- High maintainability and scalability – thanks to the shareable user interaction and application logic code
- Future-proof architecture with service abstraction and container built into the core framework
- Unified application logic significantly increase developer productivity and enable team collaboration
- Slashed development time means reduced total cost of ownership and faster time to market
- Built on portable framework, allowing the same codebase to be reused in the future platforms without major code rewrite

Note that Xamarin platforms are licensed separately from Crosslight. Please see the Licensing section for more details.

Key Features

At its heart, Crosslight features a powerful application framework that is designed to run across multiple mobile platforms namely iOS, Android, Windows Phone and Windows RT. In addition, it also ships with numerous pre-built UI components and platform-specific controllers that offer out-of-the-box functionalities which can be easily customized through simple property sets.

The following list describes the Crosslight key features:

- Comprehensive and advanced mobile frameworks leveraging MVVM design pattern
- Build native iOS, Android, Windows Phone 8 and Windows 8 apps with a single application codebase
- Elegant, developer-friendly API based on platform standards
- Universal data management with automatic binding
- Streamlined navigation services supporting push, modal and nested navigation mode
- Rich form builder with 20+ pre-built editors
- Highly customizable editor controls ranging from auto resize textbox to image picker with camera support and more
- Comprehensive mobile services for business apps
- Powerful enterprise mobile application framework featuring advanced data access services and authentication services
- Included enterprise components such as offline and synchronization services, async image loader, social services, and much more
- Native user experiences conforming to platform design guidelines
- Support Visual Studio 2012/2013 and Xamarin Studio
- Integrated to Xamarin platforms with full AOT compliance
- Time-saving Project Wizard featuring 30+ templates variants supporting iOS, Android, Windows Phone 8 and Windows 8

The Crosslight Features page elaborates in depth on what Crosslight has to offer, changing the way you develop mobile apps once and for all.

Related Topics

- Walkthroughs and How-to Topics
- Next Steps
- System Requirements
- Updating Crosslight
- Crosslight Starter Guide
- Getting Started with Crosslight