



REAL WORLD
TECHNOLOGY TRAINING & SOLUTIONS
"Training You Can Really Use"

Building Cross-Platform Apps with Xamarin, Visual Studio, and C#

Duration: 5 Days

Method: Instructor-Led Training (ILT) | Live Online Training

Course Description

This course has been designed to help participants gain maximum exposure to this platform so that they can learn its installation, configuration, controls, and web services. Participants will learn how to use Xamarin Studio and build mobile applications using C#. They will be able to write code to access the database and application resources, and the basics of testing and publishing.

Target Audience

This course is intended for:

- Persons who are looking to use the Xamarin tools and framework to design, develop and deploy enterprise applications across a variety of platforms such as:
 - Mobile Developers and Programmers
 - App Developers

Prerequisites

To attend this course, candidates must have:

- Good understanding of C#/.NET Programming concepts
 - Good understanding of Object-Oriented Programming System (OOPS) concepts.
- NOTE:** Basic Knowledge of OS X usage and navigation would be an asset.

Course Objectives

Upon successful completion of this course, attendees will be able to:

- Build native mobile apps for iOS and Android platforms and take their app development strategy to the next level.
- Write simple and concise cross-platform UI code and run the applications.
- Understand XAML, data binding and controls, and how to use them effectively within their Xamarin.Forms applications.
- Understand the data access patterns for Xamarin's iOS and Android platforms.



Course Topics

Module 1: Architecture and Setup

- .NET on Android/iOS – What is it?
- Bringing .NET to Android/iOS
- Android Tool Experience
- iOS Tool Experience
- Development Philosophy and Solution Organization

Module 2: Creating the Android App

- Create the Android Project
- Layout the User Interface
- Tie User Interface to Code
- Run the App on the Emulator
- Add Image to Layout
- Add Image Handling Code
- Run the App with Image Handling

Module 3: Preparing for iOS Development

- The iOS Development Experience
- Installation Requirements
- Installing Xamarin on OS X
- Visual Studio/Xamarin.iOS Connectivity
- Configuring Connectivity
- Verifying Setup

Module 4: Creating the iOS App

- Add the iOS Project
- Add UI Designer Support
- Add Controls to the UI
- Add Properties for the Controls
- Tie the User Interface to Code
- Run the Project
- Add Image to Layout
- Add Images and Image Handling Code
- Run the App with Image Handling

Module 5: Creating the Shared Code Library

- Creating a Portable Class Library
- Add Basic Data Representation
- Add Data Management
- Create Cursor-Like Behaviour
- Connect to Android User Interface
- Connect to iOS User Interface
- Providing Visual Feedback from the Library

Module 6: Cross-Platform Image Management

- Shared Image Management in the iOS App
- Quick-and-Dirty Shared Image Management in the Android App
- Translating Image Names to Resource Identifiers with Reflection
- Reducing Image Translation Overhead

Module 7: Introduction to Navigation and Platform Capabilities

- Swipe-Based Paging
- Multi-Screen Navigation
- Embrace Platform Capabilities and Common Behaviours

Module 8: Swipe-Navigation in Android

- Direct Navigation
- Moving to Swipe-Navigation
- Add Random Access to Course Library
- Create the Course Fragment
- Support Package and Namespace
- Add Support Library as a Xamarin Component
- Implement
FragmentStatePagerAdapter
- Create the New Activity Layout
- Create the Swipe-Navigation Activity
- Getting a Deeper Understanding





Course Topics *Continued*

Module 9: Swipe-Navigation in iOS

- Moving from Direct to Swipe Navigation
- Adding the UINavigationController
- Creating the per Course UINavigationController
- Displaying the first Course Page
- Adding Navigation Events
- Adding Position Awareness to Course UINavigationController
- Handling Navigation Events
- Adding Page Turning Effects

Module 10: Working with iOS Protocols

- What are the Protocols?
- Xamarin Sometimes Hides Protocol Use
- Swipe Navigation with Delegates vs. Protocol
- The Challenge of Protocols in C#
- Why are we Talking About This Now?
- Implementing Protocols in Xamarin
- The Implication of Protocols as Classes
- Implementing Course Swipe Navigation w/ Protocol

Module 11: Android Master/Detail Navigation

- Android Master/Detail Experience
- Creating a Master List Activity
- Working with an Array Adapter
- Why an Array Adapter Is Not the Best Choice?
- Creating a Custom List Adapter
- Implementing List Adapter GetView
- Connecting the Custom List Adapter

- Starting the Detail Activity
- Passing Intent Extras

Module 12: Android Navigation Drawer

- Overview of the Android Navigation Drawer
- Activity Architecture with Navigation Drawer
- Adding Navigation Drawer to the Layout
- Managing Master and Detail Data Together
- Populating the Navigation Drawer Choices
- Customizing the Navigation Drawer List Appearance
- Managing List Selections in the Navigation Drawer
- Making the
FragmentStatePagerAdapter Updatable
- Updating the Displayed Details

Module 13: iOS Master/Detail Navigation

- iOS Master/Detail Experience
- iOS Master/Detail Architecture
- iOS UI Options
- Creating a Master Table View Controller
- Creating a Table View Data Source
- Connecting Table View and Data Source
- Adding a Navigation Controller
- Displaying the Detail View Controller
- Passing the Master Selection to the Detail View Controller
- Fixing Project Version Issues
- Adjust Detail View Controller Layout

LABS INCLUDED