

# Handling Platform Specific Requirements



Jim Wilson

@hedgehogjim| [blog.jwhh.com](http://blog.jwhh.com)

# What to Expect in This Module



The role of the Device class

Handling platform details with  
DependencyService

Managing loosely coupled behavior with  
MessageCenter

# Device Class

Performs simple per-platform operations or value retrievals



Provides platform agnostic access to certain simple but common operations

# Retrieve Different Value for Each Platform

```
Padding = new Thickness(  
    20,  
    Device.OnPlatform(40, 20, 0),  
    10,  
    20);
```

Android

iOS

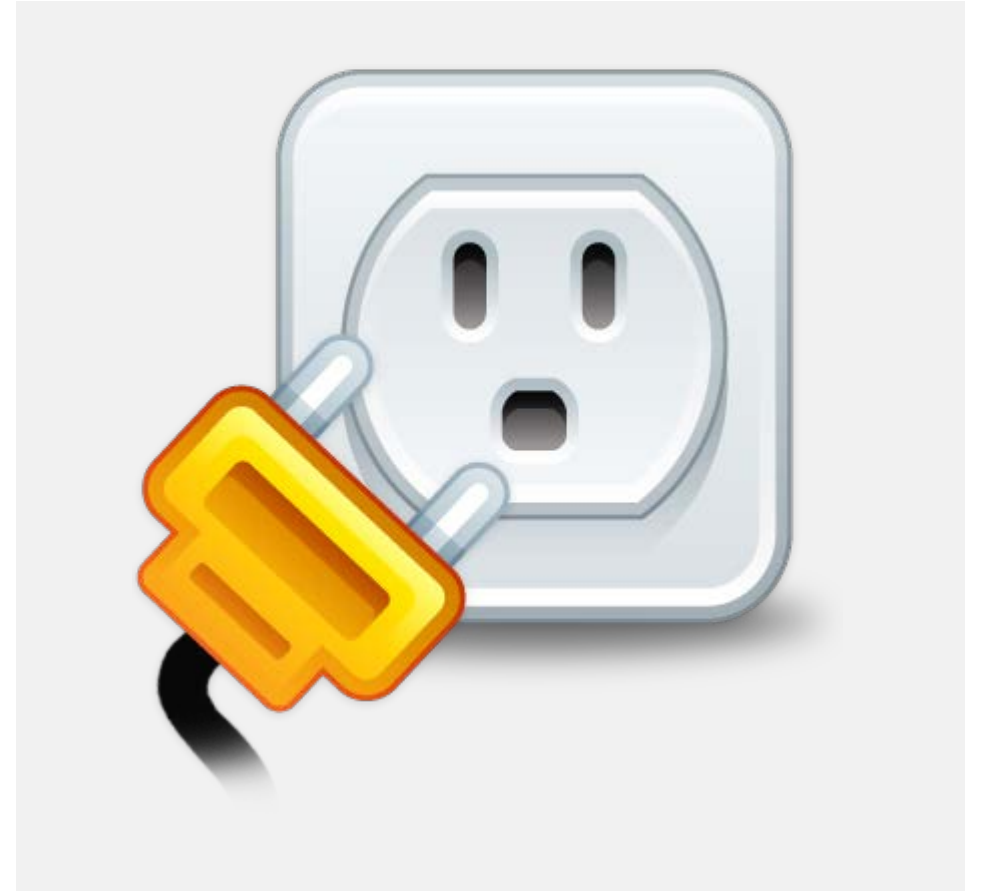
Windows  
Phone

# Execute Different Method for Each Platform

```
var deviceLabel = new Label();  
Device.OnPlatform(  
    iOS: () => {  
        deviceLabel.Text = "iOS";  
        deviceLabel.TextColor = Color.Blue;},  
    Android: () => {  
        deviceLabel.Text = "Android";  
        deviceLabel.TextColor = Color.Green;},  
    WinPhone: () => {  
        deviceLabel.Text = "Windows";  
        deviceLabel.TextColor = Color.Purple;});
```

# Platform Agnostic Access

- `BeginInvokeOnMainThread(Action)`
  - Execute action on UI thread
- `StartTimer(TimeSpan, Func<bool>)`
  - Execute function on a regular interval until function returns false
- `OpenUri(Uri)`
  - Use platform features to open Uri



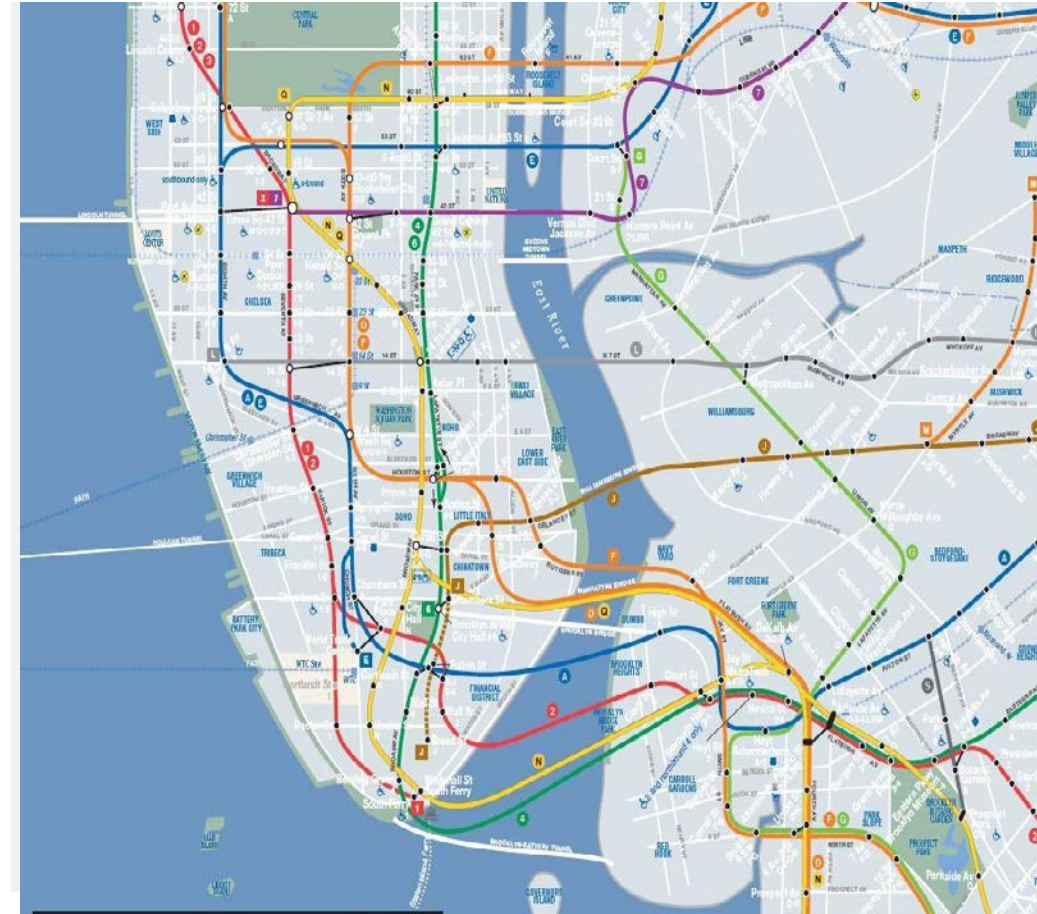
# DependencyService

- Resolve custom platform-specific implementations at runtime
  - Model behavior as an interface
  - Provide implementation for each platform
  - Request interface implementation from shared code
    - DependencyService loads the current platform's implementation



# MessageCenter

- Provides a loosely coupled message exchange
  - Interested receivers subscribe
  - Components can send messages at anytime
    - Can optionally include an argument
  - MessageCenter handles routing details
- Identifying messages
  - Messages are grouped by type
  - Individual messages qualified by string ID





# Summary

- Device class handles many common platform specific issues
  - Retrieve per-platform values
  - Execute a different action for each platform
  - Platform agnostic access to certain simple but common operations
- DependencyService is a facility for providing platform specific implementations and resolving them at runtime
- MessageCenter provides a loosely coupled message exchange

# Where to Go from Here



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

Part 1

<http://bit.ly/AndroidiOSCSharp1>

Part 2

<http://bit.ly/AndroidiOSCSharp2>

# Where to Go from Here



Android for .NET Developers series

Part1: <http://bit.ly/PSAndroidStart>

Part2: <http://bit.ly/PSBuildAndroidApps>

Part3: <http://bit.ly/PSAdoptAndroidMindset>

Part4: <http://bit.ly/PSAndroidUnderstanding>

Invest the time to learn Android

<http://pluralsight.com/training/Courses#android>

# Where to Go from Here



iOS for .NET Developers

<http://bit.ly/1hitNfJ>

iOS 7 Fundamentals

<http://bit.ly/1ebmbM5>

Invest the time to learn iOS

<http://pluralsight.com/training/Courses#ios>