

```
module Configuration
```

```
    #load "ContinuousDeliveryMobile.fsx"
```

```
    open System.IO
```

```
    open ContinuousDeliveryMobile
```

```
    open System.Collections.Generic;
```

```
    [<Sealed>]
```

```
    type Configuration () =
```

```
        let solution = System.IO.Path.Combine("GroceryShopper", "GroceryShopper.sln")
```

```
        let project path = System.IO.Path.Combine("GroceryShopper", path, path + ".csproj")
```

```
        let androidApp config = App(Android, solution, project "GroceryShopper.Forms.Droid", BuildConfiguration(config, "Any CPU"))
```

```
        let iOSApp config = App(iOS, solution, project "GroceryShopper.Forms.iOS", BuildConfiguration(config, "iPhoneSimulator"))
```

```
        let unitTests = UnitTest("Grocery.UnitTests.csproj", BuildConfiguration("Test", "Any CPU"), "Grocery.UnitTests.dll", "testresults.xml")
```

```
        let uiTests app testresults = UITest(app, UnitTest("Grocery.UnitTests.csproj", BuildConfiguration("UITest", "Any CPU"),  
            "Grocery.UnitTests.dll", testresults))
```

```
    interface IConfiguration with
```

```
        member this.Build = [|  
            androidApp "Debug";  
            iOSApp "Debug"  
        |]
```

```
        member this.UnitTest = [|  
            unitTests  
        |]
```

```
        member this.UITest = [|  
            uiTests (androidApp "Release") "android_uitestresults.xml"  
        |]
```

```
        // Use the "Skip" helper to indicate that a specific target should not be executed
```

```
        member this.Package = Skip
```

```
        member this.Publish = Skip
```

```

interface ITechnicalConfiguration with

    member this.AndroidKeystoreParams app = AndroidKeystoreParams("", "", "")

    member this.EmulatorPath =
        let dictionary = new Dictionary<BuildPlatform, string>()
        dictionary.[Windows] <- @"C:\EclipseADT\sdk\tools\emulator.exe"
        dictionary.[OSX] <- "/Users/Dev/Library/Developer/Xamarin/android-sdk-mac_x86/tools/emulator"
        dictionary

    member this.AdbPath =
        let dictionary = new Dictionary<BuildPlatform, string>()
        dictionary.[Windows] <- @"C:\EclipseADT\sdk\platform-tools\adb.exe"
        dictionary.[OSX] <- "/Users/Dev/Library/Developer/Xamarin/android-sdk-mac_x86/platform-tools/adb"
        dictionary

    member this.NUnitPath =
        let dictionary = new Dictionary<BuildPlatform, string>()
        dictionary.[Windows] <- Path.Combine(Directory.GetCurrentDirectory(), "..", "Tools", "NUnit-2.6.4", "bin", "nunit-console.exe")
        dictionary.[OSX] <- "/Library/Frameworks/Mono.framework/Versions/Current/bin/nunit-console4"
        dictionary

    // You can override the CDM default implementations of the Targets by returning another
    // implementation of the "ITargetImplementations" interface here
    member this.GetTargetImplementations defaultTargetImplementations = defaultTargetImplementations

```