## Overview of Talk

### What is Mono for Android

### Dispel some myths

### Show off the chrome

## 5Ws + 1 H

Who

### What

### Where

### When

### Why

### How

## tl;dr

### Who: Novell

### What: SDK, Runtime, and Tools enabling .NET

### Where: <http://mono-android.net/>

### When: NOW!

### Why: Well….it depends. Let’s go with b/c it is awesome right now.

### How: By running a CLR SxS Dalvik in the same process

## Who

### Novell

### More specifically the Mono guys who work for Novell

## What is Mono for Android?

### Compiler, SDK, Tools

### Works on

#### Windows (Visual Studio)

#### Mac (MonoDevelop)

### C# 4.0

### Mono 2.10 profile

### Interoperability

#### 57 Android.\* namespaces

#### 31 Java.\* namespaces

## Where

### <http://mono-android.net>

### <https://github.com/mono/monodroid-samples>

### support on #IRC and StackOverflow by the team

## When

### Just released Wedsnesday

### Commercial Product

#### $99/$399/$999/$3999

#### Go-Live License

#### Deployable to:

##### Google Marketplace

##### Amazon Appstore

##### Ad-Hoc

## Why

### Benefits

#### Cross platform mobile dev + plus non-mobile

##### Non-GUI code reuse

##### UI stuff is so different from iOS-to-Android-to-WP7

##### Common language across platforms

##### Have to learn each platform but not a new language each time.

##### Spend more time making your app look good and usable

##### More users in aggregate using the code == more testers

#### Language Features

##### Unencumbered CLR

###### Allows JIT’ing which iOS blocks and WP7 allows but has limitations

##### Linq

##### Lambas

##### Parallel

##### F#

##### Dynamic languages

###### IronPython

###### IronRuby

###### IronJS

##### C# compiler as a service

#### Mono Ecosystem

##### MonoTouch

##### MonoMac

##### Amazing support

##### Great community

### Tools

#### Investment in Visual Studio

### Myths

#### For lazy devs

#### Not native

#### Not fast

#### Not easy to use

#### No reason to use it vs. native

## So How does this work?

### There is a CLR

#### Installed as a shared run-time OR bundled with the app

### Runs along side the Dalvik VM in the same process

#### Each activity runs in it’s own process and gets its own Dalvik VM. MfA just adds a CLR

### Program Execution

#### Bootstrapped via native program

#### Starts CLR and begins execution of App

### Talk between CLR and Dalvik

#### Need to talk to Dalvik for device interaction

#### Interoperability

##### 57 Android.\* namespaces

##### 31 Java.\* namespaces

#### OpenTK

##### Abstraction over OpenGL

##### Portability for your graphics code

### Compilation and Linking

#### MfA strips the crap out

##### Pay for what you use

## Examples

### Events

### Web service

### Linq

### Lambdas

### Shared libs

### MonoDroid.Dialog

## Resources

### Links

#### [http://mono-android.net/Welcome](http://monodroid.net/Welcome)

#### [http://www.mono-android.net](http://github.com/mono/monodroid-samples)

#### [http://developer.android.com](http://developer.android.com/)

### IRC

#### #monodroid  on irc.gnome.org

### Code

#### <https://github.com/mono/monodroid-samples>

#### <https://github.com/kevinmcmahon/MonoDroid101>