XAM220

Preparing to publish your application

Download class materials from
university.xamarin.com

Microsoft

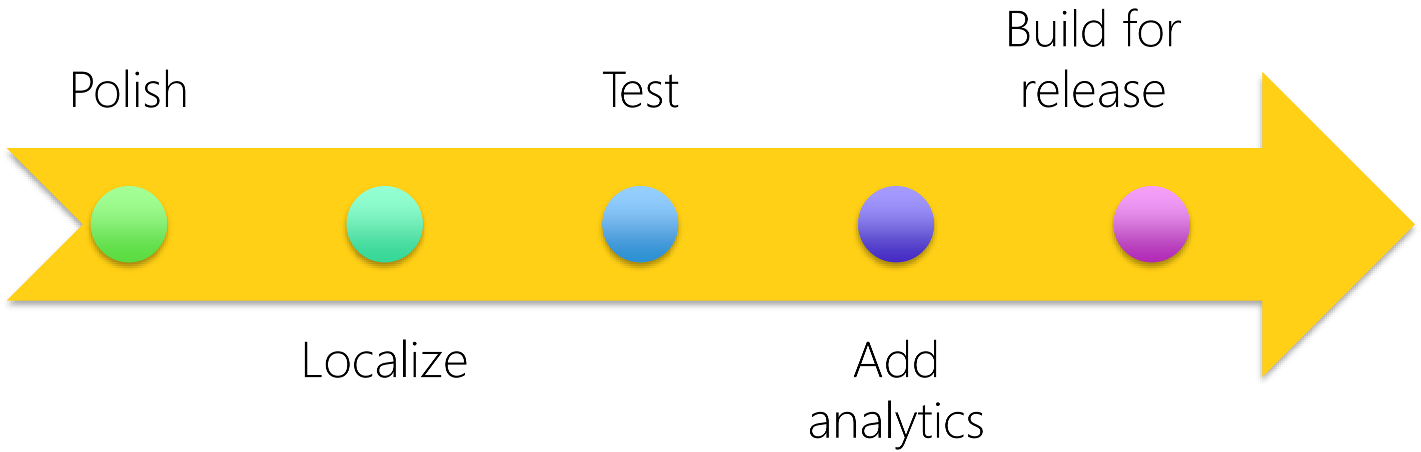
Xamarin University

Tasks

1. Update your app for publishing
2. Choose a distribution strategy
3. Publish to a store

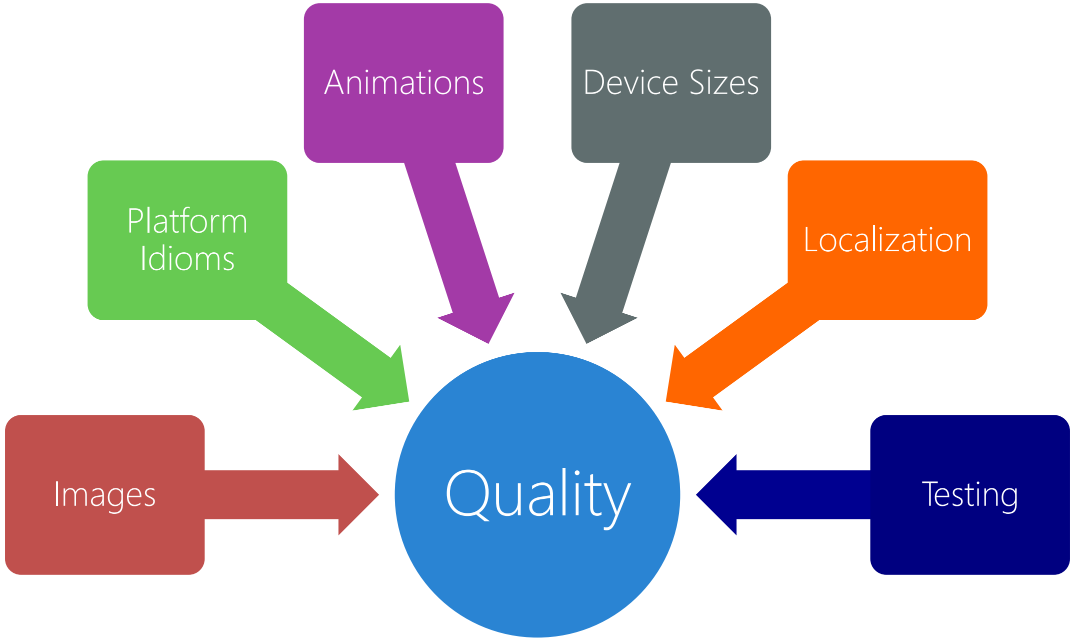
Xamarin University

Get ready to publish!



Polish your app

- Pay attention to the details in your application



Localize your app

- Millions of users cannot read the language your app is written in
- **56.2%** of consumers say that the ability to obtain information in their own language is more important than price

Test your app completely

- Your app will be automatically rejected if it crashes or misbehaves

Dropped Network

Older devices

Bad input

Orientation Changes

Xamarin University

Automated UI Testing

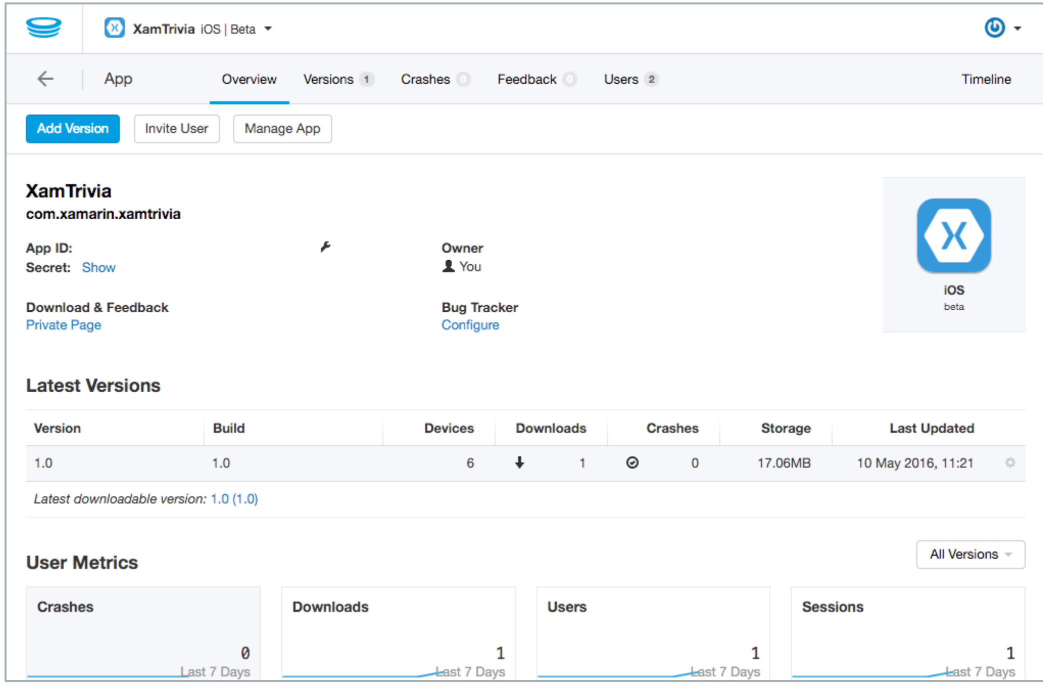
- Xamarin UI Test lets you create automated UI tests that can be run locally or in the cloud

Xamarin Test Cloud

Visual Studio Mobile Center

Consider adding analytics

- Invest in an analytics solution such as HockeyApp or Mobile Center
  - Identify crashes and live issues
  - Invite beta users and push updates to users
  - identify features people are using



XamTrivia
com.xamarin.xamtrivia

App ID:
Secret: Show

Owner
You

Download & Feedback
Private Page

Bug Tracker
Configure

iOS
beta

Latest Versions

Version	Build	Devices	Downloads	Crashes	Storage	Last Updated
1.0	1.0	6	1	0	17.06MB	10 May 2016, 11:21

Latest downloadable version: 1.0 (1.0)

User Metrics

All Versions

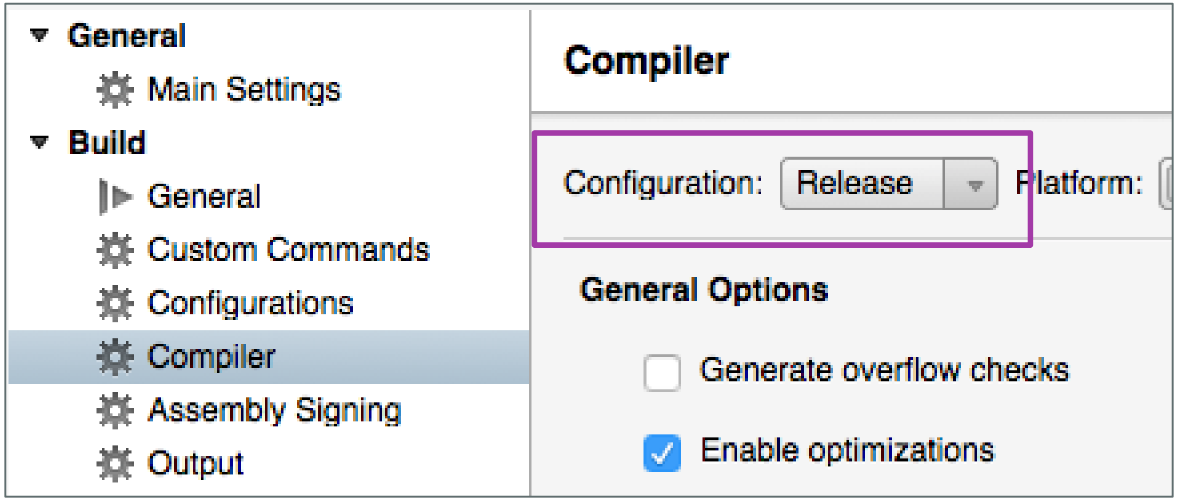
Crashes	Downloads	Users	Sessions
0	1	1	1
Last 7 Days	Last 7 Days	Last 7 Days	Last 7 Days

Preparing for release

1. Create a **Release Build**
2. Add icons and splash screen
3. Update version information
4. Configure linker
5. Create distribution package

Create a Release Build

❖ Always test your final build (what you plan to submit), and always submit release builds



Compiler

Configuration: Release

Platform:

General Options

- [ ] Generate overflow checks
- [x] Enable optimizations

Add icons and splash screens

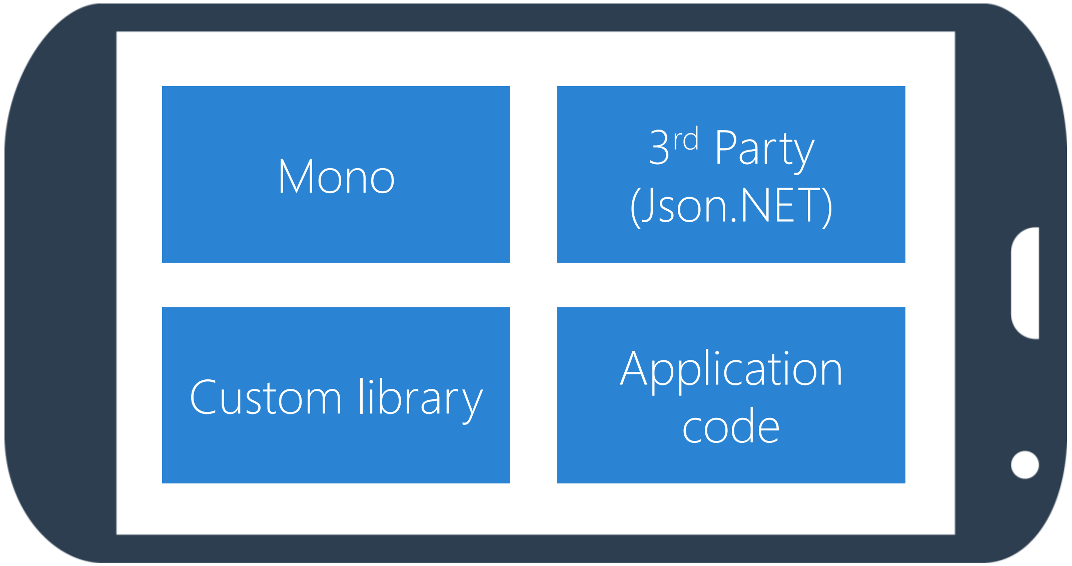
- Icon represents your app on the launch screen so it should be memorable and look *good!*
  - Follow the vendor guidance for size/shape
  - Supply multiple resolutions
  - Avoid text in the icon

Update version Info

- Versioning is important for maintenance and distribution
  - Increment major version for significant updates
  - Increment minor version for fixes

App composition

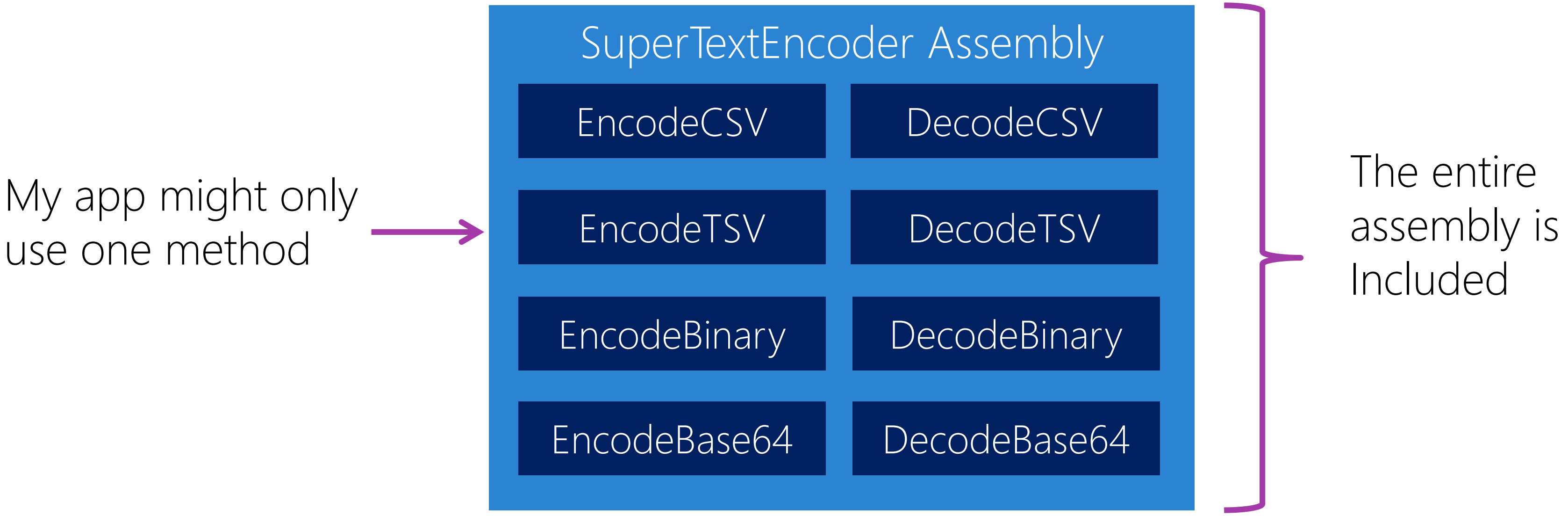
- The code that makes up an application typically includes many assemblies as well as the code you've written



Xamarin University

Referencing Assemblies

❖ By default, if you reference and use any part of an assembly, the compiler will include the entire assembly in your application

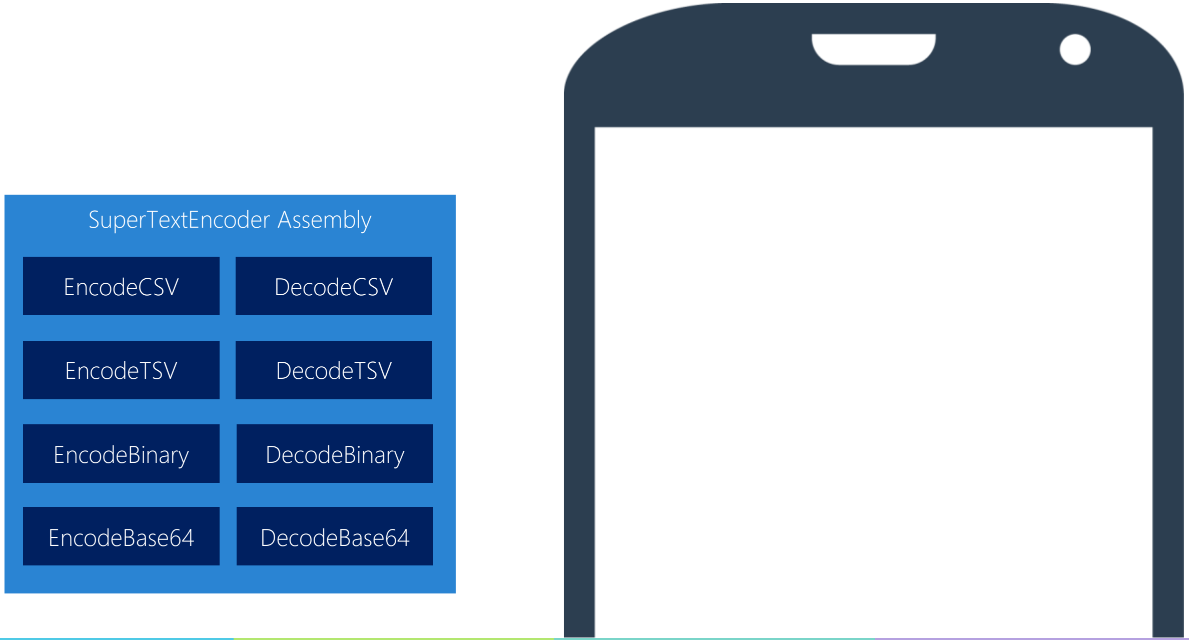


My app might only use one method
SuperTextEncoder Assembly
EncodeCSV
DecodeCSV
EncodeTSV
DecodeTSV
EncodeBinary
DecodeBinary
EncodeBase64
DecodeBase64
The entire assembly is Included

Xamarin University

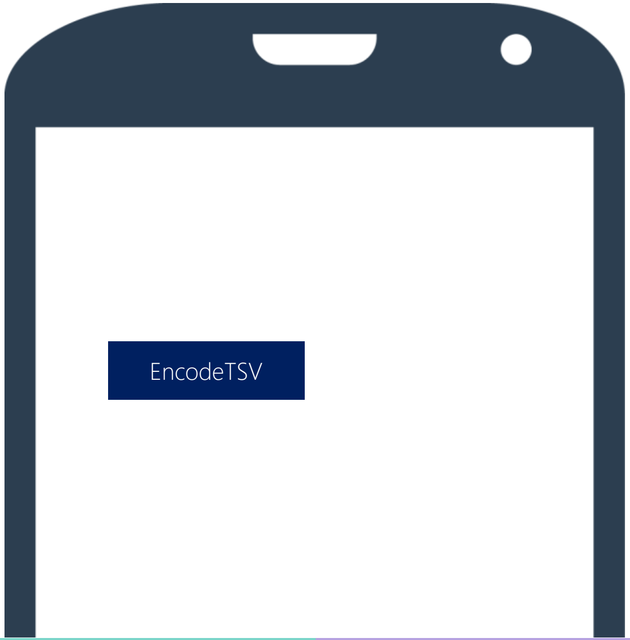
Referencing Assemblies

- By default, if you reference and use any part of an assembly, the compiler will include the entire assembly in your application



What is Linking?

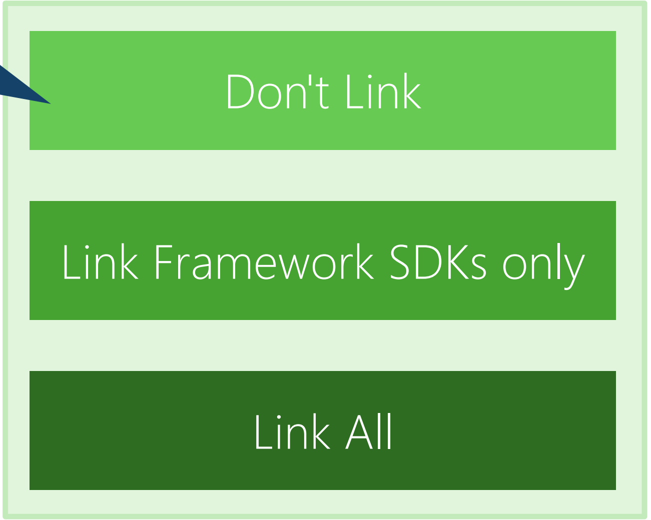
- Linking is the process of removing unused executable code from assemblies included with your application



Linker settings

- Linker settings can dramatically reduce the size of the app package, three options available for iOS and Android:

All code is added to application package – even code that is not referenced



Package size: 75.3Mb

Linker settings

❖ Linker settings can dramatically reduce the size of the app package, three options available for iOS and Android:

Don't Link	Package size: **75.3Mb**
Link Framework SDKs only	Package size: **34.1Mb**
Link All	

Core Mono and "safe" assemblies are scanned and pruned

Xamarin University

Linker settings

- Linker settings can dramatically reduce the size of the app package, three options available for iOS and Android:

Don't Link	Package size: **75.3Mb**
Link Framework SDKs only	Package size: **34.1Mb**
Link All	Package size: **22.9Mb**

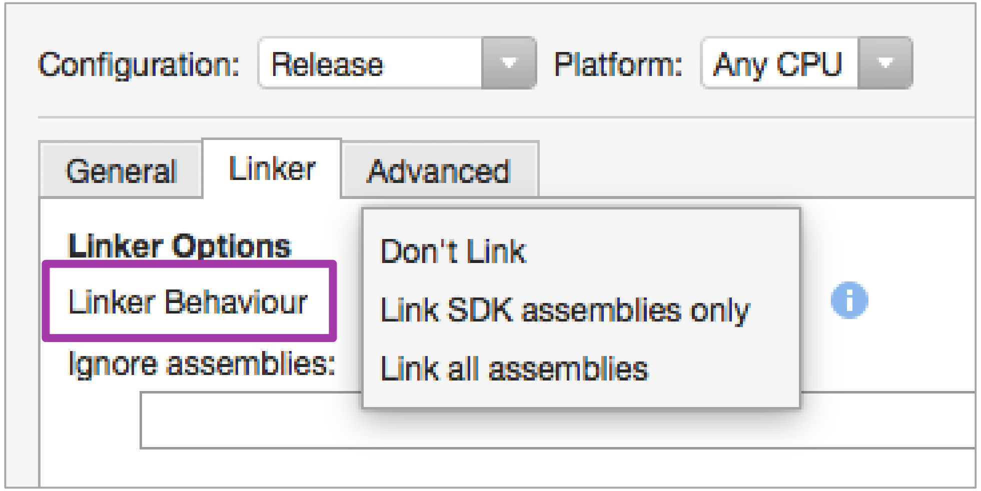
All referenced assemblies are examined by the Linker and potentially pruned

Xamarin University

Linker settings

- Linker settings can dramatically reduce the size of the app package, three options available for iOS and Android:

Project Options > [iOS | Android] Build


Configuration: Release Platform: Any CPU

General | Linker | Advanced

Linker Options

Linker Behaviour

Ignore assemblies:

Safe to Link assemblies

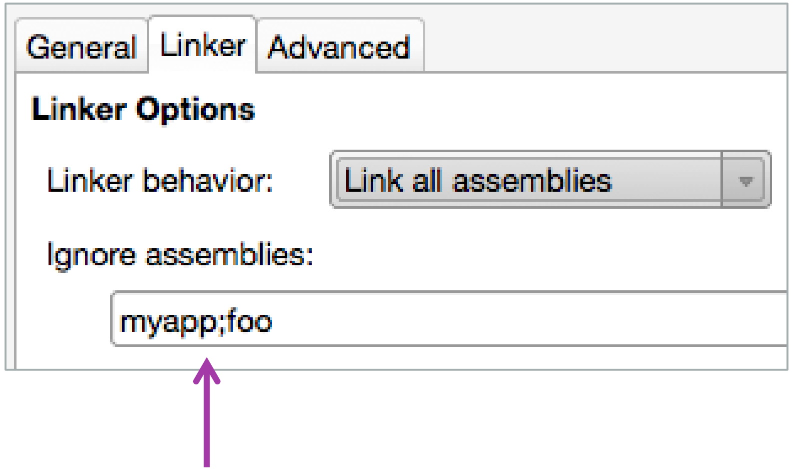
❖ Can indicate that your custom assemblies are safe to link by adding a custom assembly level attribute:

```
public sealed class LinkerSafeAttribute : Attribute
{ // Can be defined in your PCL code
}
```

```
// Then add in a single source file to tell the
// linker that this assembly should be considered an
// SDK assembly
[assembly: LinkerSafe]
```

Linking all assemblies

- You can link **ALL** assemblies to further reduce the size of your app package
- Will often remove things you actually are using
- Can create a **custom linker XML configuration** to indicate what to preserve (assemblies, types, and operations)



Linker Options

Linker behavior: Link all assemblies

Ignore assemblies: myapp;foo

For simple cases, you can tell the linker to exclude specific assemblies from it's pruning process

Xamarin University

Linker directives

- Linker can get *very* aggressive and will sometimes remove things your code actually needs – two ways to tell the linker to keep something

Code

Linker XML file

Preserving types in library code

❖ Can ensure entire types are preserved by the Linker through the `[Preserve]` attribute applied to the assembly or type itself

```
[Preserve(AllMembers=true)]
public class TodoTask
{
   [PrimaryKey, AutoIncrement]
   public int ID { get; set; }
   public string Name { get; set; }
}
```

```
[assembly: Preserve]
```

Preserving types in library code

- Can also use `[Preserve]` on fields, properties, delegates and methods which your code doesn't reference directly but are still necessary

```
public class TodoTask
{
    [PrimaryKey, AutoIncrement]
    [Preserve]
    public int ID { get; set; }
    public string Name { get; set; }
    public string Notes { get; set; }
    public bool Done { get; set; }
}
```

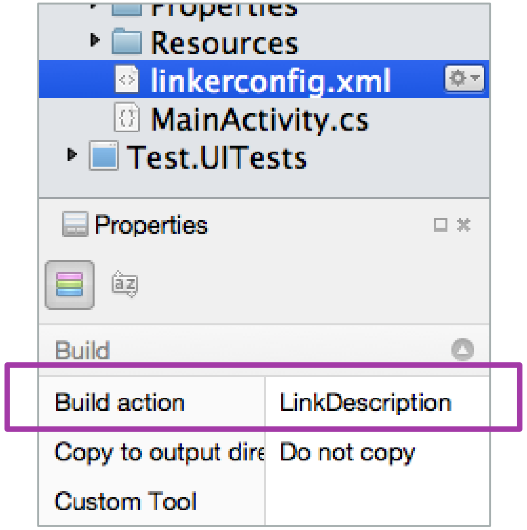
Preserving types in a PCL

- `PreserveAttribute` is defined in the core Xamarin assembly and not available in PCLs; however linker just looks for *any* attribute named `PreserveAttribute` so you can define one and use it to direct the linker

```
[AttributeUsage(AttributeTargets.All, AllowMultiple=true)]
public sealed class PreserveAttribute : System.Attribute
{
   public bool AllMembers;  // Keep all members
   public bool Conditional; // Keep member ONLY if type
                            // itself is kept
}
```

Advanced Linker settings

❖ XML linker configuration file must be added to your project



Resources
linkerconfig.xml
MainActivity.cs
Test.UITests

Properties

Build

Build action	LinkDescription
Copy to output dire	Do not copy
Custom Tool	

Set **build action** to **LinkDescription**

Preserving an entire assembly

- Can direct the linker to preserve an entire assembly – all types, methods will be retained in the final binary even if they are not referenced by your code

Assembly definitions listed here

```
<?xml version="1.0" encoding="UTF-8" ?>
<linker>
   <!-- preserve entire App.Core assembly -->
   <assembly fullname="App.Core">
       <type fullname="*"/>
   </assembly>
</linker>
```

Xamarin University

Preserving a specific type

- Can preserve a complete type (all fields and operations) in an assembly

```
<!-- preserve the App.Core.MainPage type
     in the App.Core assembly -->
<assembly fullname="App.Core">
    <type fullname="App.Core.MainPage" preserve="fields" />
</assembly>
```

Can tell linker to preserve all fields

Preserving all types in a namespace

- Can preserve all types in a namespace in the assembly

```
<!-- preserve the namespaces in the App.Core assembly -->
<assembly fullname=" App.Core">
    <namespace fullname="App.Core" />
    <namespace fullname="App.Core.Utils" />
    <namespace fullname="App.Core.ViewModels" />
</assembly>
```

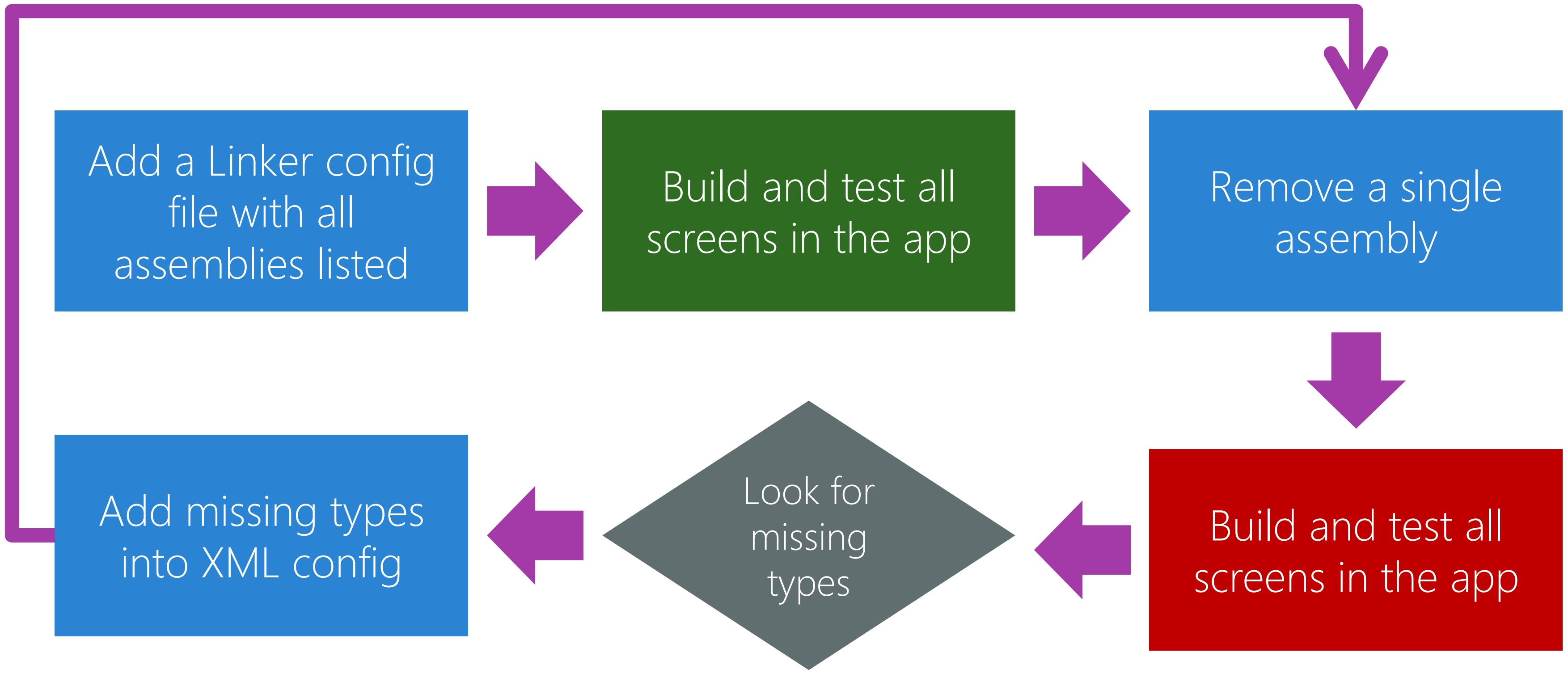
Xamarin University

Preserving a specific type

❖ Finally, can preserve specific operations in a type

```
<assembly fullname="App.Core">
    <type fullname="App.Core.MainPage">
        <!-- preserve the ValueChanged event -->
        <method name="add_ValueChanged"/>
        <method name="remove_ValueChanged"/>
        <!-- preserve the Value property -->
        <method name="get_Value"/>
        <method name="set_Value"/>
        <!-- preserve the _value field -->
        <field name="_value"/>
    </type>
</assembly>
```

Steps to link all assemblies


Add a Linker config file with all assemblies listed
Build and test all screens in the app
Remove a single assembly
Build and test all screens in the app
Look for missing types
Add missing types into XML config

Create a distribution package

- Each platform has a signed, packaged format which you must adhere to when submitting or installing apps onto a device

App Bundle (.app)

App Package (.apk)

AppX Package (.appx)

Flash Quiz

Flash Quiz

① Which of these components is necessary when preparing an app for release?

a) Disable debugging

b) Specify app icon

c) Set packaging properties

d) None of the above

Flash Quiz

① Which of these components is necessary when preparing an app for release?

a) Disable debugging
b) Specify app icon
c) Set packaging properties
d) None of the above

Flash Quiz

② Why do you need to disable debugging when you publish an app?

a) To remove the source code from the app

b) To reduce the size of the app package/bundle

c) To ensure your app is optimized

Flash Quiz

② Why do you need to disable debugging when you publish an app?

a) To remove the source code from the app

b) To reduce the size of the app package/bundle

c) To ensure your app is optimized

Flash Quiz

③ Why would you want to set the linker settings when publishing your app?

a) To reduce the size of the app
b) To discard unused assemblies, types and members
c) To protect your app from outside tampering
d) To target multiple platforms

Flash Quiz

③ Why would you want to set the linker settings when publishing your app?

a) To reduce the size of the app

b) To discard unused assemblies, types and members

c) To protect your app from outside tampering

d) To target multiple platforms

Xamarin University

Publishing Styles

- Three common ways to distribute your applications

Adhoc / Side-loading

Direct via email or website, often used for testing

Xamarin University

Publishing Styles

- Three common ways to distribute your applications

Adhoc /
Side-loading

Store

Most common approach and widest distribution model

Publishing Styles

- Three common ways to distribute your applications

Adhoc / Side-loading

Store

Enterprise

Mostly used for internal, corporate apps

Choosing a store / market

- Vendors operate branded stores where they market and distribute your app for a percentage of the sale

App Store

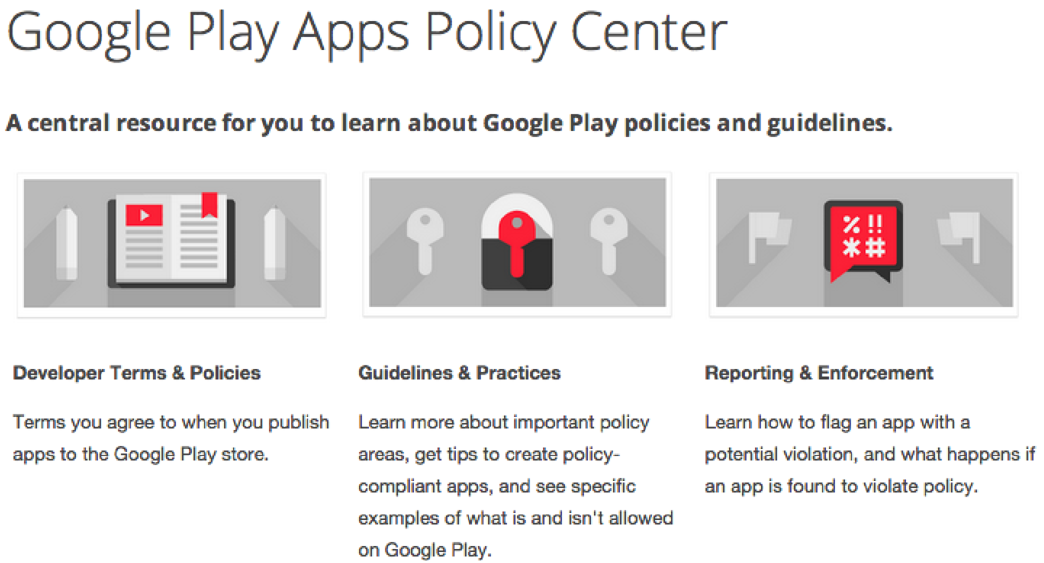
Google Play

Windows Marketplace

Xamarin University

Read the licenses carefully

- Each public store has different rules you must adhere to, read the license carefully before submitting your app to make sure you are a good citizen



Google Play Apps Policy Center

A central resource for you to learn about Google Play policies and guidelines.

Developer Terms & Policies

Terms you agree to when you publish apps to the Google Play store.

Guidelines & Practices

Learn more about important policy areas, get tips to create policy-compliant apps, and see specific examples of what is and isn't allowed on Google Play.

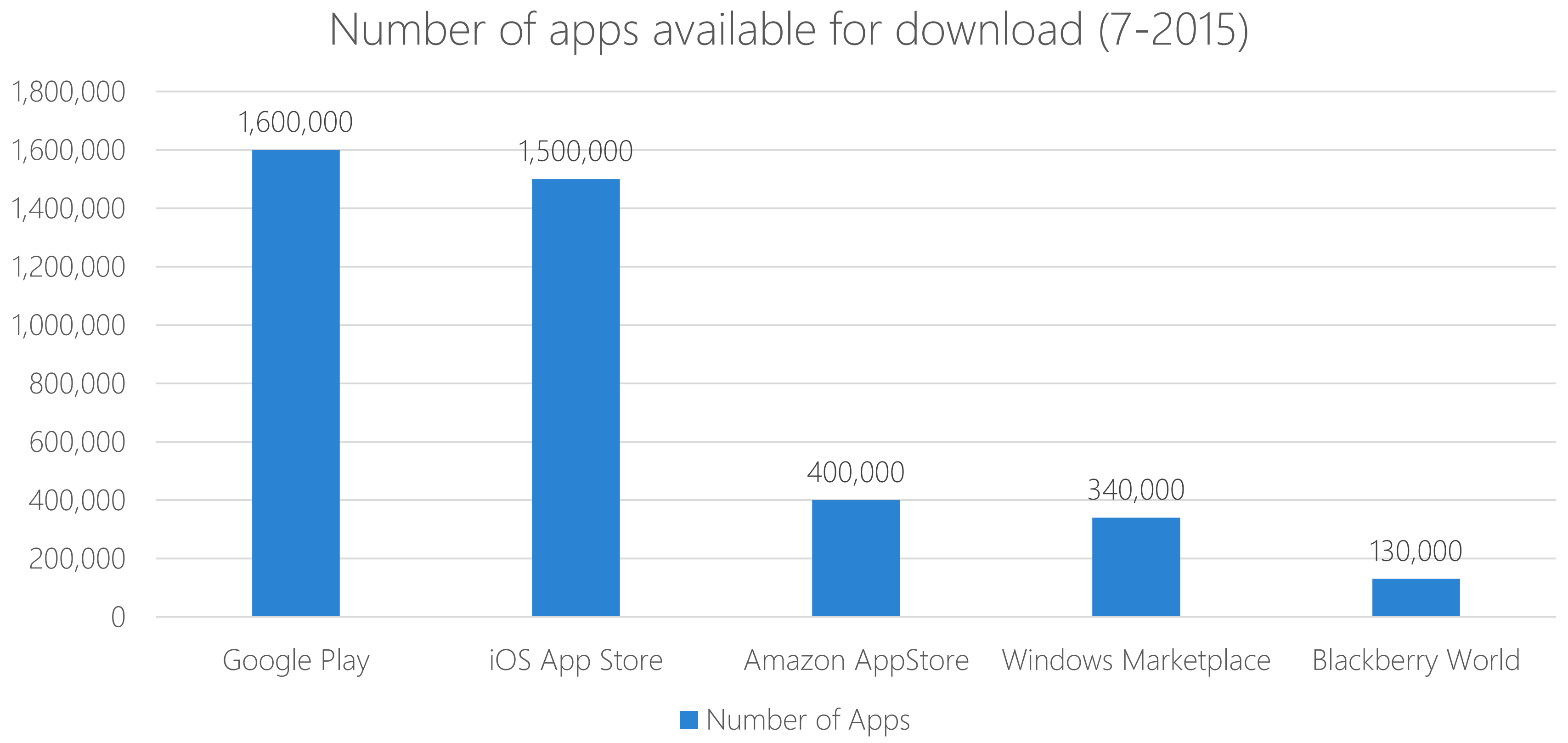
Reporting & Enforcement

Learn how to flag an app with a potential violation, and what happens if an app is found to violate policy.

Most of them publish nice guidelines – worth checking out

Xamarin University

Choosing a store / market



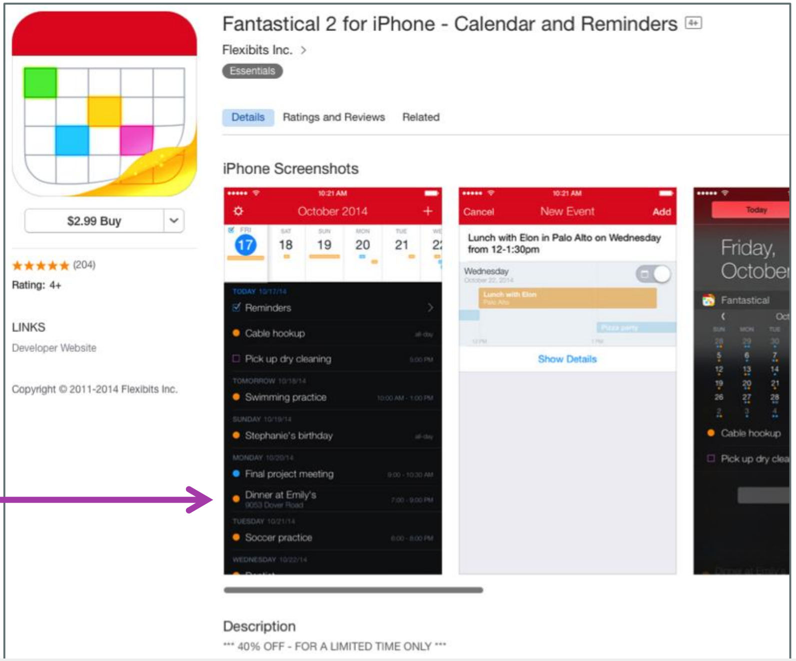
Choosing a store / market

Registration Fee	$99 / $299 annual	$25 one-time	~$19 / $99 one-time
app # limits	none	none	100 free
Market Share	~20%	~75%	~5%
Revenue sharing %	70 / 30	70 / 30	70 / 30 sliding
Reasons to put your app here	Higher daily revenue, more $$$	Best searching, new apps found quickly	Less competition = more opportunity

Creating the marketing information

- All of the app stores allow you to provide screen shots, descriptions, and requirements for your app – use these to your advantage so people notice your app!

use video and flashy screen shots showing the best aspects of the app



Fantastical 2 for iPhone - Calendar and Reminders 4+
Flexibits Inc.
Essentials

Details Ratings and Reviews Related

iPhone Screenshots

$2.99 Buy

★★★★★ (204)
Rating: 4+

LINKS
Developer Website

Copyright © 2011-2014 Flexibits Inc.

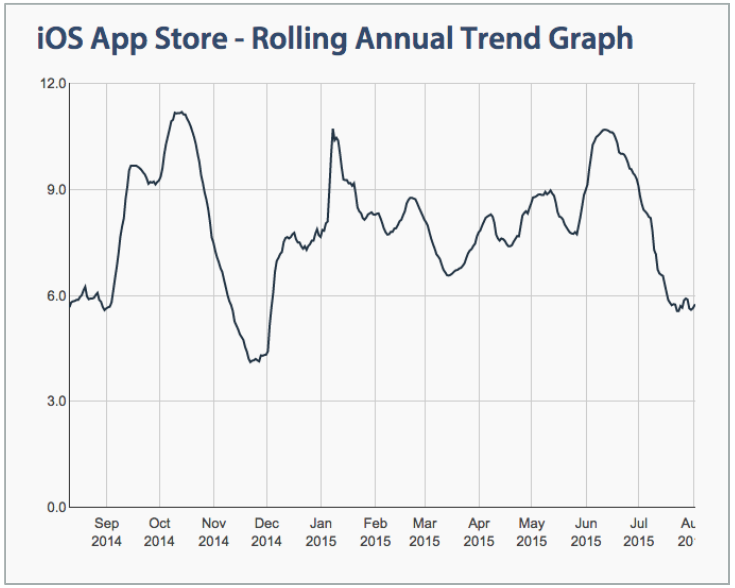
Description
*** 40% OFF - FOR A LIMITED TIME ONLY ***

This is one of the *most important* things you will do when publishing your app – keywords, images and descriptions determine how easily users find your

Xamarin University

App review process

- Each store will review your application prior to making it available to the public – times vary, but it could take a week or more before it goes online
- Will get an email notification when the app is either accepted or rejected; along with reasons for rejection



Xamarin University

Determine a revenue strategy

- Have realistic expectations of how much you will make
- What are similar apps priced?
  - Simple apps often free or $0.99
  - Higher priced apps need to look good and provide high value or you will get bad ratings
- Region influences pricing; U.S. tends to pay more for apps

Consider including ads

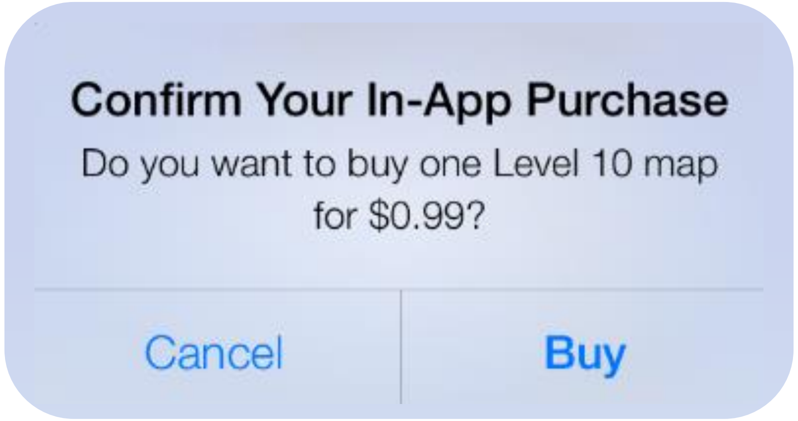
- In-app advertisements can generate additional revenue – Flappy Bird was reportedly generating $50k per day in ads

Free screen space can be used to host advertisements

Xamarin University

Consider in-app purchases

- Use In-App purchases to move from a **free** or **reduced-price** model to a full version of your app, or to add features to the app (but be careful with this!)


Confirm Your In-App Purchase

Do you want to buy one Level 10 map for $0.99?

What's Next?

- Learn how to package and upload your app to a store
  - iOS App Store
  - Google Play Store (Android)
  - Amazon App Store (Android)
  - Windows Marketplace

- Watch specific video for each platform you want to publish!

Thank You!

Please complete the class survey in your profile:
university.xamarin.com/profile

Microsoft