

 免费电子书

学习

Xamarin.Android

Free unaffiliated eBook created from
Stack Overflow contributors.

#xamarin.an

droid

.....	1
1: Xamarin.Android	2
.....	2
.....	2
Examples.....	2
Xamarin Studio.....	2
Visual Studio.....	4
2: RecyclerView	7
Examples.....	7
RecyclerView.....	7
ClickRecyclerView.....	11
3: Xamarin.Android -	14
.....	14
Examples.....	14
Xamarin.Android.....	14
4: Xamarin.Android -	17
.....	17
.....	17
Examples.....	17
.....	17
5: Xamarin.Android APK	19
.....	19
Examples.....	19
Visual StudioAPK.....	19
.....	21
Xamarin.Android APKMultiDex.....	29
Xamarin.AndroidMultiDex.....	29
Xamarin.Android APKProGuard.....	32
Xamarin.AndroidProGuard.....	32
ProGuardLinker ⁴⁰⁰	34
Xamarin.Linker.....	34

ProGuard.....	37
6: XamarinZXing.....	40
.....	40
Examples.....	40
.....	40
7: Android.....	41
.....	41
Examples.....	41
Android.....	41
8:	48
.....	48
Examples.....	48
.....	48
9:	49
.....	49
.....	49
Examples.....	50
AlertDialog.....	50
.....	50
10: - Xamarin.Andorid.....	53
.....	53
.....	53
Examples.....	53
.....	53
.....	54
.....	55
GitHub.....	57
11:	58
Examples.....	58
Toast.....	58
.....	58
.....	58

12:	60
Examples	60
.....	60
Java	60
.....	60
13: ListView	62
Examples	62
.....	62
.....	68

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xamarin-android](#)

It is an unofficial and free Xamarin.Android ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Xamarin.Android.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: Xamarin.Android

Xamarin.AndroidJavaUIAndroidC.NETBCLIDE - Xamarin StudioVisual Studio - ◦

MacWindowsXamarin.AndroidXamarin

		API	
1.0		1	2008-09-23
1.1		2	2009-02-09
1.5		3	2009-04-27
1.6		4	2009-09-15
2.0-2.1		5-7	2009-10-26
2.2-2.2.3	Froyo	8	2010-05-20
2.3-2.3.7		9-10	2010-12-06
3.0-3.2.6		11-13	2011-02-22
4.0-4.0.4		14-15	2011-10-18
4.1-4.3.1		16-18	2012-07-09
4.4-4.4.4,4.4W-4.4W.2		19-20	20131031
5.0-5.1.1		21-22	20141112
6.0-6.0.1		23	2015105
7		24	2016822

Examples

Xamarin Studio

1. >>◦
2. **Android App** “ ”◦
3. ID◦ ◦

Configure your Android app

App Name:

Organization Identifier:

com.xamarin

Package Name:

com.xamarin.appname

Target Platforms:

Maximum Compatibility

Minimum: 2.3 "Gingerbread" (API 10)

Modern Development

Minimum: 4.1 "Jelly Bean" (API 16)

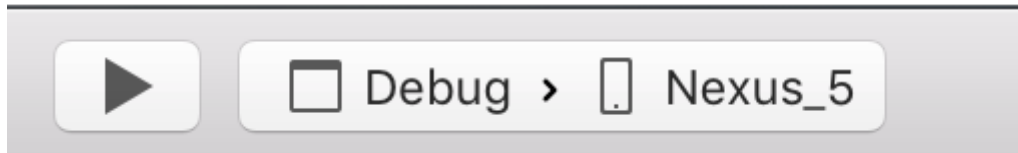
Latest and Greatest

Theme:

Default

4. ◦ “”。

5.



6. **Debug**“”

Visual Studio

1. >>“”。

2. **Visual C> AndroidBlank App**

New Project

▷ Recent

.NET Framework 4.5.2

Sort

◀ Installed

◀ Templates

◀ Visual C#

▷ Windows

Web

Android

Cloud

Cross-Platform

Extensibility

◀ iOS

Apple Watch

Extensions

iPad

iPhone

Universal

LightSwitch

Office/SharePoint

Silverlight



Blank App (Android)



Wear App (Android)



WebView App (Android)



OpenGL Game (Android)



Class Library (Android)



Bindings Library (Android)



UI Test App (Xamarin.UI)



Unit Test App (Android)

▷ Online

[Click here](#)

Name:

App2

Location:

C:\Users\Amy\Documents\

Solution:

Create new solution

Solution name:

App2

3. ◦

4.

5. “ ” “ ”



Xamarin.Android <https://riptutorial.com/zh-CN/xamarin-android/topic/403/xamarin-android>

2: RecyclerView

Examples

RecyclerView

Android Support Library v7 RecyclerView UI

RecyclerView Nuget v7 recyclerview Xamarin Android Support Library - v7 RecyclerView



A

Official NuGet Gallery



Xamarin Android Support Library - v7 AppCompat

v7 AppCompat Android Support Library C# bindings for



Xamarin Android Support Library - v7 RecyclerView

v7 RecyclerView Android Support Library C# bindings for



Xamarin Android Support Library - v7 RecyclerView

v7 RecyclerView Android Support Library C# bindings for



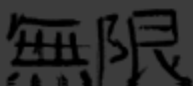
Crosslight - Xamarin Android Support Library -

Signed Xamarin Android Support Library - v7 RecyclerView
Crosslight.



v7

v7 Class Library



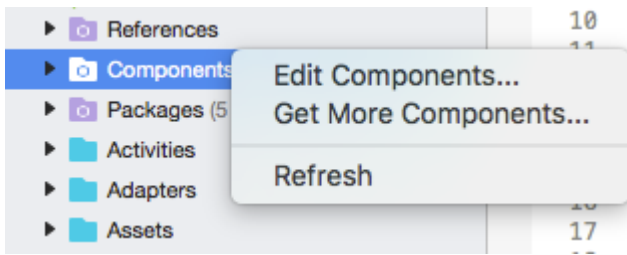
MugenMvvmToolkit - Android Support Library v7

This package adds Android Support Library v7 RecyclerView
MugenMvvmToolkit.

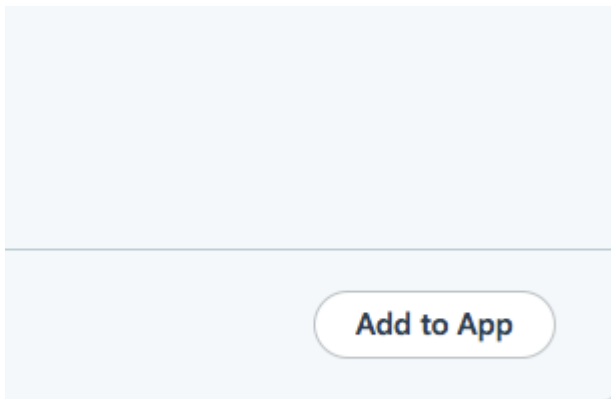


Show pre-release packages

Get More Components ◦



Component Store RecyclerView ◦ Android Support Library V7 RecyclerView ◦ “Add to App” ◦ ◦



RecyclerView ◦ axml RecyclerView ◦

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

RecyclerView AdapterViewHolder ◦ Adapter RecyclerView ◦ ViewHolder ◦ ◦

Adapter Class

```
public class MyAdapter : RecyclerView.Adapter
{
    string [] items;

    public MyAdapter (string [] data)
    {
        items = data;
    }

    // Create new views (invoked by the layout manager)
    public override RecyclerView.ViewHolder OnCreateViewHolder (ViewGroup parent, int
viewType)
    {
        // set the view's size, margins, paddings and layout parameters
        var tv = new TextView (parent.Context);
        tv.SetWidth (200);
        tv.Text = "";
    }
}
```

```

        var vh = new MyViewHolder (tv);
        return vh;
    }

    // Replace the contents of a view (invoked by the layout manager)
    public override void OnBindViewHolder (RecyclerView.ViewHolder viewHolder, int position)
    {
        var item = items [position];

        // Replace the contents of the view with that element
        var holder = viewHolder as MyViewHolder;
        holder.TextView.Text = items[position];
    }

    public override int getItemCount {
        get {
            return items.Length;
        }
    }
}

```

OnCreateViewHolder **ViewHolder** ◦ ◦ **ViewHolder** ◦ ◦ **RecyclerView.ViewHolder** ◦

OnBindViewHolder ◦ **ViewHolder** ◦

TextView **ViewHolder** ◦

```

public class MyViewHolder : RecyclerView.ViewHolder
{
    public TextView TextView { get; set; }

    public MyViewHolder (TextView v) : base (v)
    {
        TextView = v;
    }
}

```

Activity ◦

```

RecyclerView mRecyclerView;
MyAdapter mAdapter;
protected override void OnCreate (Bundle bundle)
{
    base.OnCreate (bundle);
    SetContentView (Resource.Layout.Main);
    mRecyclerView = FindViewById<RecyclerView> (Resource.Id.recyclerView);

    // Plug in the linear layout manager:
    var layoutManager = new LinearLayoutManager (this) { Orientation =
LinearLayoutManager.Vertical };
    mRecyclerView.SetLayoutManager (layoutManager);
    mRecyclerView.HasFixedSize = true;

    var recyclerViewData = GetData();
    // Plug in my adapter:
    mAdapter = new MyAdapter (recyclerViewData);
}

```

```

    mRecyclerView.setAdapter (mAdapter);
}

string[] GetData()
{
    string[] data;
    .
    .
    .
    return data;
}

```

LayoutManagerRecyclerView ◦ RecyclerView ListView GridView ◦ LayoutMangerRecyclerView ◦
 LinearLayoutManager ◦ ListView ◦ GridLayoutManager ◦

ClickRecyclerView

Xamarin.Android RecyclerViewClick EventHandlers ◦

Android Java ClickonClickListener

```

ImageView picture = findViewById(R.id.item_picture);
picture.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // do stuff
    }
});

```

Xamarin.AndroidClick EventHandler

1◦

```

ImageView picture = FindViewById<ImageView>(Resource.Id.item_picture);
picture.Click += delegate {
    // do stuff
};

```

2◦

```

ImageView picture = FindViewById<ImageView>(Resource.Id.item_picture);
picture.Click += async delegate {
    // await DoAsyncMethod();
    // do async stuff
};

```

3◦

```

ImageView picture = FindViewById<ImageView>(Resource.Id.item_picture);
picture.Click += Picture_Click;
... // rest of your method

private void Picture_Click(object sender, EventArgs e)
{

```

```
// do stuff
}
```

EventHandler。 GridView / ListViewGetViewClick EventHandlerRecyclerView.Adapter
OnBindViewHolderEventHandler。 EventHandler。

EventHandlersGetViewOnBindViewHolder。 **3.EventHandlerEventHandler**。

ClickRecyclerView.Adapter

```
public class ViewHolderPerson : Android.Support.V7.Widget.RecyclerView.ViewHolder
{
    public View Item { get; private set; }
    public ImageView Picture { get; private set; }
    public TextView Name { get; private set; }

    public ViewHolderPerson(View itemView) : base(itemView)
    {
        this.Item = itemView;
        this.Picture = itemView.FindViewById<ImageView>(Resource.Id.Item_Person_Picture);
        this.Name = itemView.FindViewById<TextView>(Resource.Id.Item_Person_Name);
    }
}

public class AdapterPersons : Android.Support.V7.Widget.RecyclerView.Adapter
{
    private Context context;
    private Android.Support.V7.Widget.RecyclerView recyclerView;
    private List<Person> persons;

    public AdapterPersons(Context context, Android.Support.V7.Widget.RecyclerView
recyclerView, List<Person> persons)
    {
        this.context = context;
        this.recyclerView = recyclerView;
        this.persons = persons;
    }

    public override int getItemCount => persons.Count;

    public override void onBindViewHolder(RecyclerView.ViewHolder holder, int position)
    {
        Person person = this.persons[position];
        ((ViewHolderPerson)holder).Name.Text = person.Name;
        ((ViewHolderPerson)holder).Picture.SetImageBitmap(person.Picture);

        // Unsubscribe and subscribe the method, to avoid setting multiple times.
        ((ViewHolderPerson)holder).Item.Click -= Person_Click;
        ((ViewHolderPerson)holder).Item.Click += Person_Click;
    }

    private void Person_Click(object sender, EventArgs e)
    {
        int position = this.recyclerView.GetChildAdapterPosition((View)sender);
        Person personClicked = this.persons[position];
        if(personClicked.Gender == Gender.Female)
        {
            Toast.MakeText(this.context, "The person clicked is a female!",
ToastLength.Long).Show();
        }
    }
}
```



```
    }
    else if(personClicked.Gender == Gender.Male)
    {
        Toast.MakeText(this.context, "The person clicked is a male!",
ToastLength.Long).Show();
    }
}

public override RecyclerView.ViewHolder OnCreateViewHolder(ViewGroup parent, int viewType)
{
    View itemView =
LayoutInflater.From(parent.Context).Inflate(Resource.Layout.item_person, parent, false);
    return new ViewHolderPerson(itemView);
}
}
```

RecyclerView <https://riptutorial.com/zh-CN/xamarin-android/topic/3452/recyclerview>

3: Xamarin.Android -

Android

<https://developer.android.com/reference/android/support/v7/widget/Toolbar.html>

Android.Support.v7

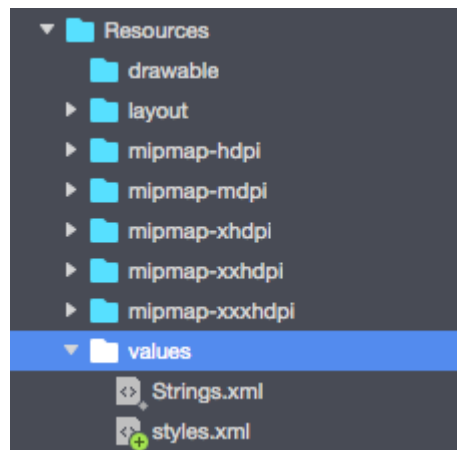
<https://developer.android.com/training/appbar/index.html>

Examples

Xamarin.Android

NuGetXamarin.Android.Support.V7.AppCompat [https](https://www.nuget.org/packages/Xamarin.Android.Support.v7.AppCompat/)

[//www.nuget.org/packages/Xamarin.Android.Support.v7.AppCompat/](https://www.nuget.org/packages/Xamarin.Android.Support.v7.AppCompat/)



“Resources”“values”“styles.xml”xml

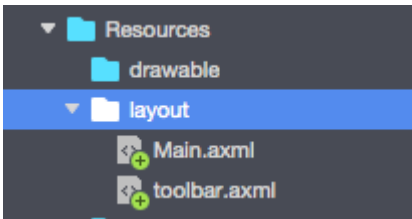
“styles.xml”

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
<style name="MyTheme" parent="MyTheme.Base">
</style>

<!-- Base theme applied no matter what API -->
<style name="MyTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">
<item name="windowNoTitle">true</item>
<!--We will be using the toolbar so no need to show ActionBar-->
<item name="windowActionBar">false</item>
<!-- Set theme colors from http://www.google.com/design/spec/style/color.html#color-color-
palette-->
<!-- colorPrimary is used for the default action bar background -->
<item name="colorPrimary">#2196F3</item>
<!-- colorPrimaryDark is used for the status bar -->
<item name="colorPrimaryDark">#1976D2</item>
<!-- colorAccent is used as the default value for colorControlActivated
which is used to tint widgets -->
<item name="colorAccent">#FF4081</item>
```

```
<item name="colorControlHighlight">#FF4081</item>
<!-- You can also set colorControlNormal, colorControlActivated
colorControlHighlight and colorSwitchThumbNormal. -->
```

“toolbar.axml”“layout”



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:minHeight="?attr/actionBarSize"
android:background="?attr/colorPrimary"
android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
```

“Main.xml”。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">

    <include android:id="@+id/toolbar" layout="@layout/toolbar" />

</LinearLayout>
```

。 “AndroidManifest”“application”

```
<application android:theme="@style/MyTheme" android:allowBackup="true"
android:icon="@mipmap/icon" android:label="@string/app_name">
```

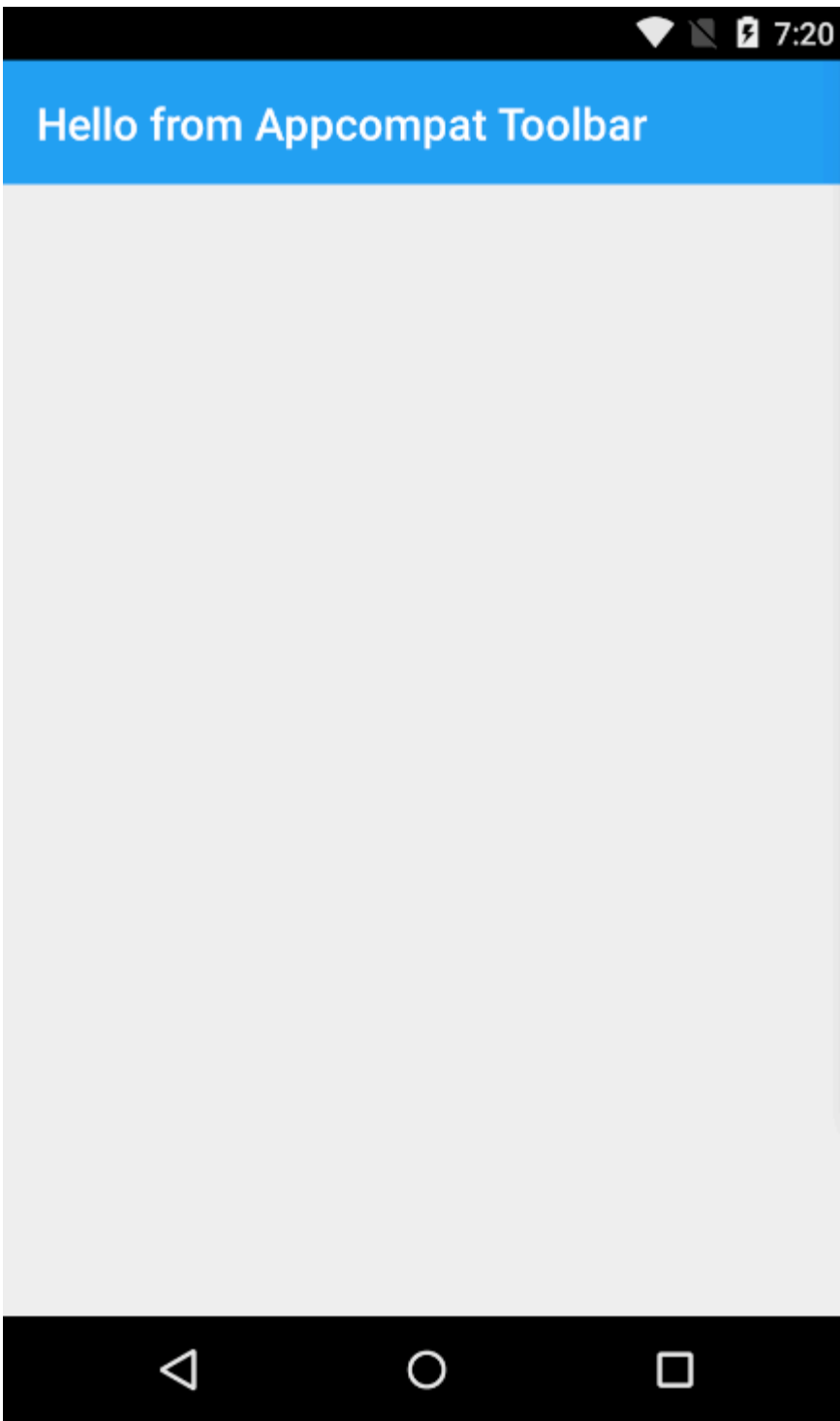
Activity。 “MainActivity.cs”。 “AppCompatActivity”。 “OnCreate”。

```
var toolbar = FindViewById<Android.Support.V7.Widget.Toolbar>(Resource.Id.toolbar);
SetSupportActionBar(toolbar);
SupportActionBar.Title = "Hello from AppCompatActivity";
```

```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    SetContentView(Resource.Layout.Main);

    var toolbar = FindViewById<Android.Support.V7.Widget.Toolbar>(Resource.Id.toolbar);
    SetSupportActionBar(toolbar);
```

```
SupportActionBar.Title = "Hello from Appcompat Toolbar";  
}
```



Xamarin.Android - <https://riptutorial.com/zh-CN/xamarin-android/topic/4755/xamarin-android--->

4: Xamarin.Android -

Xamarin.Android BluetoothSocket.InputStreamBluetoothSocket.OutputStream
System.IO.Stream ◦ System.IO.Stream◦

BluetoothSocket◦ ◦	
CMD	BT◦
_mx	◦ System.Threading.Mutex◦
	◦

Examples

[Android.Runtime.InputStreamInvokerAndroid.Runtime.OutputStreamInvokerJava.IO.InputStream](#)
[Java.IO.OutputStream](#) ◦ [java.io.InputStream.Available.Read](#)

```
byte[] Talk2BTsocket(BluetoothSocket socket, byte[] cmd, Mutex _mx, int timeOut = 150)
{
    var buf = new byte[0x20];

    _mx.WaitOne();
    try
    {
        using (var ost = socket.OutputStream)
        {
            var _ost = (ost as OutputStreamInvoker).BaseOutputStream;
            _ost.Write(cmd, 0, cmd.Length);
        }

        // needed because when skipped, it can cause no or invalid data on input stream
        Thread.Sleep(timeOut);

        using (var ist = socket.InputStream)
        {
            var _ist = (ist as InputStreamInvoker).BaseInputStream;
            var aa = 0;
            if ((aa = _ist.Available()) > 0)
            {
                var nn = _ist.Read(buf, 0, aa);
                System.Array.Resize(ref buf, nn);
            }
        }
    }
    catch (System.Exception ex)
    {
        DisplayAlert(ex.Message);
    }
    finally
    {

```

```
        _mx.ReleaseMutex();    // must be called here !!!  
    }  
  
    return buf;  
}
```

Xamarin.Android - <https://riptutorial.com/zh-CN/xamarin-android/topic/10844/xamarin-android---->

5: Xamarin.Android APK

Xamarin.Android。

Examples

Visual StudioAPK

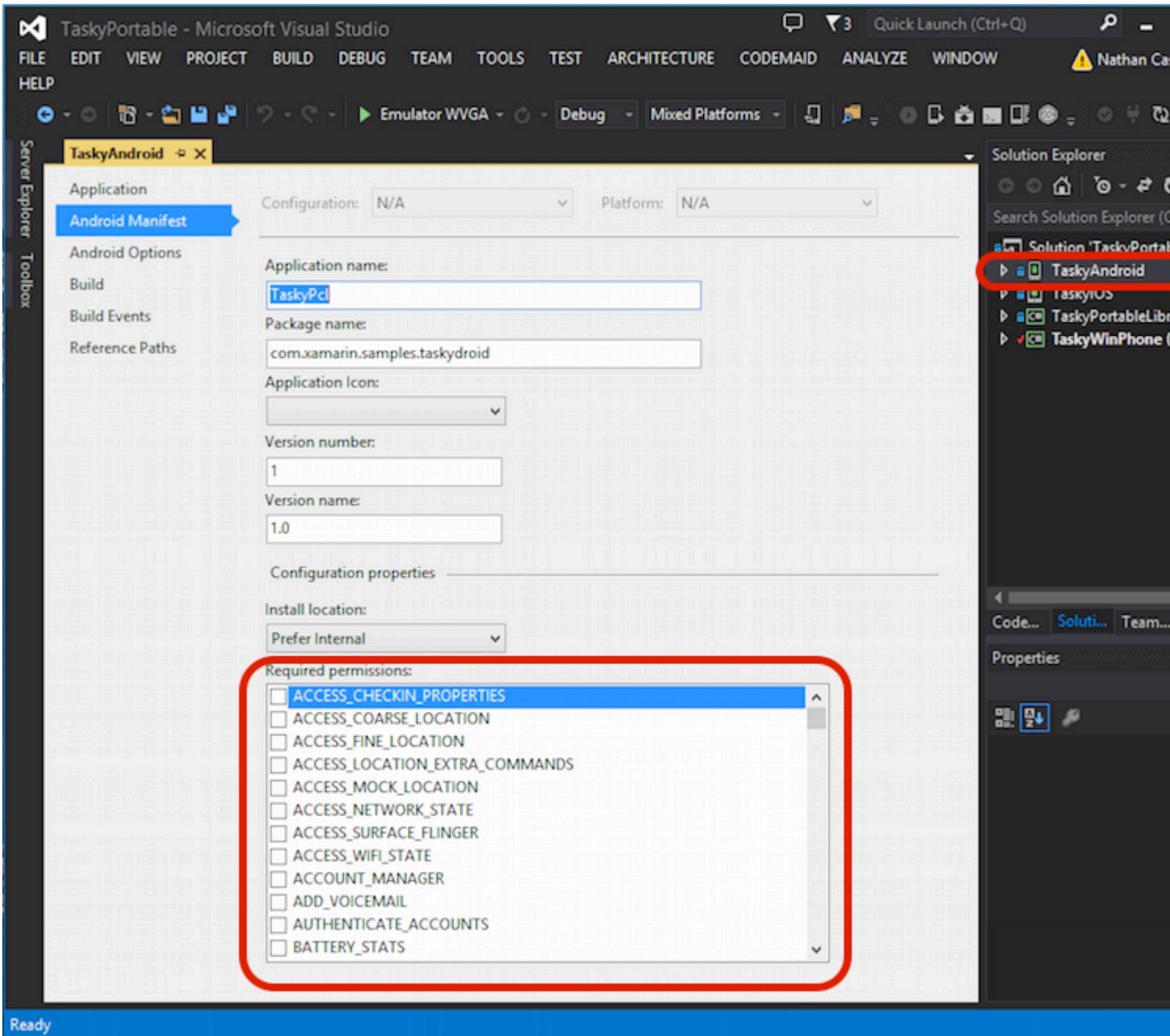
- Google Play。

Xamarin

https://developer.xamarin.com/guides/android/deployment,_testing,_and_metrics/publishing_an_application

Android Manifest

Visual StudioXamarin.Android“”。 Android Manifest

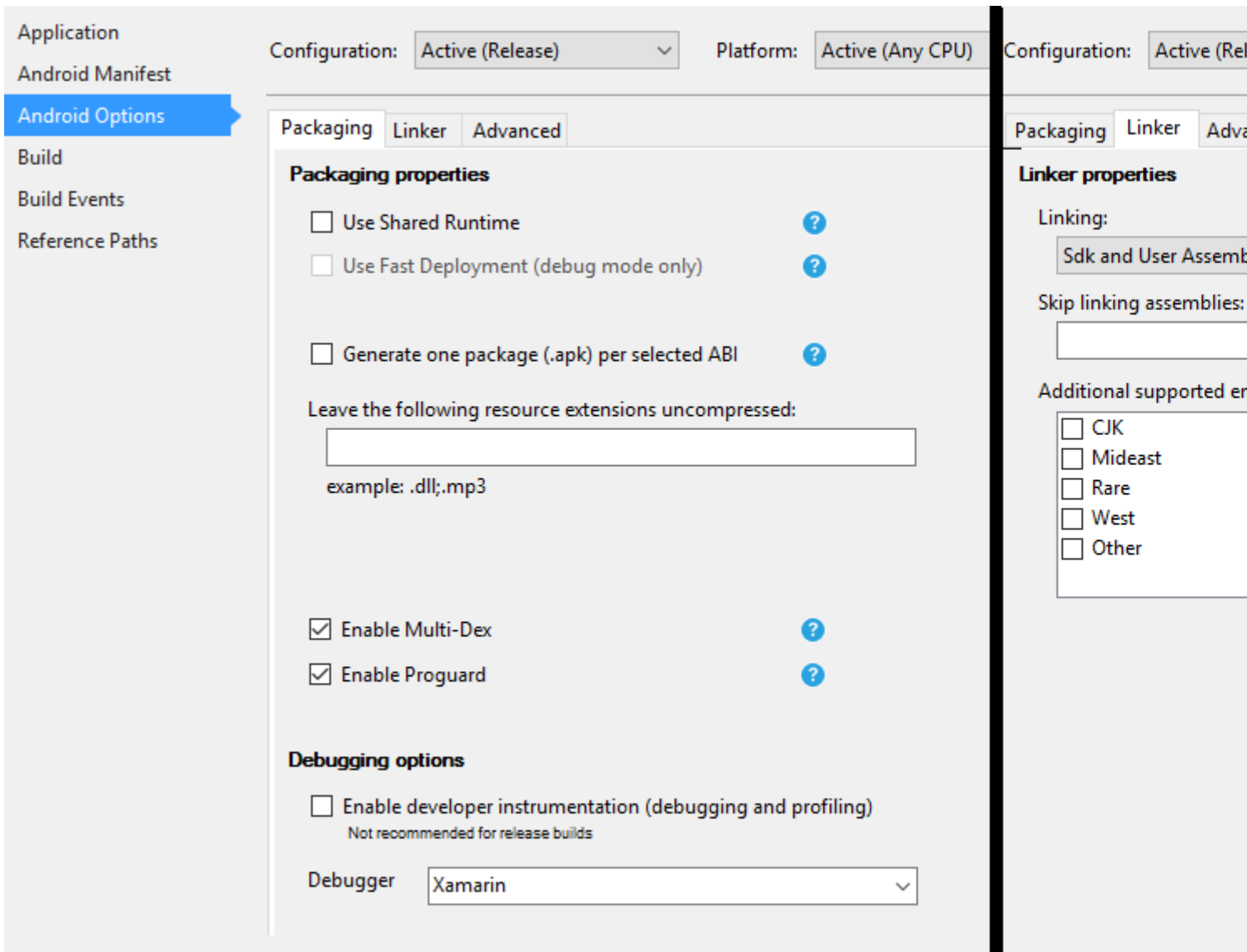


Android StudioEclipseAndroidManifest.xml; XamarinVisual Studio。 BroadcastReceiversAndroid Manifest 。

- 。
- 。
- Google Play。
- Android StudioEclipse@ drawable / ic_launcher。
- Google Play。 APK1。
- 。
- APK SD。
- 。

Android

- APK。



-
- **CPU** ◦ Google PlayAPK。 “”Google Play。
- **false** ◦ trueAPKMono Runtime。 USBMono RuntimeRelease APK。 Mono RuntimeRelease APKtrue。
- **ABI.apk false** ◦ APK。
- **Multi-Dex true** 65536 false。
- **Proguard true** ◦ ProguardJava。 .NET;.NETDotfuscator。 Proguard for Xamarin.Android
-
- Release APK**false** ◦
- **SDK** ◦ XamarinSDKAPK。

Xamarin.LinkerCorePCL。 “Sdk”

PreserveAttribute.cs

```
namespace My_App_Core.Models
{
```

```
public sealed class PreserveAttribute : System.Attribute
{
    public bool AllMembers;
    public bool Conditional;
}
}
```

```
using System;

namespace My_App_Core.Models
{
    [Preserve(AllMembers = true)]
    public class ServiceException : Exception
    {
        public int errorCode;

        [Preserve(AllMembers = true)]
        public ServiceException() { }

        [Preserve(AllMembers = true)]
        public ServiceException(int errorCode)
        {
            this.errorCode = errorCode;
        }
    }
}
```

- **all** ◦

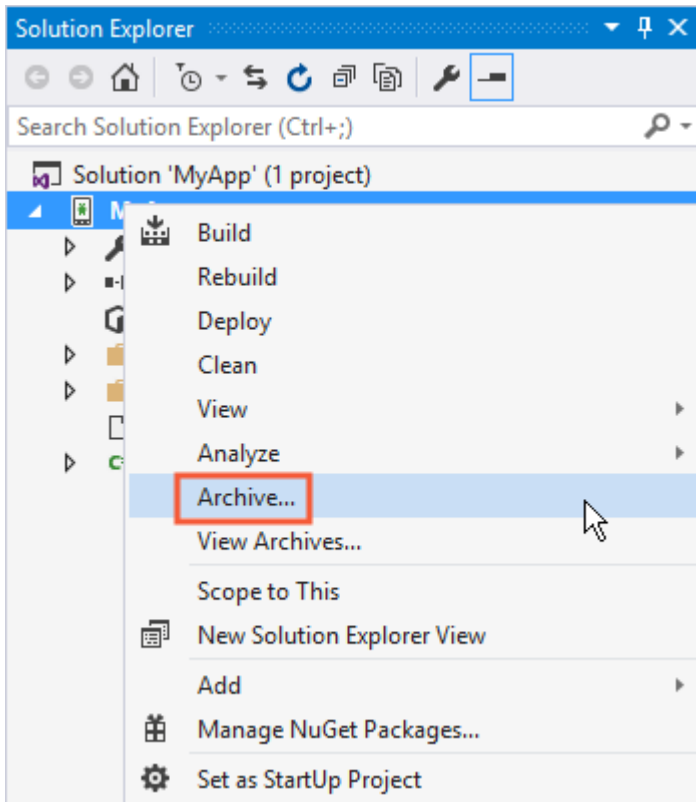
◦

APK

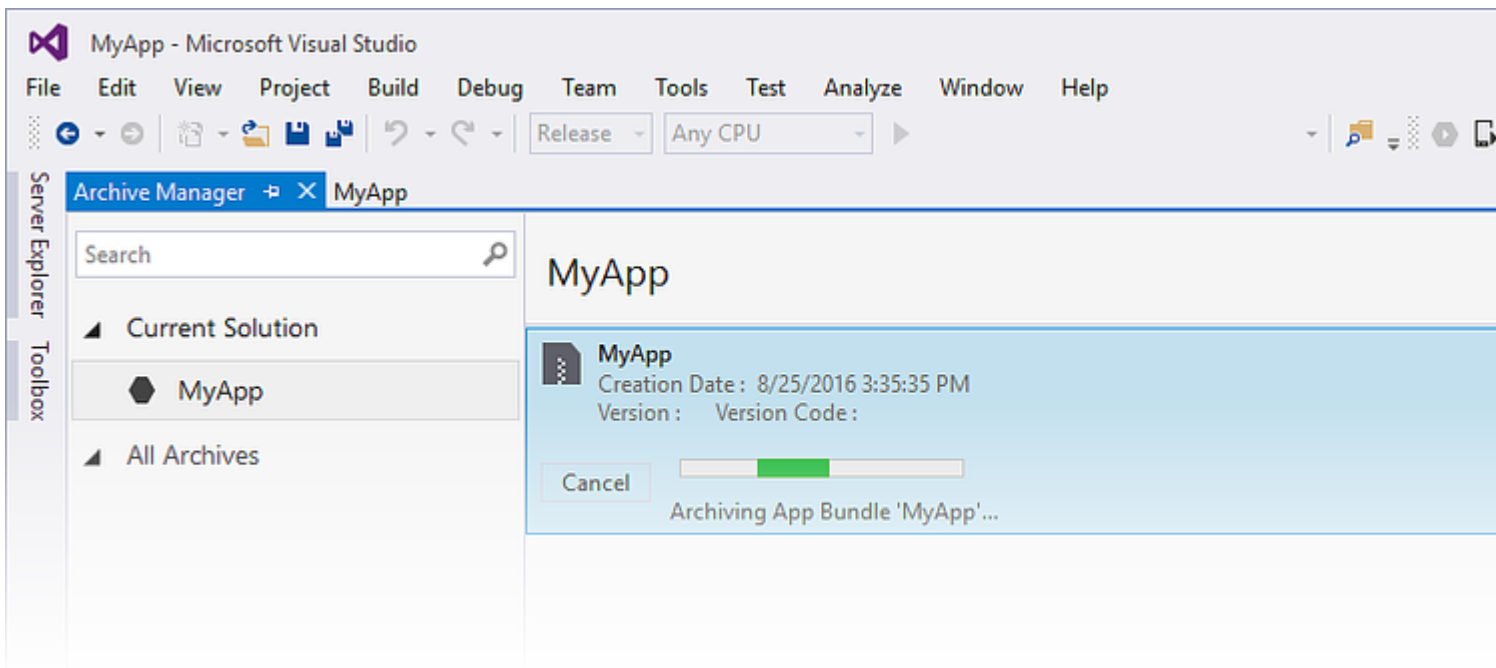
ReleaseAndroid◦ Visual StudioAPK◦ Xamarin

https://developer.xamarin.com/guides/android/deployment,_testing,_and_metrics/publishing_an_application/_signing_the_android_application_package/

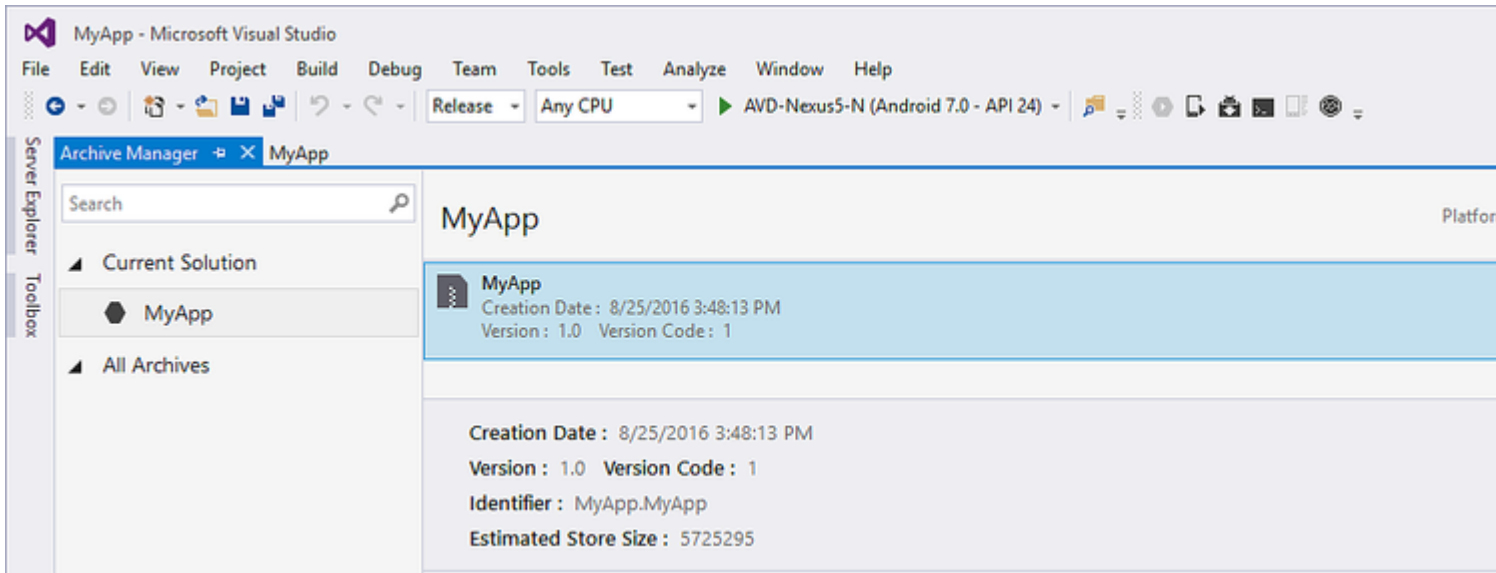
APKXamarin.Android...



APK。

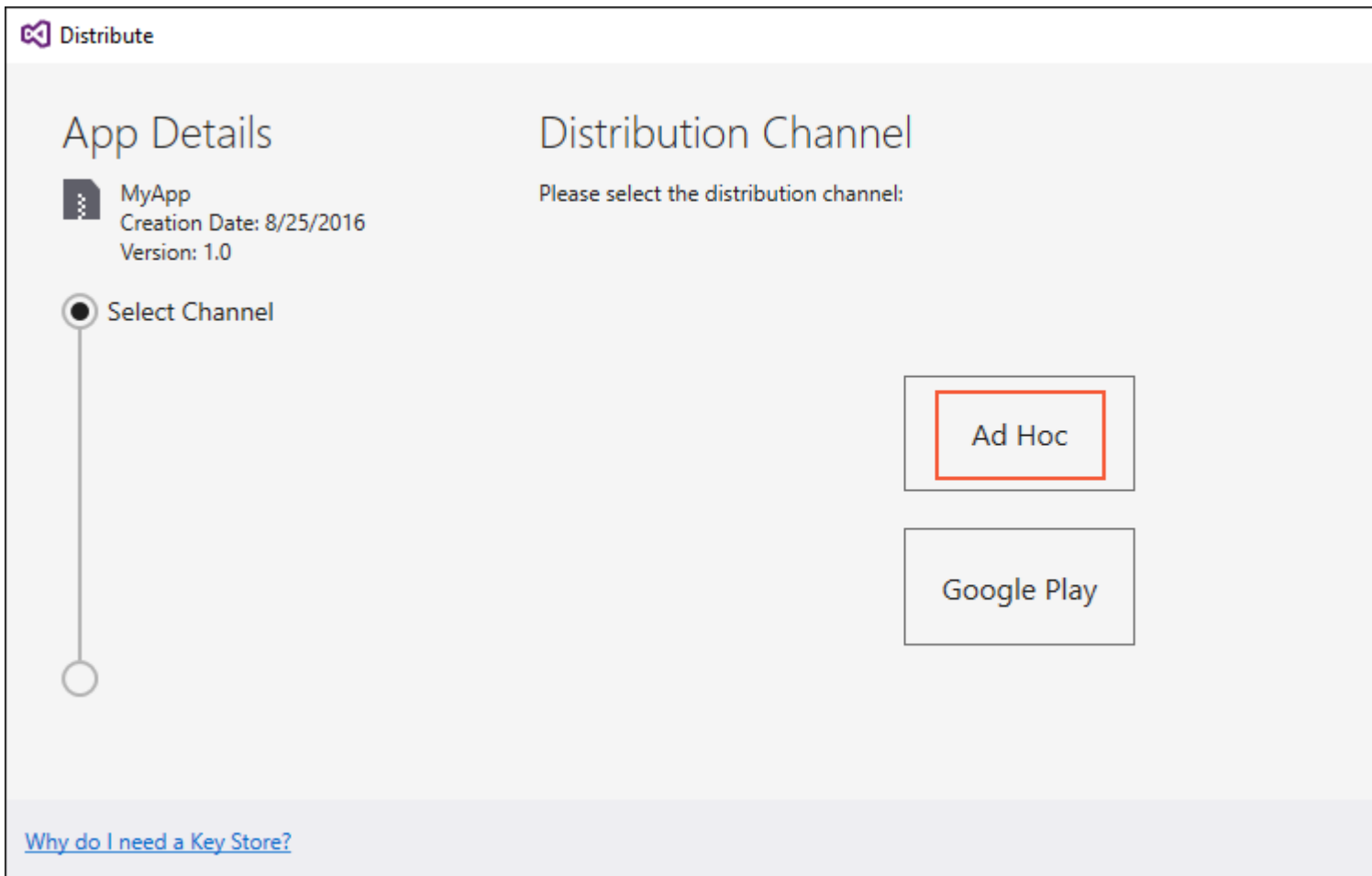


“...”。



DistributeAd-hocGoogle Play。APK。Google Play。

APK。



AndroidAPK。...;+Android。

App Details



MyApp

Creation Date: 8/25/2016

Version: 1.0



Select Channel

Ad Hoc



Signing Identity



Signing Identity

Search

Name	Expiration
chimp	Sat Aug 18 15:59:13 PDT 2046



Import...

Specify a Time Stamping Authority:

[Why do I need a Key Store?](#)

Back

Save As

Android Key Store

Android Key Store

Create Android Key Store

Alias:

Password: Confirm:

Validity: (Years)

Enter at least one of the following:

Full Name:

Organizational Unit:

Organization:

City or Locality:

State or Province:

Country Code: (2 digits)

[What is a Key Store?](#)

APK”。

App Details



MyApp

Creation Date: 8/25/2016

Version: 1.0



Select Channel

Ad Hoc



Signing Identity

Signing Identity

Search

Name	Expiration
chimp	Sat Aug 18 15:59:13 PDT 2046



Import...

Specify a Time Stamping Authority:

[Why do I need a Key Store?](#)

Back

Save As

typhon-dev > Documents >

Search Documents

Organize

New folder

Quick access

Downloads

Desktop

Documents

Name

Date modified

Type

Size

Visual Studio

8/25/2016 2:36 PM

File folder

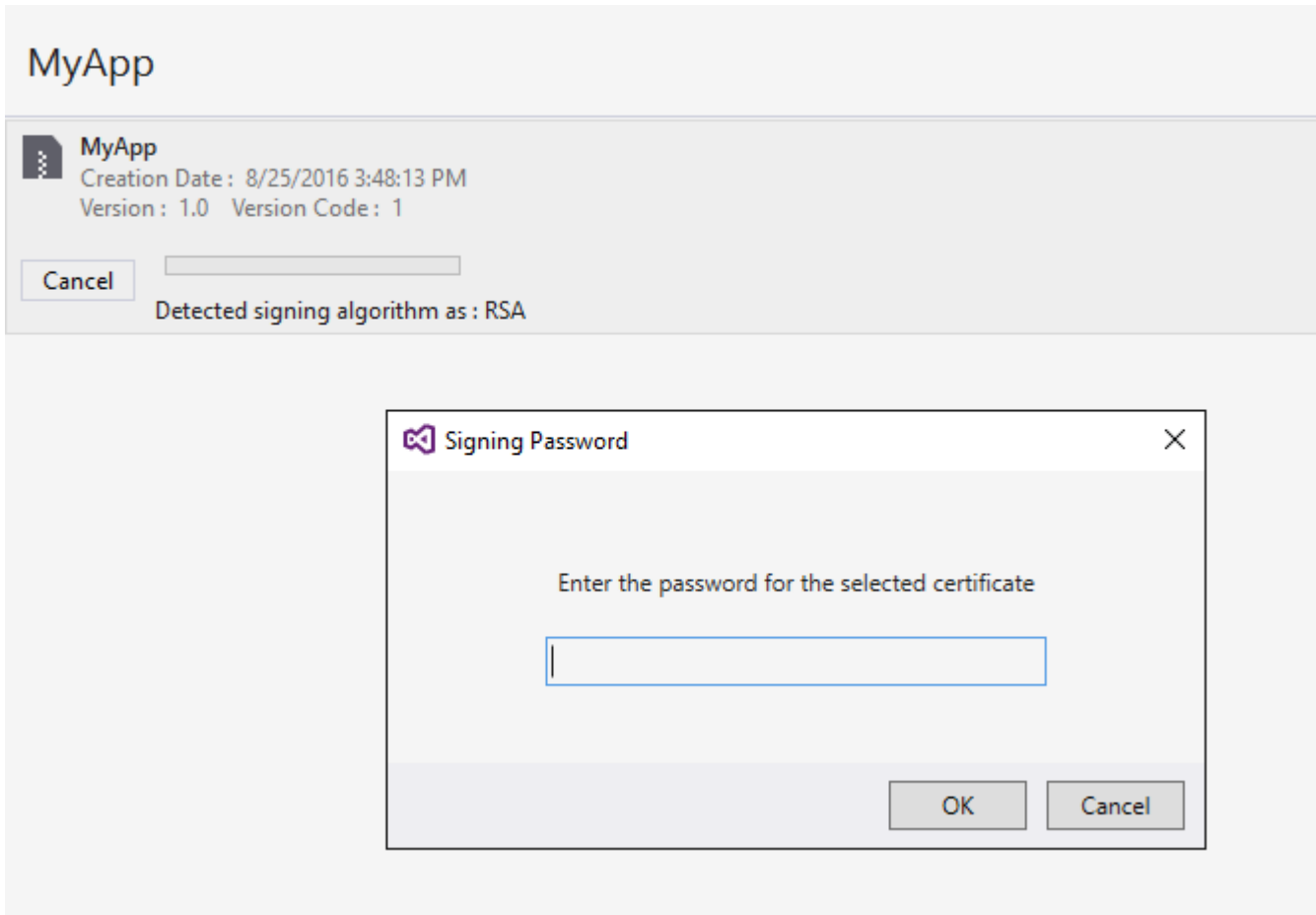
File name: MyApp.MyApp.apk

Save as type: Output APK file (.apk) (*.apk)

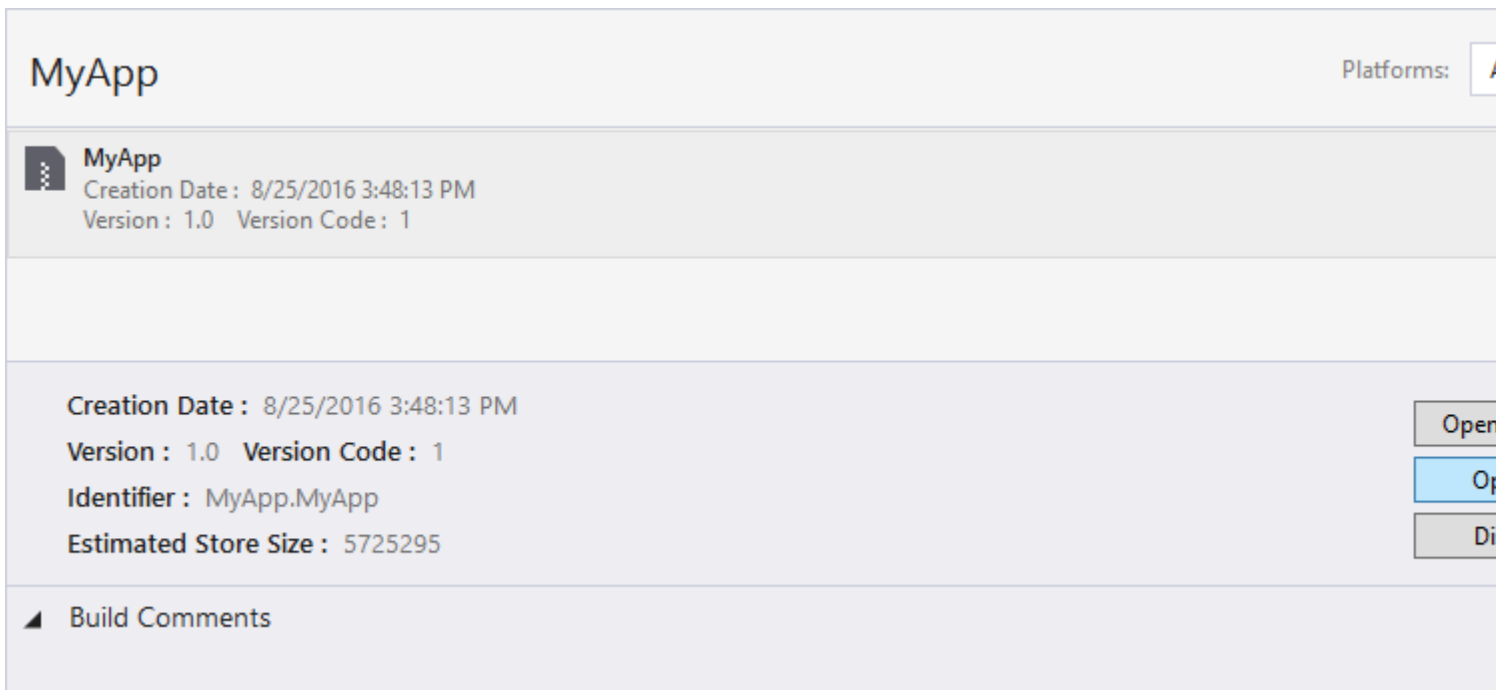
Hide Folders

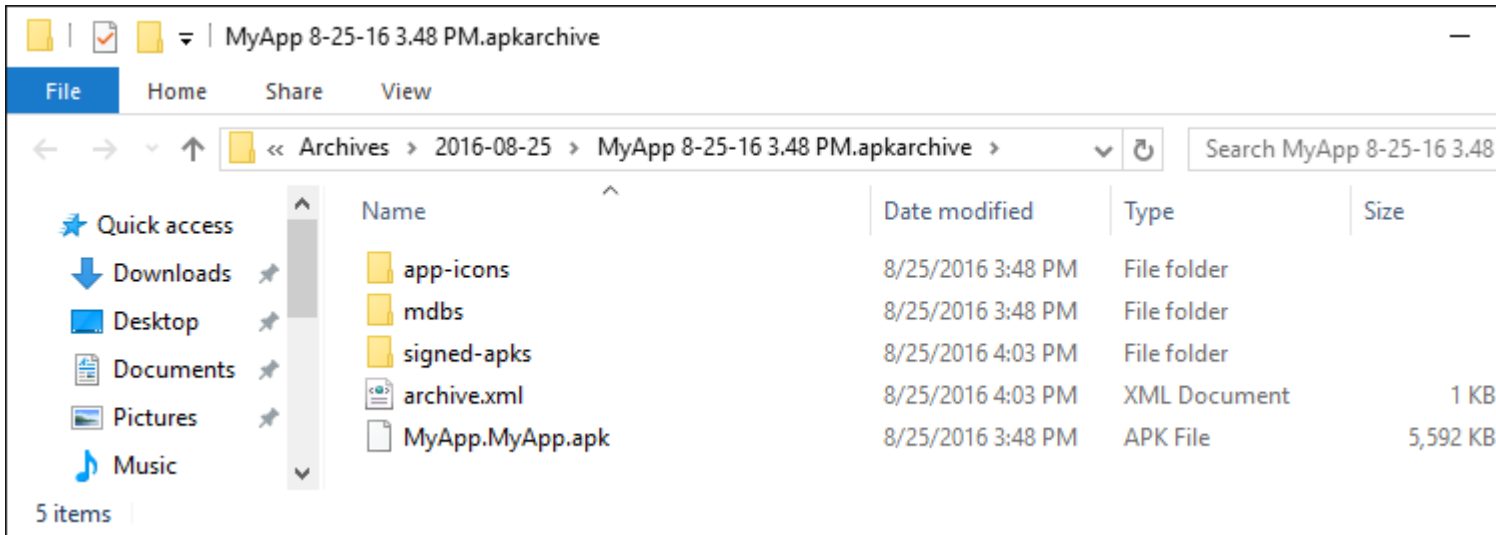
Save

Cancel



“””APK。





Xamarin.Android APKMultiDex

MultiDexAndroid APK65,536。

Android APKDalvik.dexJava。 .dex65,5362 ^ 16。

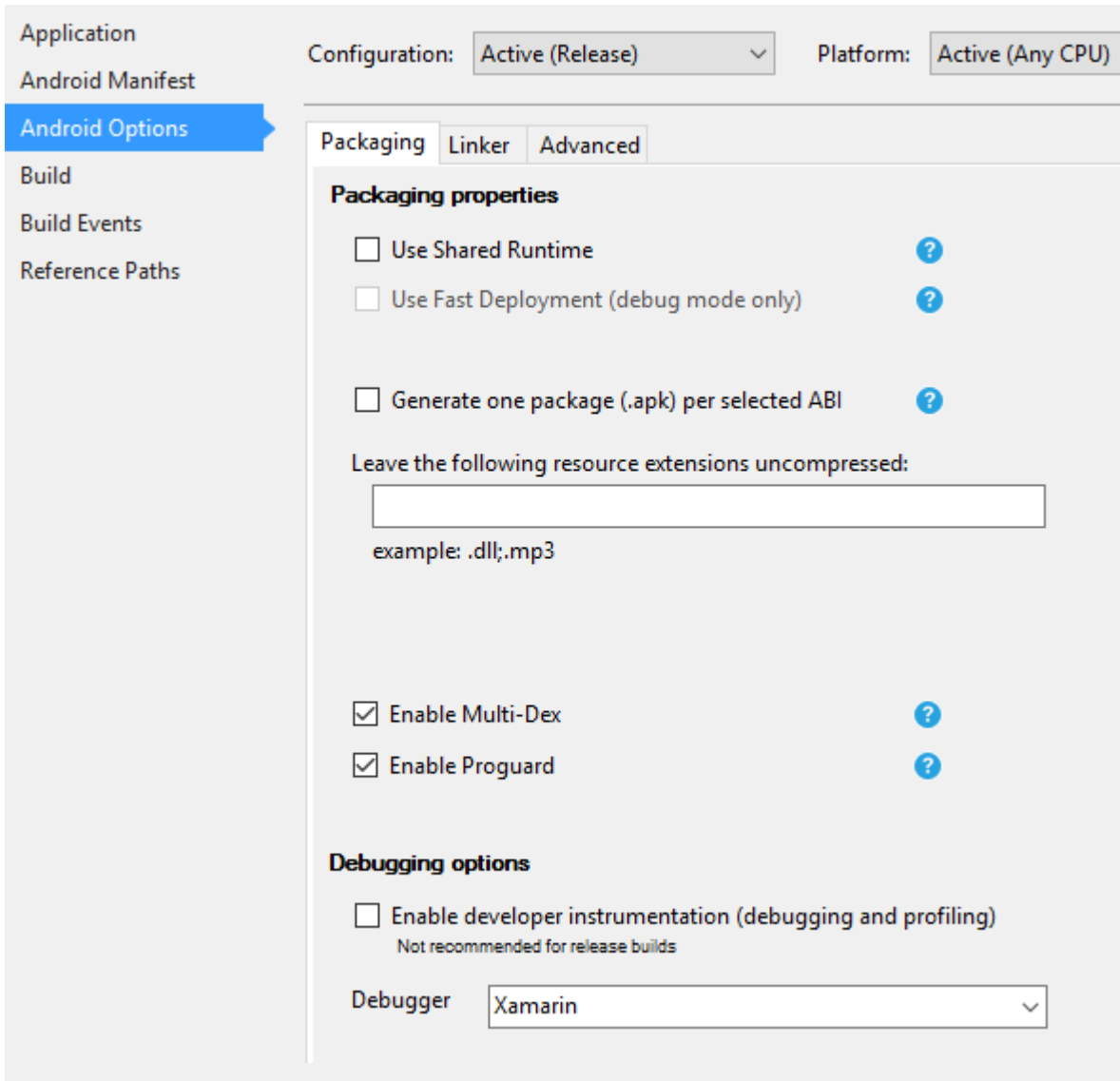
Android 5.0 LollipopAPI 21Android OSDalvikAPK.dexAPK65,536。 Android 5.0AndroidARTAPK .dex。

API 21Android65kMultiDex。 MultiDexclasses.dexclasses.dexclasses2.dexclasses3.dex...。 MultiDexApplication.dex。

AndroidAPI 21Android 5.0 LollipopSDKMultiDex.dex。 Android/MultiDex。

Xamarin.AndroidMultiDex

Xamarin.AndroidMultiDex - > Android - > - >Multi-Dex



MultiDexApplication. .. ->Shift + Alt + C. SampleXamarin.Android.

```
using System;
using Android.App;
using Android.Runtime;
using Java.Interop;

namespace Sample
{
    [Register("android/support/multidex/MultiDexApplication", DoNotGenerateAcw = true)]
    public class MultiDexApplication : Application
    {
        internal static readonly JniPeerMembers _members =
            new XAPeerMembers("android/support/multidex/MultiDexApplication", typeof(
MultiDexApplication));

        internal static IntPtr java_class_handle;

        private static IntPtr id_ctor;

        [Register(".ctor", "()V", "", DoNotGenerateAcw = true)]
        public MultiDexApplication()
            : base(IntPtr.Zero, JniHandleOwnership.DoNotTransfer)
        {
```

```

        if (Handle != IntPtr.Zero)
            return;

        try
        {
            if (GetType() != typeof (MultiDexApplication))
            {
                SetHandle(
                    JNIEnv.StartCreateInstance(GetType(), "()V"),
                    JniHandleOwnership.TransferLocalRef);
                JNIEnv.FinishCreateInstance(Handle, "()V");
                return;
            }

            if (id_ctor == IntPtr.Zero)
                id_ctor = JNIEnv.GetMethodID(class_ref, "<init>", "()V");
            SetHandle(
                JNIEnv.StartCreateInstance(class_ref, id_ctor),
                JniHandleOwnership.TransferLocalRef);
            JNIEnv.FinishCreateInstance(Handle, class_ref, id_ctor);
        }
        finally
        {
        }
    }

    protected MultiDexApplication(IntPtr javaReference, JniHandleOwnership transfer)
        : base(javaReference, transfer)
    {
    }

    internal static IntPtr class_ref
    {
        get { return JNIEnv.FindClass("android/support/multidex/MultiDexApplication", ref
java_class_handle); }
    }

    protected override IntPtr ThresholdClass
    {
        get { return class_ref; }
    }

    protected override Type ThresholdType
    {
        get { return typeof (MultiDexApplication); }
    }
}
}

```

◦

Visual Studio for WindowsAndroid SDKclasses.dex◦

Android SDKbuild-toolsAndroid SDK

C:\AndroidSDK\23.0.3\

C:\AndroidSDK\24.0.1\

mainClassesDex.batclasses.dex。 NotepadNotepad ++mainClassesDex.bat

```
if DEFINED output goto redirect
call "%java_exe%" -Djava.ext.dirs="%frameworkdir%" com.android.multidex.MainDexListBuilder
"%disableKeepAnnotated%" "%tmpJar%" "%params%"
goto afterClassReferenceListBuilder
:redirect
call "%java_exe%" -Djava.ext.dirs="%frameworkdir%" com.android.multidex.MainDexListBuilder
"%disableKeepAnnotated%" "%tmpJar%" "%params%" 1>"%output%"
:afterClassReferenceListBuilder
```

```
SET params=%params:'=%
if DEFINED output goto redirect
call "%java_exe%" -Djava.ext.dirs="%frameworkdir%" com.android.multidex.MainDexListBuilder
%disableKeepAnnotated% "%tmpJar%" %params%
goto afterClassReferenceListBuilder
:redirect
call "%java_exe%" -Djava.ext.dirs="%frameworkdir%" com.android.multidex.MainDexListBuilder
%disableKeepAnnotated% "%tmpJar%" %params% 1>"%output%"
:afterClassReferenceListBuilder
```

。

mainClassesDex.bat。

MultiDexXamarin.Android。

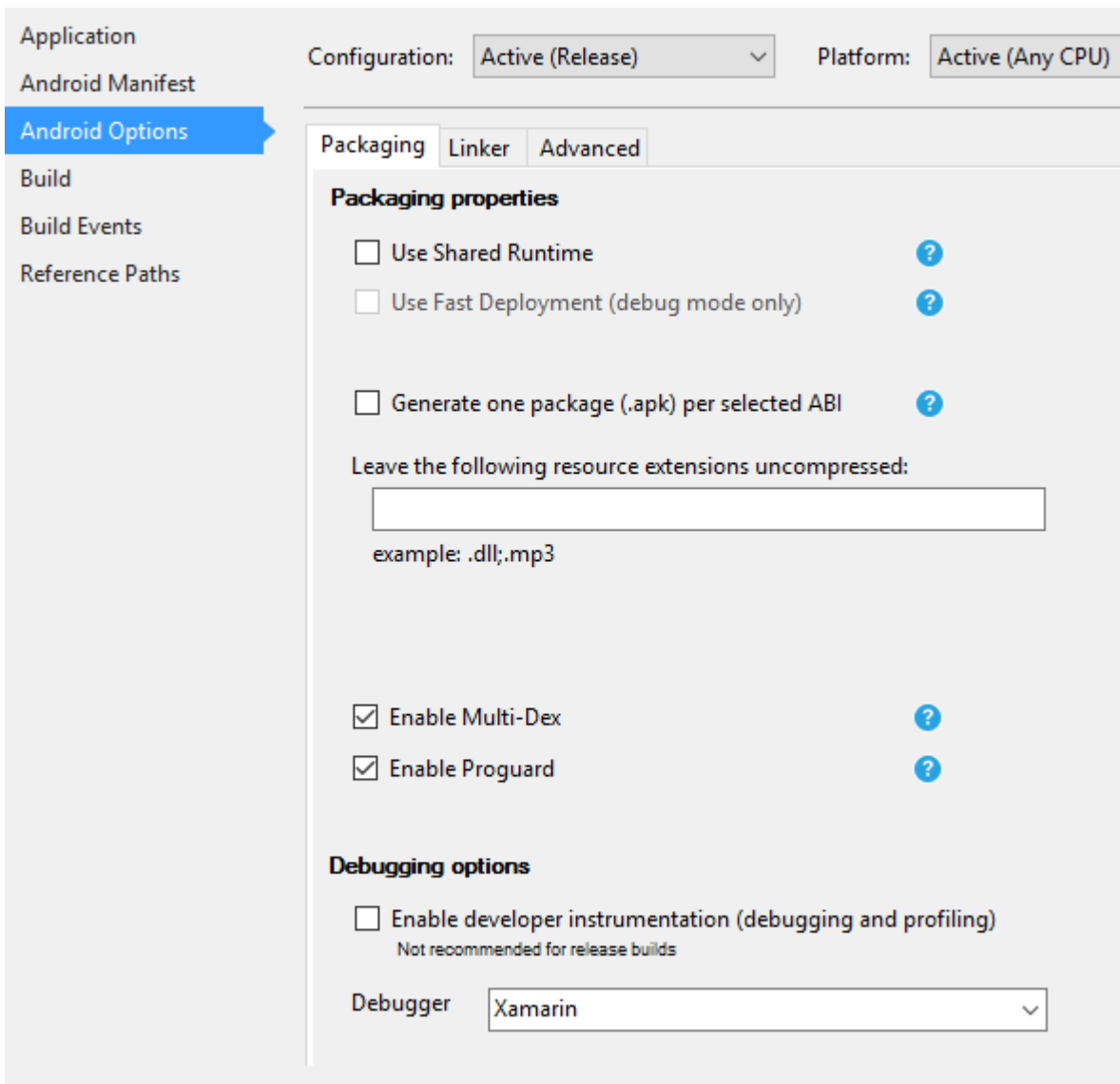
Xamarin.Android APKProGuard

[ProGuardAPKJava](#)。 [ProGuardAPK](#)。

[ProGuardXamarin.AndroidAPKJava](#)。 [ProGuardJava](#)。 [.NETDotfuscator](#)。

Xamarin.AndroidProGuard

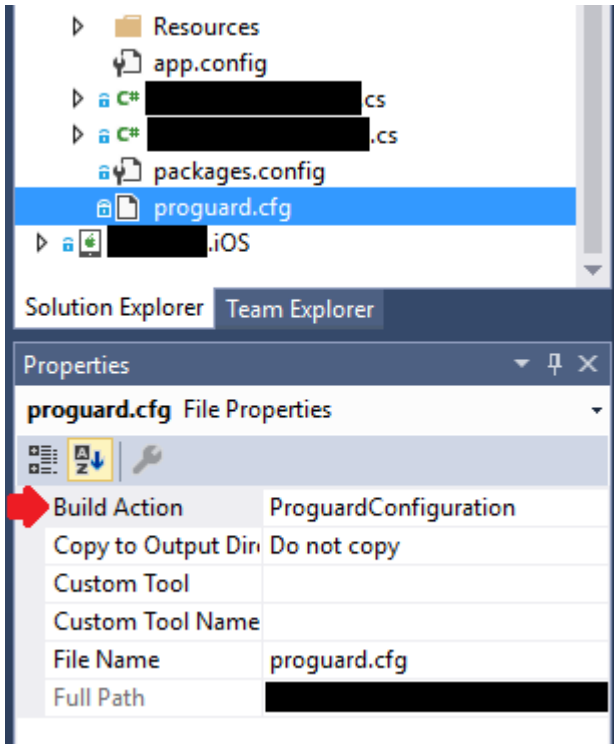
Xamarin.AndroidProGuard - > Android - > - >ProGuard



ProGuard。

Xamarin.AndroidProGuardproguard_project_primary.cfg proguard_project_references.cfg
 proguard_xamarin.cfgojb/Debug/proguardobj/Release/proguard。 ProGuardXamarin。

ProGuard/proguard.cfg .cfgBuild ActionProguardConfiguration



ProGuard -dontwarn -dontwarn -keep class ◦

20174Android SDKProGuardJava 1.8◦

```
Error
Can't read [C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\MonoAndroid\v7.0\mono.android.jar]
(Can't process class [android/app/ActivityTracker.class] (Unsupported class version number
[52.0] (maximum 51.0, Java 1.7))) [CREATEMULTIDEXMAININDEXCLASSLIST]
```

◦

ProGuard .zip android-sdk\tools\proguard\ ◦ ProGuard◦

ProGuardXamarin.Android◦

ProGuardLinker“”

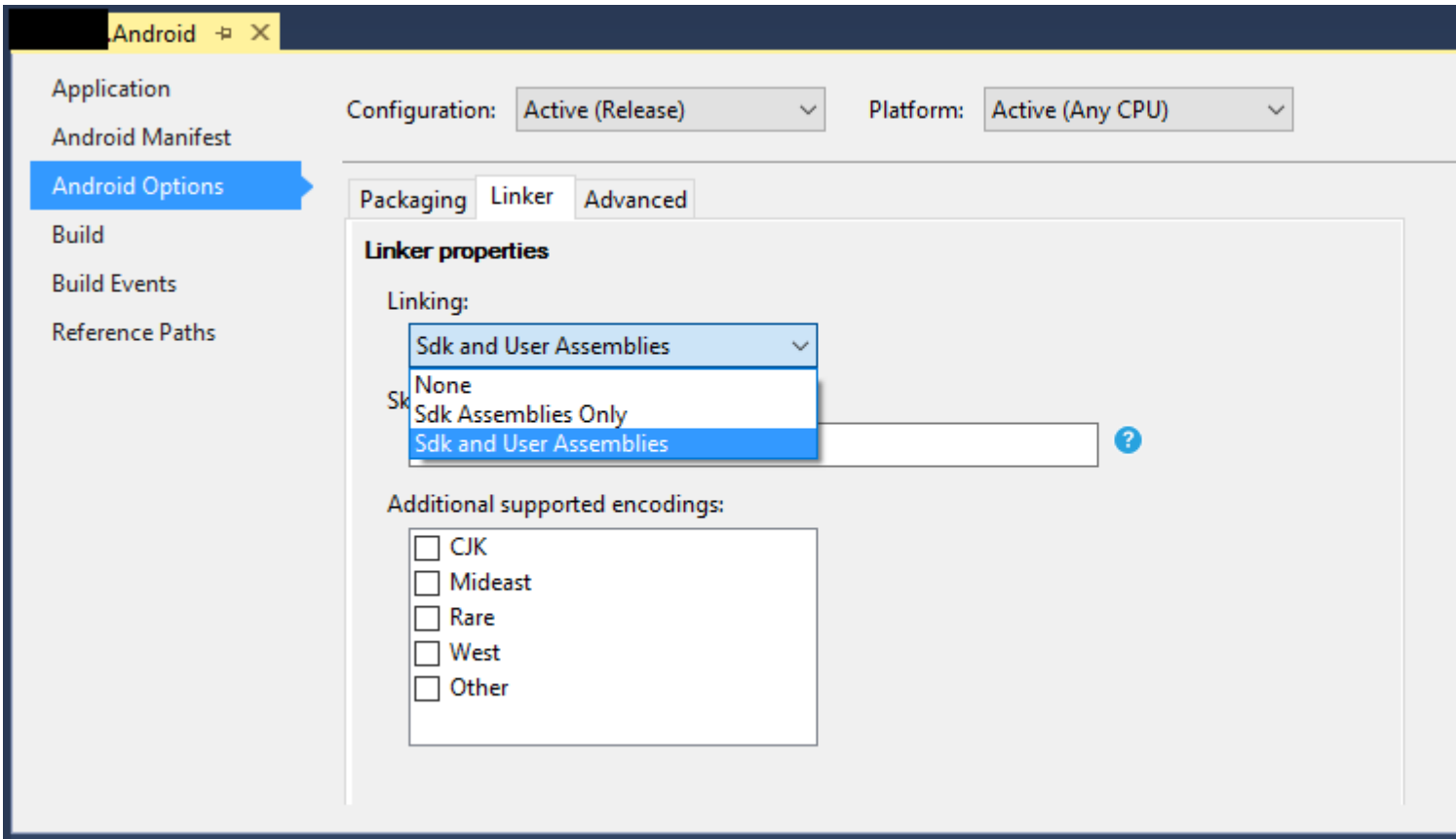
Debug◦

◦ MultiDexProGuardLinker◦

ProGuardLinker◦

Xamarin.Linker

Xamarin.Linker.NETJava ◦ - > Android - >



o

Sdk Xamarin.LinkerXamarin。

Sdk Xamarin.LinkerXamarinPCLXamarinNuGet。

SdkXamarin.Linker。

Xamarin.Linker3

1. ""Sdk”;
2. ;
3. “”。

2.

Xamarin.LinkerNuGet [Octokit](#) Internet

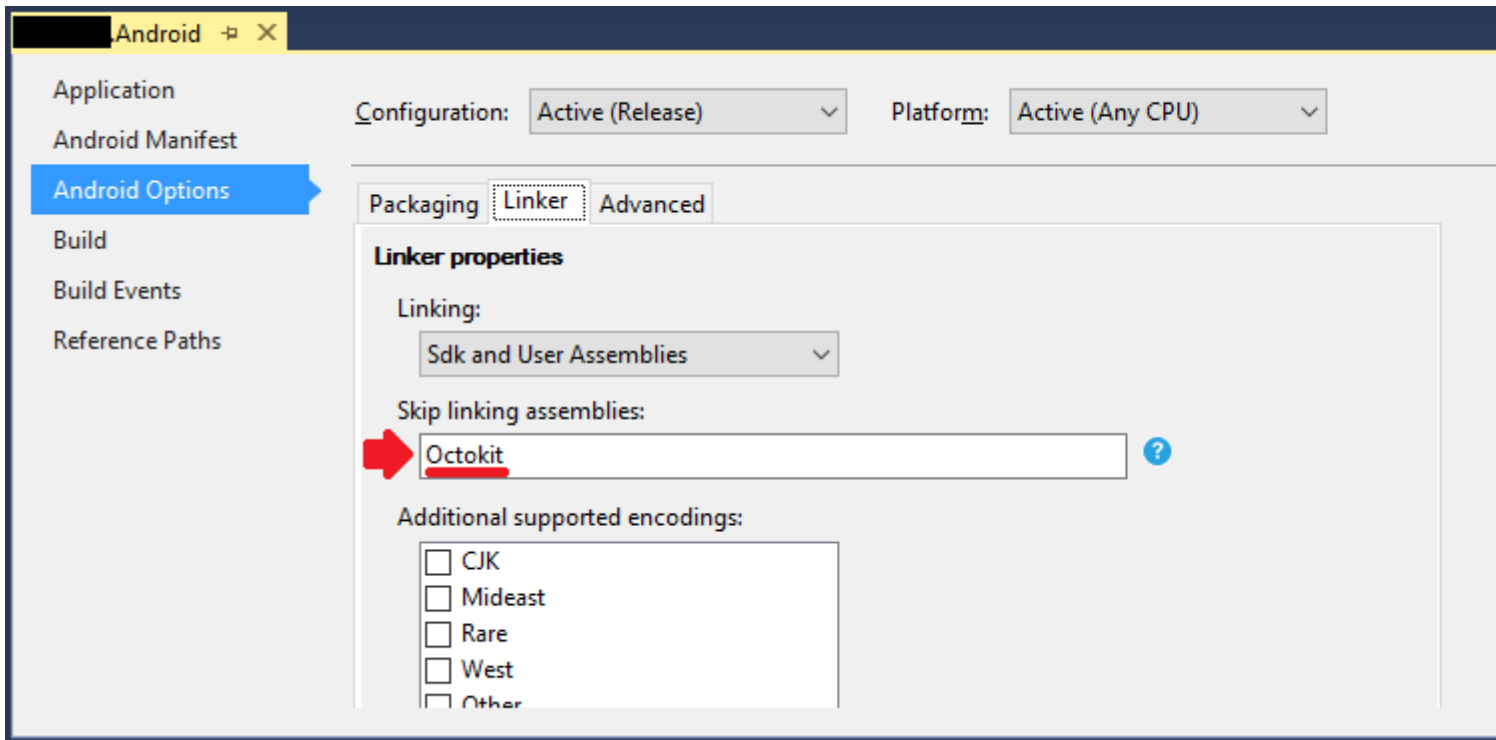
```
[0:] ERROR
[0:] SOURCE: mscorlib
[0:] MESSAGE: Object reference not set to an instance of an object.
[0:] STACK TRACE:   at Octokit.PocoJsonSerializerStrategy.DeserializeObject (System.Object
value, System.Type type) [0x003d8] in D:\repos\octokit.net\Octokit\SimpleJson.cs:1472
   at Octokit.Internal.SimpleJsonSerializer+GitHubSerializerStrategy.DeserializeObject
(System.Object value, System.Type type) [0x001c3] in
D:\repos\octokit.net\Octokit\Http\SimpleJsonSerializer.cs:165
   at Octokit.SimpleJson.DeserializeObject (System.String json, System.Type type,
Octokit.IJsonSerializerStrategy jsonSerializerStrategy) [0x00007] in
D:\repos\octokit.net\Octokit\SimpleJson.cs:583
```

```

at Octokit.SimpleJson.DeserializeObject[T] (System.String json,
Octokit.IJsonSerializerStrategy jsonSerializerStrategy) [0x00000] in
D:\repos\octokit.net\Octokit\SimpleJson.cs:595
at Octokit.Internal.SimpleJsonSerializer.Deserialize[T] (System.String json) [0x00000] in
D:\repos\octokit.net\Octokit\Http\SimpleJsonSerializer.cs:21
at Octokit.Internal.JsonHttpPipeline.DeserializeResponse[T] (Octokit.IResponse response)
[0x000a7] in D:\repos\octokit.net\Octokit\Http\JsonHttpPipeline.cs:62
at Octokit.Connection+<Run>d__54`1[T].MoveNext () [0x0009c] in
D:\repos\octokit.net\Octokit\Http\Connection.cs:574
--- End of stack trace from previous location where exception was thrown ---

```

Skip links assemblies - > Properties - > Android Options - > Linker



3.Preserve

Xamarin.Linker

“”

PreserveAttribute.cs

PreserveAttribute.cs

```

namespace My_App_Core.Models
{
    public sealed class PreserveAttribute : System.Attribute
    {
        public bool AllMembers;
        public bool Conditional;
    }
}

```


Preserve

Country.cs

```
using System;
using System.Collections.Generic;

namespace My_App_Core.Models
{
    [Preserve(AllMembers = true)]
    public class Country
    {
        public String name { get; set; }
        public String ISOcode { get; set; }

        [Preserve(AllMembers = true)]
        public Country(String name, String ISOCode)
        {
            this.name = name;
            this.ISOCode = ISOCode;
        }
    }
}
```

。

ProGuard

ProGuardJava。

ProGuard。 AndroidVisual StudioProGuard。

ProGuardAXMLAndroid.Support.V7.Widget.FitWindowsLinearLayout

Android.Support.Design.Widget.AppBarLayout。 JavaClassNotFoundException

layout_activitymain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activitymain_drawerlayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true" <!-- ### HERE ### -->
    tools:openDrawer="start">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true">
        <!-- ### HERE ## -->
        <android.support.design.widget.AppBarLayout
            android:id="@+id/activitymain_appbarlayout"
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:theme="@style/AppTheme.AppBarOverlay">  
...
```

SetContentViewLogCat

The screenshot shows the Android Studio interface. The top toolbar includes icons for DDMS, File Explorer, and Emulator. The 'Devices' panel on the left shows a virtual device named 'com.android.deskclock'. The 'LogCat' panel at the bottom displays a stack trace for a fatal exception. The exception is a `java.lang.RuntimeException` with the message 'Unable to start activity ComponentInfo{com. [REDACTED] / com. [REDACTED].activitymain}: android.view.InflateException: Binary XML file line #17: Error inflating class android.support.v7.widget.FitWindowsLinearLayout'. The stack trace includes the following methods:

Tag	Text
AndroidRun...	FATAL EXCEPTION: main
AndroidRun...	java.lang.RuntimeException: Unable to start activity ComponentInfo{com. [REDACTED] / com. [REDACTED].activitymain}: android.view.InflateException: Binary XML file line #17: Error inflating class android.support.v7.widget.FitWindowsLinearLayout
AndroidRun...	at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2059)
AndroidRun...	at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2084)
AndroidRun...	at android.app.ActivityThread.access\$600(ActivityThread.java:130)
AndroidRun...	at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:1195)

ProGuard

```
-keep public class android.support.v7.widget.FitWindowsLinearLayout  
-keep public class android.support.design.widget.AppBarLayout
```

。

ProGuard

ProGuard。 。

PicassoProGuard okio.Okio: can't find referenced class (...) can't write resource [META-INF/MANIFEST.MF] (Duplicate zip entry [okhttp.jar:META-INF/MANIFEST.MF]) (...)

Xamarin.Android APK <https://riptutorial.com/zh-CN/xamarin-android/topic/9601/xamarin-android-apk>

6: XamarinZXing

Zxing。 Zxingjava.Netxamarin。 ◦ <http://zxingnet.codeplex.com/>

library。

1ZXing.Net.Mobile。

MobileBarcodeScanner。

3。

Examples

```
button.Click +=async delegate
{
var MScanner = new MobileBarcodeScanner();
var Result = await MScanner.Scan();
if(Result == null)
{
return;
}
//get the bar code text here
string BarcodeText = Result.text;
}
```

XamarinZXing <https://riptutorial.com/zh-CN/xamarin-android/topic/9526/xamarinzxing>

7: Android

1. GitHub

<https://github.com/Daniel-Krzyczkowski/XamarinAndroid/tree/master/AndroidPictureOrientation/PictureOrientationApp>

2. Xamarin Mobile

<https://components.xamarin.com/view/xamarin.mobile>

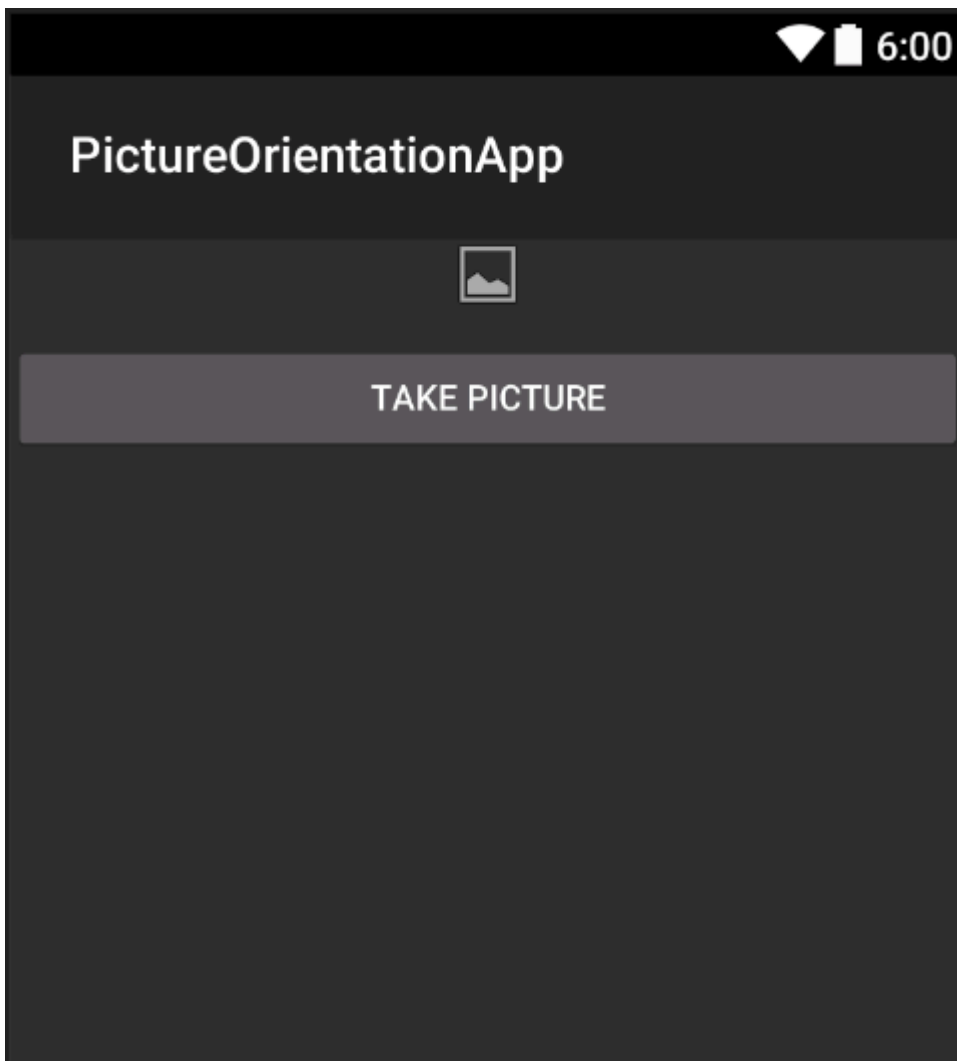
Examples

Android

Android。

imageview。

1. “TakePictureButton”“TakenPictureImageView”imageview



2.

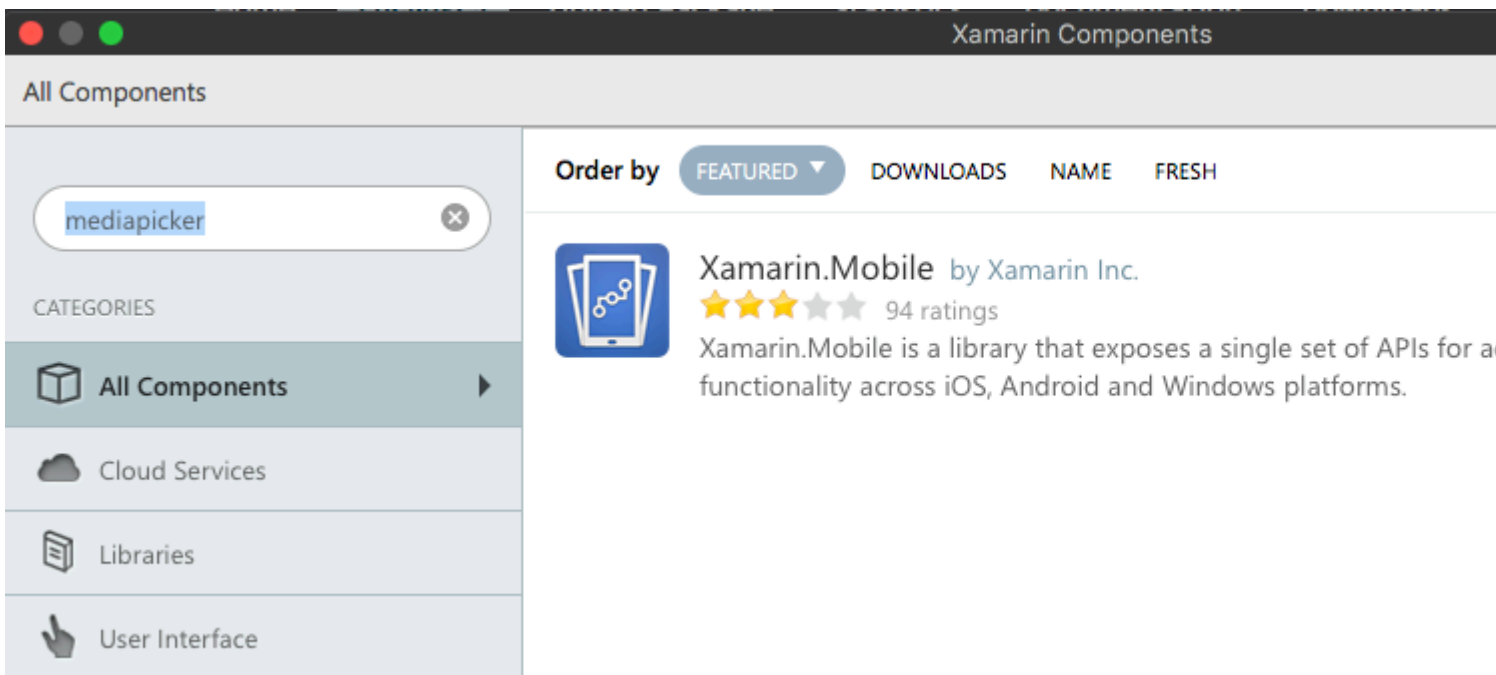
```
ImageView _takenPictureImageView;
Button _takePictureButton;

protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    SetContentView(Resource.Layout.Main);

    _takenPictureImageView = FindViewById<ImageView>(Resource.Id.TakenPictureImageView);
    _takePictureButton = FindViewById<Button>(Resource.Id.TakePictureButton);

    _takePictureButton.Click += delegate
    {
        takePicture();
    };
}
```

3. Components StoreXamarin Mobile



4. . .

```
void takePicture()
{
    var picker = new MediaPicker(this);
    DateTime now = DateTime.Now;
    var intent = picker.GetTakePhotoUI(new StoreCameraMediaOptions
    {
        Name = "picture_" + now.Day + "_" + now.Month + "_" + now.Year + ".jpg",
        Directory = null
    });
    StartActivityResult(intent, 1);
}
```

5. . . exif

```

Bitmap loadAndResizeBitmap(string filePath)
{
    BitmapFactory.Options options = new BitmapFactory.Options { InJustDecodeBounds =
true };
    BitmapFactory.DecodeFile(filePath, options);

    int REQUIRED_SIZE = 100;
    int width_tmp = options.OutWidth, height_tmp = options.OutHeight;
    int scale = 4;
    while (true)
    {
        if (width_tmp / 2 < REQUIRED_SIZE || height_tmp / 2 < REQUIRED_SIZE)
            break;
        width_tmp /= 2;
        height_tmp /= 2;
        scale++;
    }

    options.InSampleSize = scale;
    options.InJustDecodeBounds = false;
    Bitmap resizedBitmap = BitmapFactory.DecodeFile(filePath, options);

    ExifInterface exif = null;
    try
    {
        exif = new ExifInterface(filePath);
        string orientation = exif.GetAttribute(ExifInterface.TagOrientation);

        Matrix matrix = new Matrix();
        switch (orientation)
        {
            case "1": // landscape
                break;
            case "3":
                matrix.PreRotate(180);
                resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
                matrix.Dispose();
                matrix = null;
                break;
            case "4":
                matrix.PreRotate(180);
                resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
                matrix.Dispose();
                matrix = null;
                break;
            case "5":
                matrix.PreRotate(90);
                resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
                matrix.Dispose();
                matrix = null;
                break;
            case "6": // portrait
                matrix.PreRotate(90);
                resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
                matrix.Dispose();
                matrix = null;
                break;
        }
    }
}

```

```

        case "7":
            matrix.PreRotate(-90);
            resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
            matrix.Dispose();
            matrix = null;
            break;
        case "8":
            matrix.PreRotate(-90);
            resizedBitmap = Bitmap.CreateBitmap(resizedBitmap, 0, 0,
resizedBitmap.Width, resizedBitmap.Height, matrix, false);
            matrix.Dispose();
            matrix = null;
            break;
    }

    return resizedBitmap;
}

catch (IOException ex)
{
    Console.WriteLine("An exception was thrown when reading exif from media
file...:" + ex.Message);
    return null;
}
}

```

6. OnActivityResult

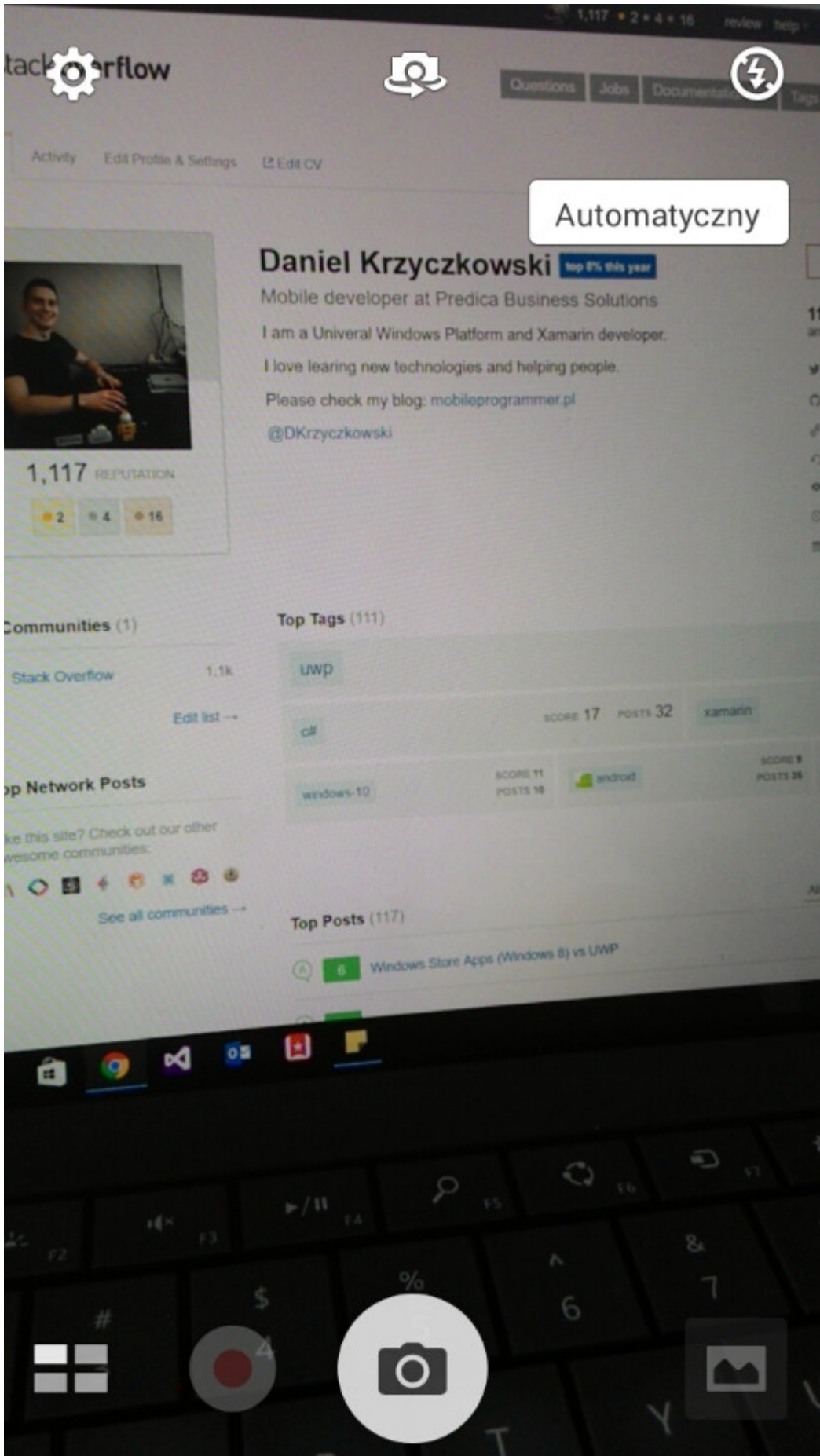
```

protected override void OnActivityResult(int requestCode, Result resultCode, Intent data)
{
    base.OnActivityResult(requestCode, resultCode, data);

    if (requestCode == 1)
    {
        if (resultCode == Result.Ok)
        {
            data.GetMediaFileExtraAsync(this).ContinueWith(t =>
            {
                using (Bitmap bmp = loadAndResizeBitmap(t.Result.Path))
                {
                    if (bmp != null)
                        _takenPictureImageView.SetImageBitmap(bmp);
                }
            }, TaskScheduler.FromCurrentSynchronizationContext());
        }
    }
}

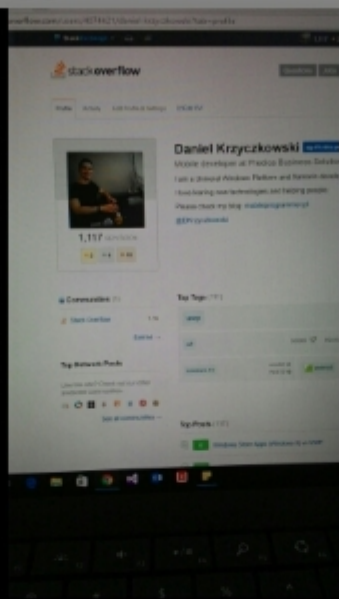
```

7. ◦



Automatyczny

PictureOrientationApp



Take picture

Android <https://riptutorial.com/zh-CN/xamarin-android/topic/6683/android>

8:

Context

AlertDialog this ◦

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

FragmentsContext ◦

```
AlertDialog.Builder builder = new AlertDialog.Builder(Context);
```

SetNeutralButton() ◦ SetPositiveButton() "" SetNegativeButton() ◦

SetCanceable(false) ◦ ◦

okcancel ◦ ◦

DialogFragment ◦

Examples

```
AlertDialog.Builder builder = new AlertDialog.Builder(Context);
builder.SetIcon(Resource.Drawable.Icon);
builder.SetTitle(title);
builder.SetMessage(message);

builder.SetNeutralButton("Neutral", (evt, args) => {
    // code here for handling the Neutral tap
});

builder.SetPositiveButton("Ok", (evt, args) => {
    // code here for handling the OK tap
});

builder.SetNegativeButton("Cancel", (evt, args) => {
    // code here for handling the Cancel tap
});

builder.SetCancelable(false);
builder.Show();
```

<https://riptutorial.com/zh-CN/xamarin-android/topic/2510/>

9:

setTitle	
setIcon	
setMessage	◦
setNegativeButtonStringEventHandler	◦
setPositiveButtonStringEventHandler	◦
setNeutralButtonStringEventHandler	◦
setOnCancelListenerIDialogInterfaceOnCancelListener	◦
setOnDismissListenerIDialogInterfaceOnDismissListener	◦
	AlertDialogDialog.Show◦

Android.App

Mono.AndroidMono.Android.dll

0.0.0.0

AlertDialog.BuilderContext -

AlertDialog◦

AlertDialog.BuilderContextInt32 -

AlertDialog◦

Material Design AlertDialog

AlertDialog

1. NuGetSupport v7 AppCompatActivity
2. AlertDialogAndroid.Support.V7.App.AlertDialog◦

```
using AlertDialog = Android.Support.V7.App.AlertDialog;
```

Examples

AlertDialog

```
// 1. Instantiate an AlertDialog.Builder with its constructor
// the parameter this is the context (usually your activity)
AlertDialog.Builder builder = new AlertDialog.Builder(this);

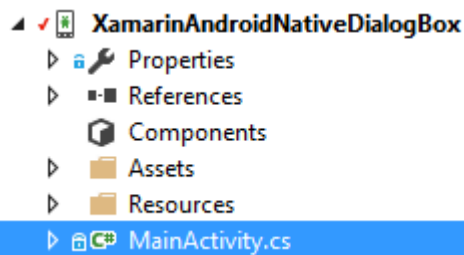
// 2. Chain together various setter methods to set the dialog characteristics
builder.SetMessage(Resource.String.dialog_message)
        .SetTitle(Resource.String.dialog_title);

// 3. Get the AlertDialog from create()
AlertDialog dialog = builder.Create();

dialog.Show();
```

Xamarin.Android

。



```
public class MainActivity : Activity
{
    int count = 1;

    protected override void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);

        // Set our view from the "main" layout resource
        SetContentView(Resource.Layout.Main);

        // Get our button from the layout resource,
        // and attach an event to it
        Button button = FindViewById<Button>(Resource.Id.MyButton);

        button.Click += delegate { button.Text = string.Format("{0} clicks!", count++); };
    }
}
```

“”””。

```
button.Click += delegate {
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.SetTitle("Specify Action");
    alert.SetMessage("Do you want to add or subtract?");
```

```
alert.SetPositiveButton("Add", (senderAlert, args) =>
{
count++;
button.Text = string.Format("{0} clicks!", count);
});

alert.SetNegativeButton("Substract", (senderAlert, args) =>
{
count--;
button.Text = string.Format("{0} clicks!", count);
});

Dialog dialog = alert.Create();
    dialog.Show();
};
```



XamarinAndroidNativeDialogBox

3 CLICKS!

Specify Action

Do you want to add or subtract?

SUBTRACT

ADD

10: - Xamarin.Android

Xamarin.AndroidAndroid。 ◦

◦ AndroidXamarinWeb。

Android

<https://developer.android.com/reference/android/app/Activity.html>

<http://www.vogella.com/tutorials/AndroidLifeCycle/article.html>

<https://github.com/xxv/android-lifecycle>

<https://developer.android.com/guide/components/fragments.html>

https://developer.xamarin.com/guides/android/platform_features/fragments/part_1_-_creating_a_fragment/

<https://developer.android.com/guide/components/activities/activity-lifecycle.html>

Examples

Android.Application

- OnCreate - MainActivity。
- OnTerminate - ◦ Android; ◦ <https://developer.android.com/reference/android/app/Application.html#onTerminate>

Xamarin.AndroidApplication。 “MyApplication.cs”

```
[Application]
public class MyApplication : Application
{
    public MyApplication(IntPtr handle, JniHandleOwnership ownership) : base(handle,
ownership)
    {
    }

    public override void OnCreate()
    {
        base.OnCreate();
    }

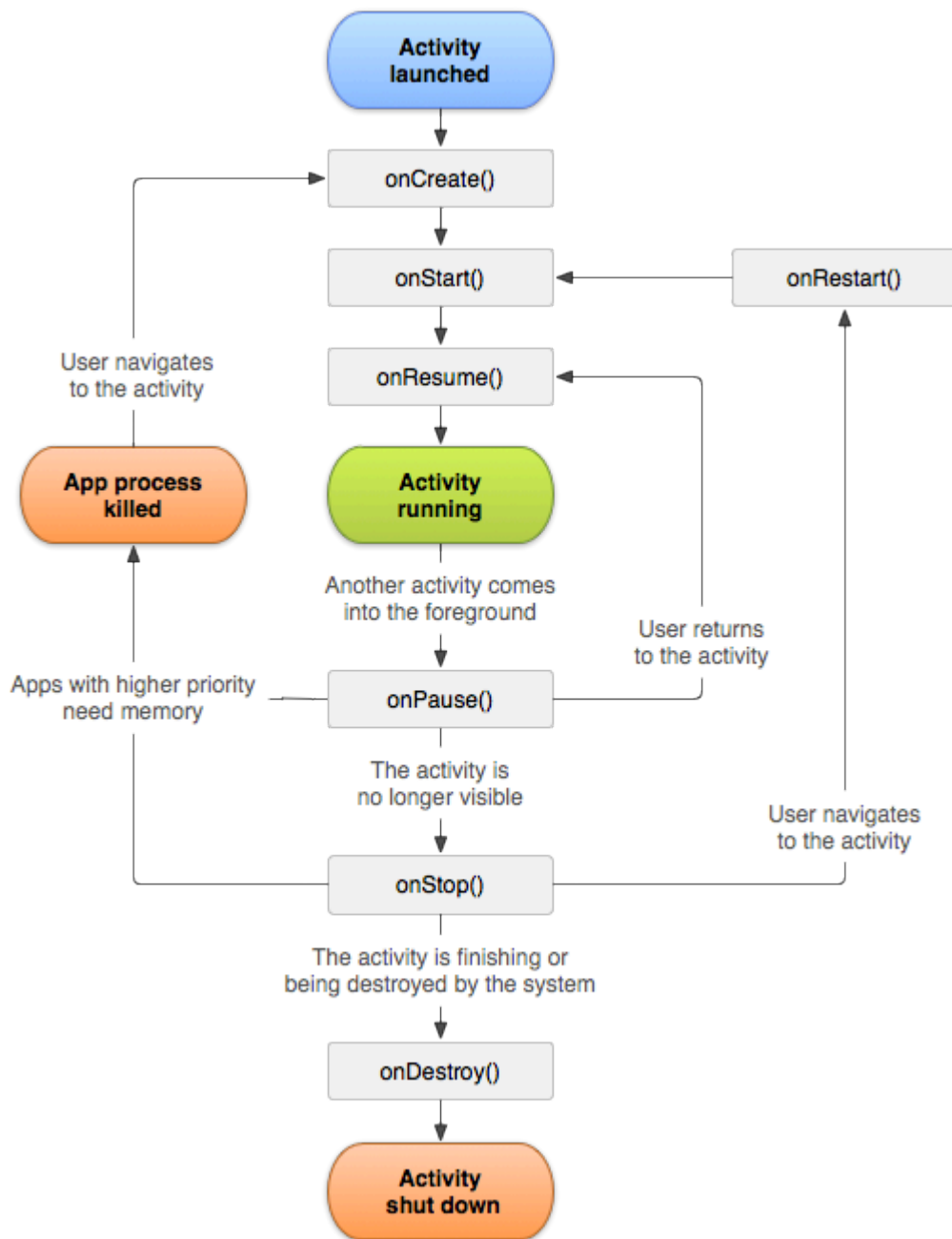
    public override void OnTerminate()
    {
        base.OnTerminate();
    }
}
```

OnCreate。 ◦

OnConfigurationChangedOnLowMemory。

◦ ActivityAndroid。

Android Activity



Activity。 Activity

```
[Activity(Label = "LifecycleApp", MainLauncher = true, Icon = "@mipmap/icon")]  
public class MainActivity : Activity  
{  
    protected override void onCreate(Bundle savedInstanceState)  
    {  
        base.onCreate(savedInstanceState);  
    }  
}
```

```

    Log.Debug("OnCreate", "OnCreate called, Activity components are being created");

    // Set our view from the "main" layout resource
    setContentView(Resource.Layout.MainActivity);
}

protected override void OnStart()
{
    Log.Debug("OnStart", "OnStart called, App is Active");
    base.OnStart();
}

protected override void OnResume()
{
    Log.Debug("OnResume", "OnResume called, app is ready to interact with the user");
    base.OnResume();
}

protected override void OnPause()
{
    Log.Debug("OnPause", "OnPause called, App is moving to background");
    base.OnPause();
}

protected override void OnStop()
{
    Log.Debug("OnStop", "OnStop called, App is in the background");
    base.OnStop();
}

protected override void OnDestroy()
{
    base.OnDestroy();
    Log.Debug("OnDestroy", "OnDestroy called, App is Terminating");
}
}

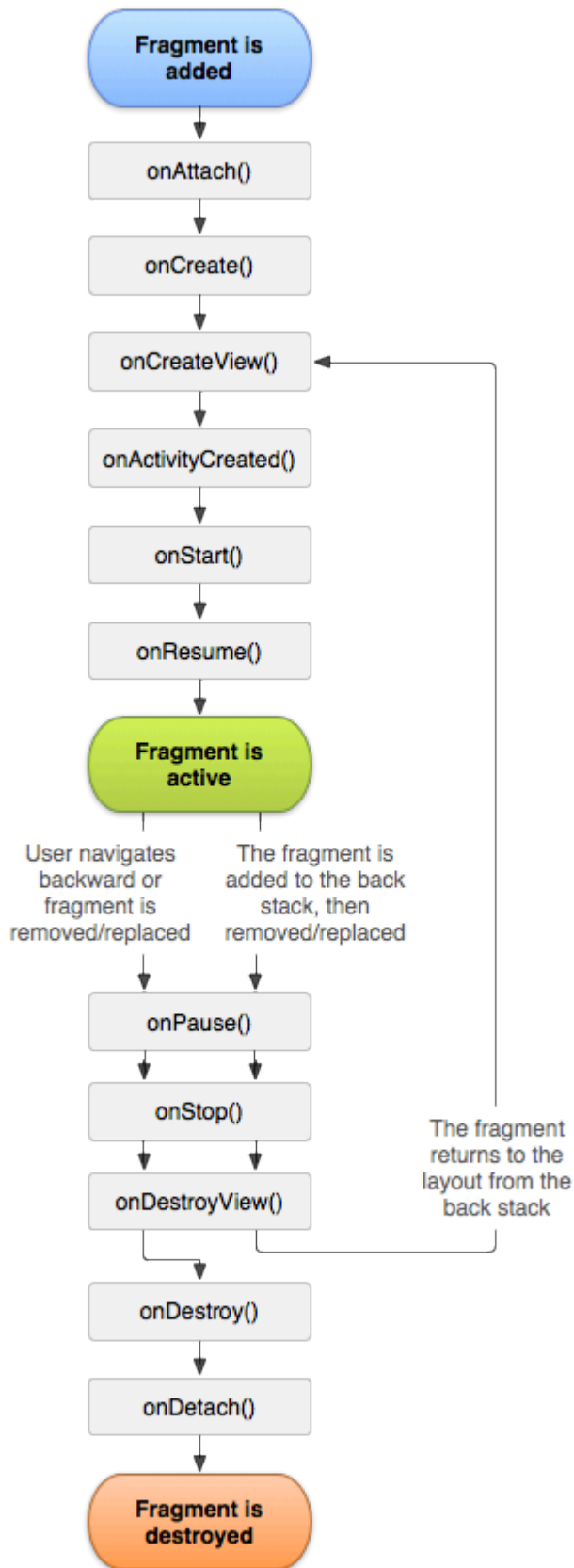
```

Android

- onCreateBundleonDestroy。 onCreate""onDestroy。 onCreateonDestroy。
- onStartonStop。 。 。 onStartBroadcastReceiverUlonStop。 onStartonStop。
- onResumeonPause。 。 - - 。

。 。

Android



Android

- onCreate - . . .
- onCreateView - . UI. UI.null.

- onPause - . . .

Xamarin.Android

```
public class MainActivity : AppCompatActivity
{
    public override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);

        // Create your fragment here
        // You should initialize essential components of the fragment
        // that you want to retain when the fragment is paused or stopped, then resumed.
    }

    public override View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
    {
        // Use this to return your custom view for this Fragment
        // The system calls this when it's time for the fragment to draw its user interface
        // for the first time.

        var rootView = inflater.Inflate(Resource.Layout.MainFragment, container, false);
        return rootView;
    }

    public override void onPause()
    {
        // The system calls this method as the first indication that the user is leaving the
        // fragment

        base.OnPause();
    }
}
```

◦

GitHub

GitHubXamarin.Android◦

-
-
-

<https://github.com/Daniel-Krzyczkowski/XamarinAndroid/tree/master/AndroidLifecycle/LifecycleApp>

- Xamarin.Android <https://riptutorial.com/zh-CN/xamarin-android/topic/8842/---xamarin-android>

11:

Examples

Toast

[MakeText\(\)](#) [Toast](#) [Context](#) [Toast](#) [Toast](#) [Show\(\)](#) [Toast](#)

```
Context context = Application.Context;
string text = "Hello toast!";
ToastLength duration = ToastLength.Short;

var toast = Toast.MakeText(context, text, duration);
toast.Show();
```

[Toast](#)

Toast

```
Toast.MakeText(Application.Context, "Hello toast!", ToastLength.Short).Show();
```

[Android](#) .

ColorMatrixColorFilter

```
Toast t = Toast.MakeText(context, message, duration);
Color c = */your color*/;
ColorMatrixColorFilter CM = new ColorMatrixColorFilter(new float[]
{
    0,0,0,0,c.R,
    0,0,0,0,c.G,
    0,0,0,0,c.B,
    0,0,0,1,0
});
t.View.Background.SetColorFilter(CM);
t.Show();
```

```
if (((float)(c.R) + (float)(c.G) + (float)(c.B)) / 3) >= 128)
    t.View.FindViewById<TextView>(Android.Resource.Id.Message).SetTextColor(Color.Black);
else
    //text color is white by default
```

SetGravity

```
//Toast at bottom left corner of screen
Toast t = Toast.MakeText(context, message, duration);
t.SetGravity(GravityFlags.Bottom | GravityFlags.Left, 0, 0);
t.Show();

//Toast at a custom position on screen
```

```
Toast t = Toast.MakeText(context, message, duration);  
t.SetGravity(GravityFlags.Top | GravityFlags.Left, x, y);  
t.Show();
```

<https://riptutorial.com/zh-CN/xamarin-android/topic/3550/>

12:

Examples

Xamarin.Android Bindings GeneratorJava^o remove-node XMLmetadata.xml

```
<remove-node path="/api/package[@name='{package_name}']/class[@name='{name}']" />
```

Java

javaView.IOnClickListenercallbacksJava.Lang.ObjectJava.Lang.Throwable^o

'MyListener'Android.Runtime.IJavaObjectJava.Lang.Object^o ^o

^o

```
class MyListener : View.IOnClickListener
{
    public IntPtr Handle { get; }

    public void Dispose()
    {
    }

    public void OnClick(View v)
    {
        // ...
    }
}
```

```
class MyListener :
    Java.Lang.Object, // this is the important part
    View.IOnClickListener
{
    public void OnClick(View v)
    {
        // ...
    }
}
```

CJava^o

C“l”Java^o JavaSomeInterfaceISomeInterface^o

JavaC^o Java gettersetter^o Java

```
public int getX() { return someInt; }

public int setX(int someInt) { this.someInt = someInt; }
```



```
public int X { get; set; }
```

<https://riptutorial.com/zh-CN/xamarin-android/topic/771/>

13: ListView

Examples

o

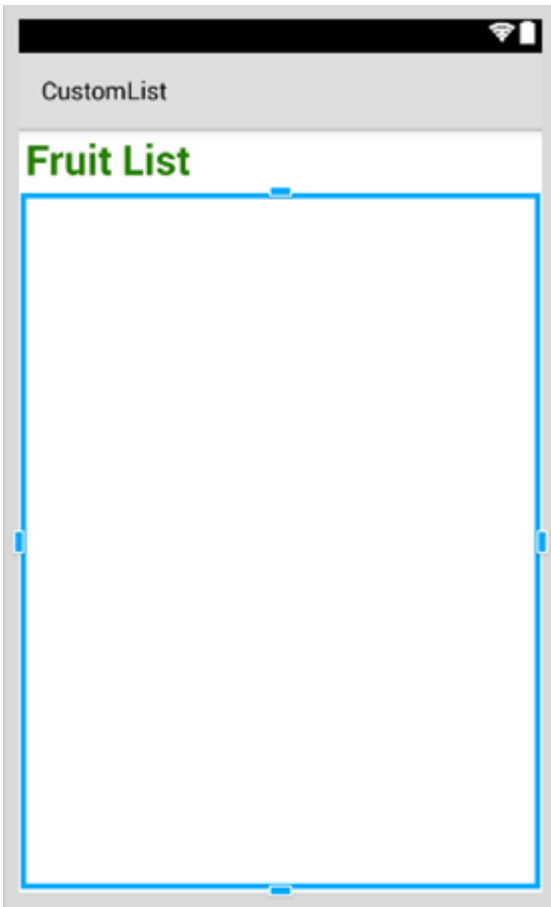


customrow.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="8dp">
    <ImageView
        android:id="@+id/Image"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_alignParentLeft="true"
        android:layout_marginRight="8dp"
        android:src="@drawable/icon" />
    <TextView
        android:id="@+id/Text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@id/Image"
        android:layout_toRightOf="@id/Image"
        android:layout_marginTop="5dip"
        android:text="This is Line1"
        android:textSize="20dip"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/Text2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/Text1"
        android:layout_marginTop="1dip"
        android:text="This is line2"
```

```
        android:textSize="15dip"
        android:layout_toRightOf="@id/Image" />
</RelativeLayout>
```

main.xml。



.....

Data.cs

```
public class Data
{
    public string Heading;
    public string SubHeading;
    public string ImageURI;

    public Data ()
    {
        Heading = "";
        SubHeading = "";
        ImageURI = "";
    }
}
```

DataAdapter.csAdapters

```
public class DataAdapter : BaseAdapter<Data> {

    List<Data> items;
```

```

Activity context;
public DataAdapter(Activity context, List<Data> items)
    : base()
{
    this.context = context;
    this.items = items;
}
public override long GetItemId(int position)
{
    return position;
}
public override Data this[int position]
{
    get { return items[position]; }
}
public override int Count
{
    get { return items.Count; }
}
public override View GetView(int position, View convertView, ViewGroup parent)
{
    var item = items[position];
    View view = convertView;
    if (view == null) // no view to re-use, create new
        view = context.LayoutInflater.Inflate(Resource.Layout.CustomRow, null);

    view.FindViewById<TextView>(Resource.Id.Text1).Text = item.Heading;
    view.FindViewById<TextView>(Resource.Id.Text2).Text = item.SubHeading;

    var imageBitmap = GetImageBitmapFromUrl(item.ImageURI);
    view.FindViewById<ImageView>(Resource.Id.Image).SetImageBitmap(imageBitmap);
    return view;
}

private Bitmap GetImageBitmapFromUrl(string url)
{
    Bitmap imageBitmap = null;
    if(! (url=="null"))
        using (var webClient = new WebClient())
        {
            var imageBytes = webClient.DownloadData(url);
            if (imageBytes != null && imageBytes.Length > 0)
            {
                imageBitmap = BitmapFactory.DecodeByteArray(imageBytes, 0,
imageBytes.Length);
            }
        }

    return imageBitmap;
}
}

```

GetView。

```

view.FindViewById<TextView>(Resource.Id.Text1).Text
view.FindViewById<TextView>(Resource.Id.Text2).Text

var imageBitmap = GetImageBitmapFromUrl(item.ImageUrl);
view.FindViewById<ImageView> (Resource.Id.image).SetImage(imageBitmap);
return view;

```

Linking the Data object
with the custom row
list view

GetImageBitmapFromUrl dataadapter.

MainActivity.cs

```

public class MainActivity : Activity
{
    ListView listView;

    protected override void OnCreate (Bundle bundle)
    {
        base.OnCreate (bundle);

        // Set our view from the "main" layout resource
        SetContentView (Resource.Layout.Main);
        listView = FindViewById<ListView>(Resource.Id.List);

        List<Data> myList = new List<Data> ();

        Data obj = new Data ();
        obj.Heading = "Apple";
        obj.SubHeading = "An Apple a day keeps the doctor away";
        obj.ImageURI =
"http://www.thestar.com/content/dam/thestar/opinion/editorials/star_s_view_/2011/10/12/an_apple_a_day_r

        myList.Add (obj);

        Data obj1 = new Data ();
        obj1.Heading = "Banana";

```

```

obj1.SubHeading = "Bananas are an excellent source of vitamin B6 ";
obj1.ImageURI =
"http://www.bbcgoodfood.com/sites/bbcgoodfood.com/files/glossary/banana-crop.jpg";

myList.Add(obj1);

Data obj2 = new Data();
obj2.Heading = "Kiwi Fruit";
obj2.SubHeading = "Kiwifruit is a rich source of vitamin C";
obj2.ImageURI = "http://www.wiffens.com/wp-content/uploads/kiwi.png";

myList.Add(obj2);

Data obj3 = new Data();
obj3.Heading = "Pineapple";
obj3.SubHeading = "Raw pineapple is an excellent source of manganese";
obj3.ImageURI =
"http://www.medicalnewstoday.com/images/articles/276/276903/pineapple.jpg";

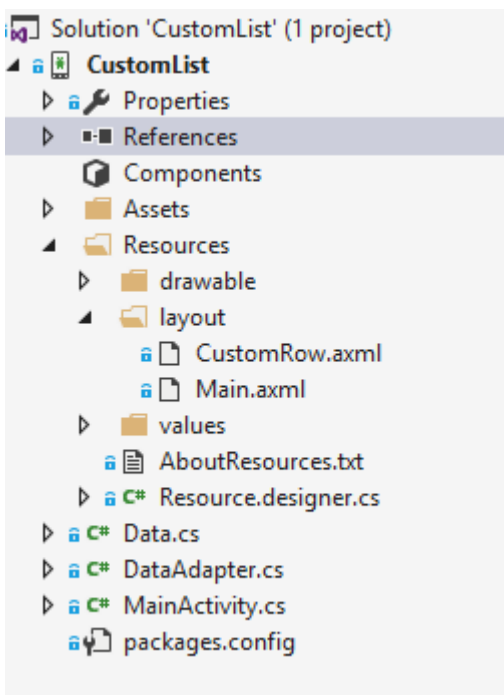
myList.Add(obj3);

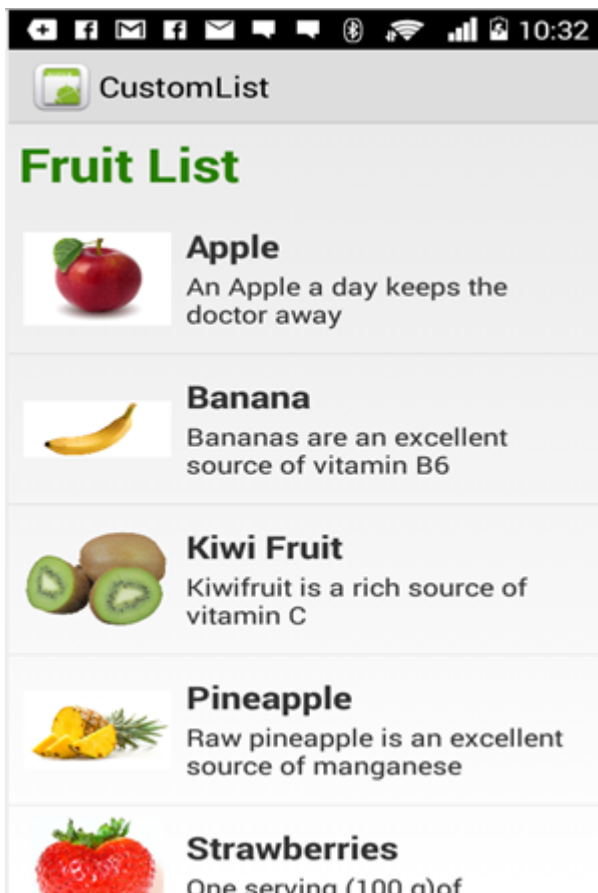
Data obj4 = new Data();
obj4.Heading = "Strawberries";
obj4.SubHeading = "One serving (100 g)of strawberries contains approximately 33
kilocalories";
obj4.ImageURI = "https://ecs3.tokopedia.net/newimg/product-
1/2014/8/18/5088/5088_8dac78de-2694-11e4-8c99-6be54908a8c2.jpg";

myList.Add (obj4);
listView.Adapter = new DataAdapter(this,myList);
}

```

o





ListView <https://riptutorial.com/zh-CN/xamarin-android/topic/6406/listview>

S. No		Contributors
1	Xamarin.Android	Amy Burns , Community , Jon Douglas , Kevin Montrose , Ryan Weaver
2	RecyclerView	Alexandre , Matthew , Ryan Alford , Sreeraj , Zverev Eugene
3	Xamarin.Android -	Daniel Krzyczkowski , tylerjgarland
4	Xamarin.Android -	Ladislav
5	Xamarin.Android APK	Alexandre
6	XamarinZXing	GvSharma
7	Android	Daniel Krzyczkowski
8		JimBobBennett , Pilatus
9	- Xamarin.Andorid	CDrosos , Daniel Krzyczkowski , Steven Mark Ford
10		GONeale , Matthew , Piet , user2912553
11		EJoshuaS , Jon Douglas , jonp , Matthew , Prashant C , Sven-Michael Stübe
12	ListView	user3814750