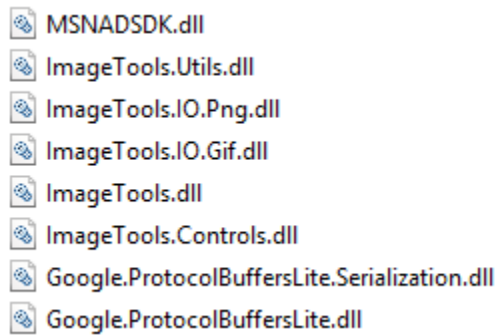


- 1) 将广告 SDK 放到本地目录,并解压



- 2) 在 Visual Studio 中找到你工程下的"Reference"目录, 右键单击, 选择"添加引用", 选择将广告 SDK 的 dll"加入工程引用目录
- 3) 配置权限

配置 WAppManifest.xml

```
<Capabilities>

    <Capability Name="ID_CAP_NETWORKING"/>

    <Capability Name="ID_CAP_IDENTITY_DEVICE"/>

    <Capability Name="ID_CAP_WEBBROWSERCOMPONENT"/>

</Capabilities>
```

如果希望能够展示点击后为打电话行为的广告,请在 WAppManifest.xml 的<Capabilities>中加入

```
<Capability Name="ID_CAP_PHONEDIALER"/>
```

同时设置广告控件的 TelCapability 为 true

*如果不加入此权限不影响广告 SDK 的使用, 但是可能会导致被投放的广告数量变少

如果希望能够获取展示点击后为在本地相册存图片(coupon)的 MRAID 广告,请在 WAppManifest.xml 的<Capabilities>中加入

```
<Capability Name="ID_CAP_MEDIALIB_PHOTO"/>
```

同时设置广告控件的 StorePic 为 true

*如果不加入此权限不影响广告 SDK 的使用

如果希望有基于位置的广告投放，请在 WManifest.xml 的<Capabilities>中加入

```
<Capability Name="ID_CAP_LOCATION"/>
```

*如果不加入此权限不影响广告 SDK 的使用，只是广告内容不会基于地理位置投放

为了提高广告的精准度，建议在 WManifest.xml 的<Capabilities>中加入

```
<Capability Name="ID_CAP_IDENTITY_USER"/>
```

*如果不加入此权限不影响广告 SDK 的使用，只是会影响广告的投放精准性

4) 在代码中使用 SDK

4.1)广告功能

在代码中可以用两种方式使用 SDK:

4.1.1)通过 xaml 使用 SDK

在要使用广告控件的页面<phone:PhoneApplicationPage>里定义广告控件的命名空间,

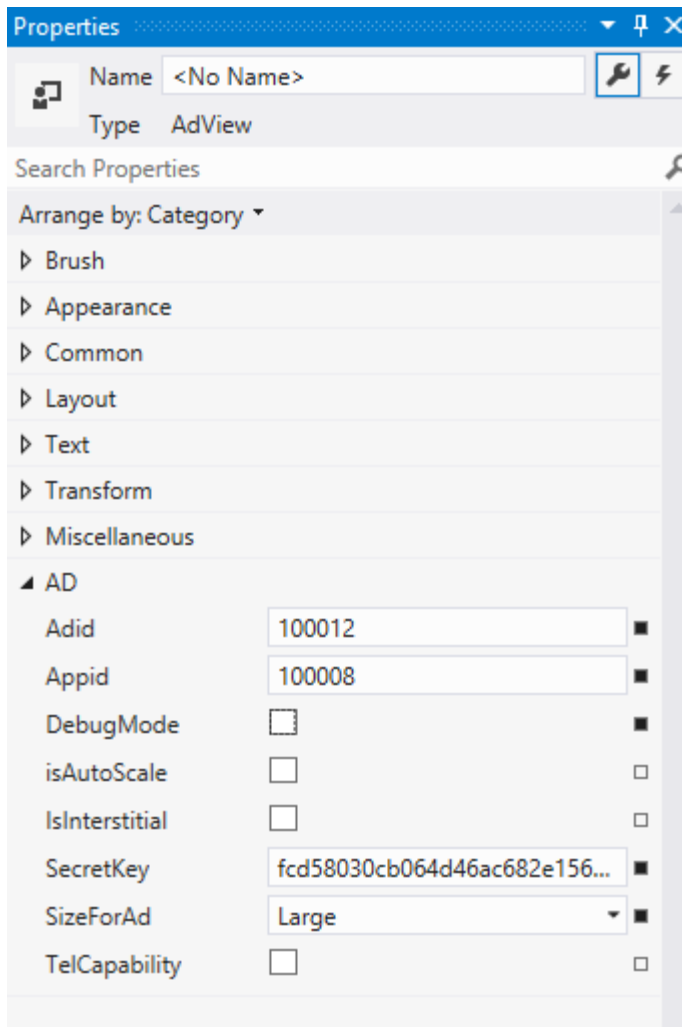
```
xmlns:msnad="clr-namespace:MSNADSDK.AD;assembly=MSNADSDK"
```

然后在页面上合适的地方加上对广告控件的使用

```
<msnad:AdView VerticalAlignment="Bottom" SizeForAd="Large"
    Appid="100008" Adid="100012" SecretKey="fcd58030cb064d46ac682e1563b5489b">

</msnad:AdView>
```

广告控件的属性也可以在 VS 的 Properties 窗口的 AD 栏下面调整



*如果需要使用插页广告，请设置广告控件的 IsInterstitial 属性为 true，并为广告控件的 Name 属性设置，在每次需要获取插页广告的时候，使用 Name 值.GetInterstitialAd()来获取插页广告。

4.1.2)通过代码使用 SDK

可以使用类型下面的代码创建广告位

```
AdView adView = new AdView();  
adView.Appid = "100008";  
adView.Adid = "100012";  
adView.SecretKey = "fcd58030cb064d46ac682e1563b5489b";  
adView.SizeForAd = AdSize.Large
```

并加入到页面合适的地方

```
adArea.Children.Add(adView);
```

*如果需要使用插页广告，请设置广告控件的 `IsInterstitial` 属性为 `true`，并在每次需要获取插页广告的时候，调用 `adView.GetInterstitialAd()`来获取插页广告。

4.2)应用统计功能

可以通过在App的`Application_Launching`方法里加上

```
MSNADSDK.Windata.SendWinData(string appId, string appVersion, string secretkey,  
string channel, AppOrentation orientation);
```

来激活 SDK 的应用相关统计功能，相关的统计结果可以在开发者后台看到。

`appid` 为应用在开发者后台注册的应用 ID

`appVersion` 是应用自身的版本号

`secretKey` 是身份识别密钥，在开发者后台注册的时候获得

`channel` 是应用的渠道，不填为默认值“Marketplace”

`orientation` 是应用的初始屏幕朝向，不填为默认值“Portrait”

5) 参数解释：

Appid:为应用在开发者后台注册的 Pub ID

Adid: 为开发者后台注册的广告位 ID

SerectKey: 身份识别密钥，在开发者后台注册的时候获得

SizeForAd: 目前提供屏幕全宽度铺满的广告(Large)， 2/3 屏幕宽度铺满的广告(Normal,Small)

TelCapability:广告是否支持打电话行为(需要同时在 `WMApManifest.xml` 中加入 `ID_CAP_PHONEDIALER` 权限)

AdRequestSuccessEvent:广告请求成功的回调

AdRequestError:广告请求失败的回调

LeavingApplication:广告点击行为要将应用推入后台时的回调（比如打开外部浏览器）

AdSdkExceptionEvent:广告 SDK 的异常输出回调

AdActionEvent:广告点击行为的回调,供自定义广告行为的响应使用，如果开发者关联事件并将 AdActionEventArgs.Cancel 设为 true 则广告本身预制的行为就不起作用

Age,Interest,City,Income 开发者可以根据拥有的用户的 Profile 信息，选择性的填写

6) 可以用来测试的参数

在 5)中需要的 Appid，Adid，SerectKey 还没有通过注册的方式获取到的时候，可以使用以下测试值测试广告的效果。

Inline 广告:

Appid: 100000
Adid: 100000
SerectKey: 1d89bd7887494e39ad5b0d606f1b7360

Interstitial 广告:

Appid: 100000
Adid: 100004
SerectKey: 1d89bd7887494e39ad5b0d606f1b7360