



## **WP8 DBOwnerSDK 使用文档**

V1.0.1

更多资源请访问 DBOwner 知识库

<http://wiki.downer.com>

<b>WP8 DBOwnerSDK 使用文档.....</b>	<b>1</b>
<b>一、 环境设置 .....</b>	<b>4</b>
1、 添加 DBOwnerLib 类库 .....	4
2、 设置 Capabilities .....	5
<b>二、 修改 App.xaml.cs .....</b>	<b>9</b>
1、 引入类库 DBOwnerLib.....	9
2、 在 App 类中声明类库对象 .....	9
3、 在 InitializePhoneApplication 方法中添加以下方法 .....	9
4、 在 Application_Launching、Application_Activated、Application_Deactivated、Application_Closing 四个方法中分别添加 DBOwnerLaunching、DBOwnerActivated、DBOwnerDeactivated、DBOwnerClosing 四个方法，如下面代码所示.....	10
5、 在方法 Application_UnhandledException 中添加 DBOwnerListenError 方法，如下面代码.....	11
<b>三、 实现 Auth 登陆验证.....</b>	<b>12</b>
1、 引入类库 DBOwnerLib.....	12
2、 在类中声明类库对象 .....	12
3、 在需要登录的地方添加以下代码.....	12
4、 DBOwner 中关于登录需要说明的方法 .....	13
<b>四、 实现 View 统计 .....</b>	<b>14</b>

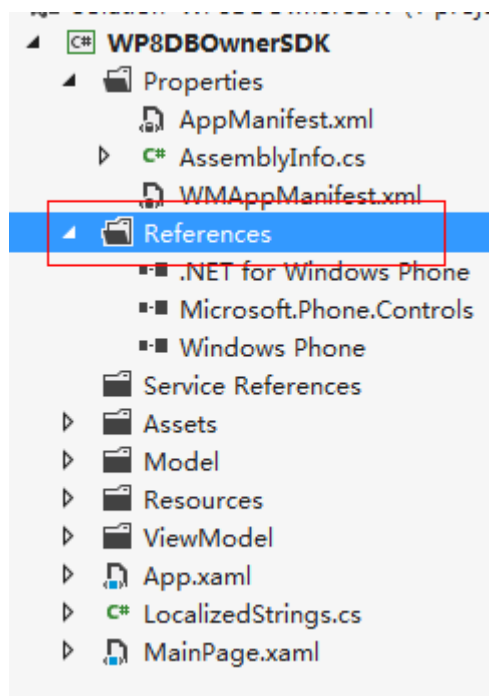
1、	引入 DBOwnerLib .....	14
2、	在类中声明类库对象 .....	14
3、	在类的构造方法中加入统计代码.....	14
五、	实现广告投放 .....	15
1、	引入 DBOwnerLib .....	15
2、	在类中声明类库对象 .....	15
3、	广告使用方法 .....	15

# 一、 环境设置

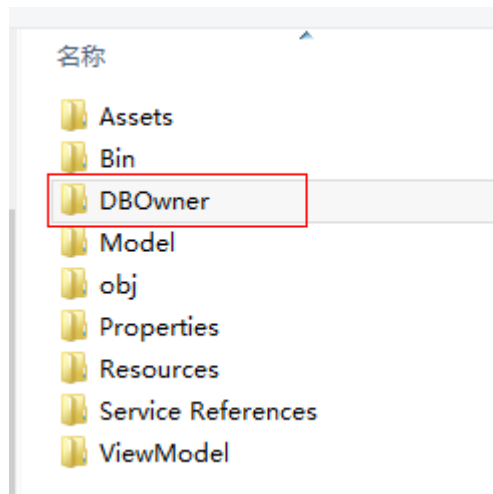
## 1、 添加 DBOwnerLib 类库

( 1 ) 首先下载 Demo 样例 , 然后新建项目工程文件

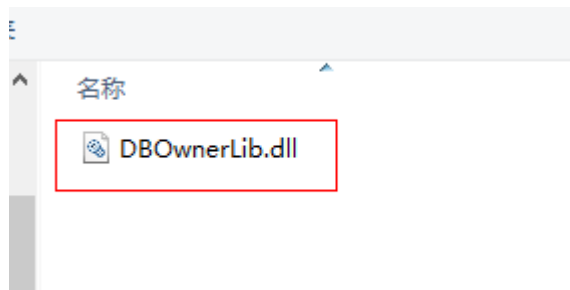
( 2 ) 将 DBOwnerLib 引入项目中 , 右击项目的 References->Add Reference...->Brower...



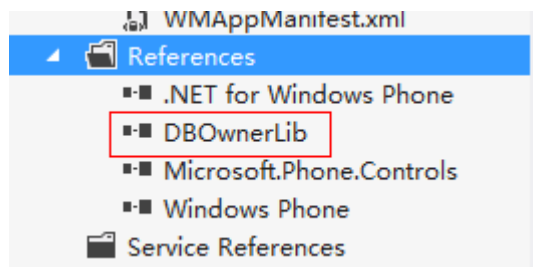
找到下载的文件 DBOwner



加载打开文件夹里面的 DBOwnerLib.dll



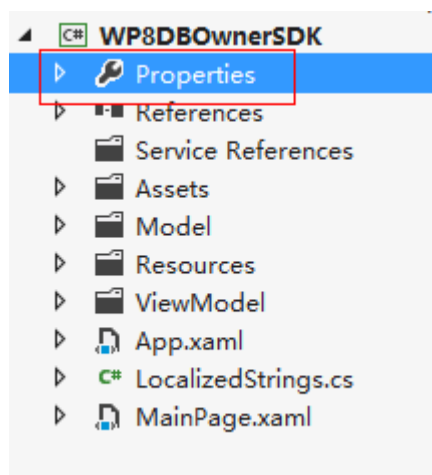
引入文件之后，如下图所示



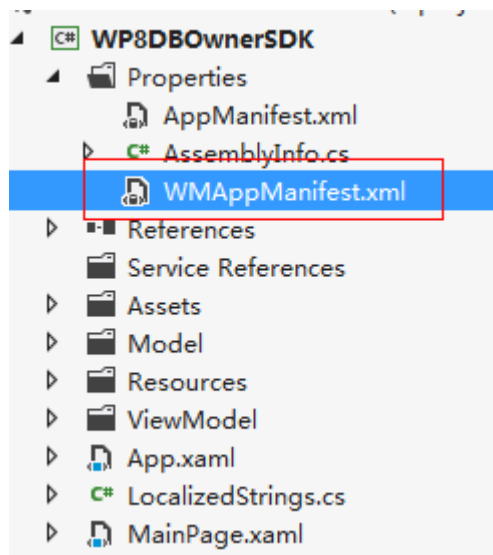
## 2、设置 Capabilities

添加项目所需要的能力，以便使用 DBOwnerSDK

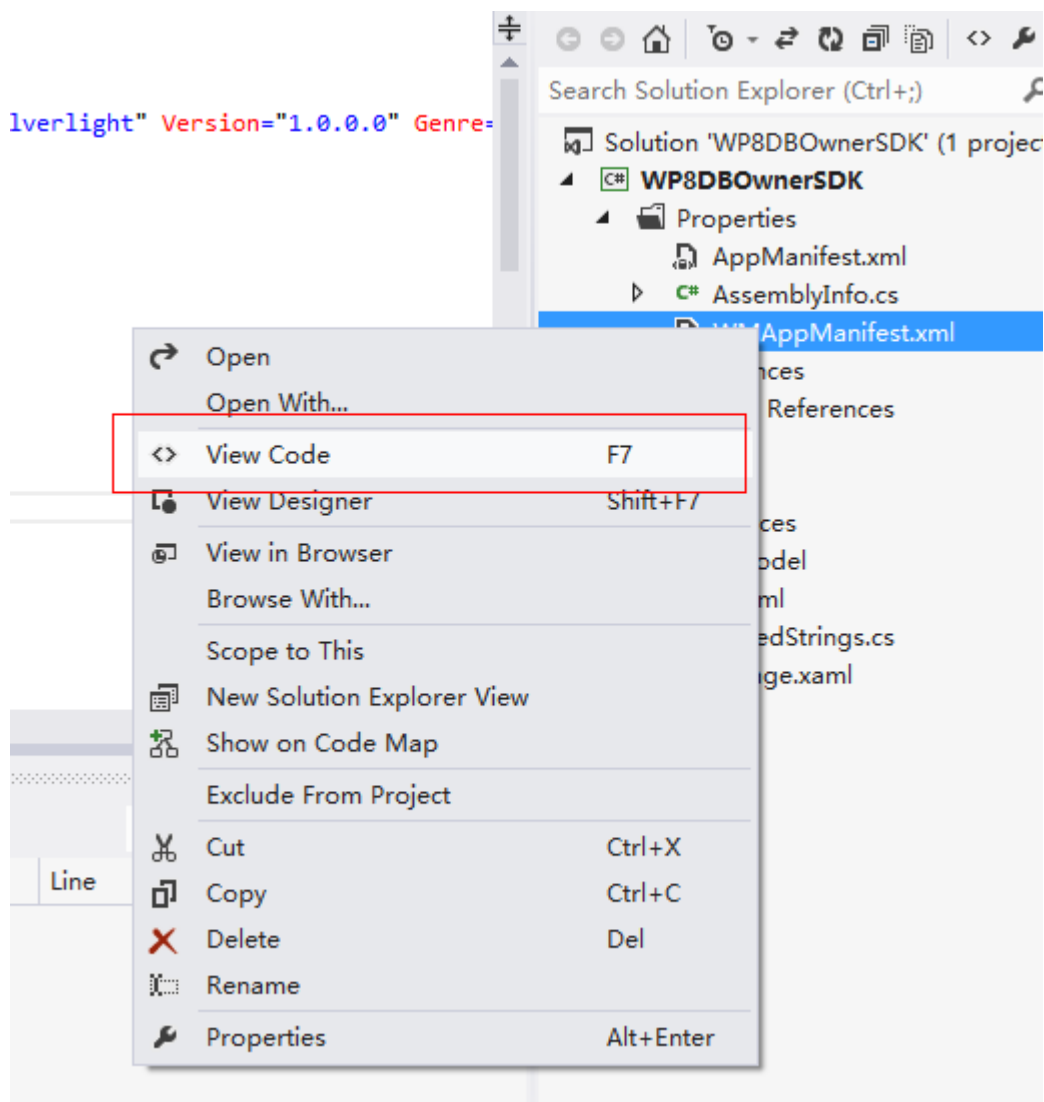
首先找到项目文件 Properites



找到 WAppManifest.xml



右击 WAppManifest.xml , 然后选择 View Code



打开文件，找到 Capabilities，添加

```
<Capability Name="ID_CAP_IDENTITY_DEVICE" />
<Capability Name="ID_CAP_IDENTITY_USER" />
<Capability Name="ID_CAP_LOCATION" />
<Capability Name="ID_CAP_PUSH_NOTIFICATION"/>
<Capability Name="ID_CAP_PHONEDIALER"/>
```

五个能力

```
App.xaml.cs  WMAAppManifest.xml  [icon] [x]
<?xml version="1.0" encoding="utf-8"?>
<Deployment xmlns="http://schemas.microsoft.com/windowsphone/2012/deploy"
  <DefaultLanguage xmlns="" code="en-US" />
  <App xmlns="" ProductID="{91151110-5cfd-417e-8034-381214475c4e}" Title="
    <IconPath IsRelative="true" IsResource="false">Assets\ApplicationIcon
  <Capabilities>
    <Capability Name="ID_CAP_NETWORKING" />
    <Capability Name="ID_CAP_MEDIALIB_AUDIO" />
    <Capability Name="ID_CAP_MEDIALIB_PLAYBACK" />
    <Capability Name="ID_CAP_SENSORS" />
    <Capability Name="ID_CAP_WEBBROWSERCOMPONENT" />
    <Capability Name="ID_CAP_IDENTITY_DEVICE" />
    <Capability Name="ID_CAP_IDENTITY_USER" />
    <Capability Name="ID_CAP_LOCATION" />
    <Capability Name="ID_CAP_PUSH_NOTIFICATION" />
    <Capability Name="ID_CAP_PHONEDIALER"/>
  </Capabilities>
  <Tasks>
    <DefaultTask Name="_default" NavigationPage="MainPage.xaml" />
  </Tasks>
  .
```



## 二、 修改 App.xaml.cs

### 1、 引入类库 DBOwnerLib

```
using DBOwnerLib;
```

### 2、 在 App 类中声明类库对象

```
//声明参数  
public static DBOwnerSDK dbOwnerSDK;
```

### 3、 在 InitializePhoneApplication 方法中添加以下方法

```
//引入DBOwnerSDK  
dbOwnerSDK = new DBOwnerSDK("client_id", "client_secret", "ADCode", "PublicKey", "PublicIV",  
"ChannelID");
```

参数说明：

client\_id: 应用 ID  
client\_secret: 应用密钥  
ADCode: 广告 code  
PublicKey: 预留值  
PublicIV: 预留值  
ChannelID: 来源平台

4、在 Application\_Launching、Application\_Activated、Application\_Deactivated、Application\_Closing 四个方法中分别添加 DBOwnerLaunching、DBOwnerActivated、DBOwnerDeactivated、DBOwnerClosing 四个方法，下面代码所示

```
// Code to execute when the application is launching (eg, from Start)
// This code will not execute when the application is reactivated
private void Application_Launching(object sender, LaunchingEventArgs e)
{
    dbOwnerSDK.DBOwnerLaunching();
}

// Code to execute when the application is activated (brought to foreground)
// This code will not execute when the application is first launched
private void Application_Activated(object sender, ActivatedEventArgs e)
{
    dbOwnerSDK.DBOwnerActivated();
}

// Code to execute when the application is deactivated (sent to background)
// This code will not execute when the application is closing
private void Application_Deactivated(object sender, DeactivatedEventArgs e)
{
    dbOwnerSDK.DBOwnerDeactivated();
}

// Code to execute when the application is closing (eg, user hit Back)
// This code will not execute when the application is deactivated
private void Application_Closing(object sender, ClosingEventArgs e)
{
    dbOwnerSDK.DBOwnerClosing();
}
```

## 5、在方法 Application\_UnhandledException 中添加 DBOwnerListenError 方法，如

下面代码

```
// Code to execute on Unhandled Exceptions
private void Application_UnhandledException(object sender,
ApplicationUnhandledExceptionEventArgs e)
{
    if (Debugger.IsAttached)
    {
        dbOwnerSDK.DBOwnerListenError(e);
        //e.Handled = true;

        // An unhandled exception has occurred; break into the debugger

        Debugger.Break();
    }
}
```

## 三、 实现 Auth 登陆验证

Auth 登陆支持新浪微博、腾讯 QQ、等多大 16 个合作网站的账号，使用标准 Auth2.0 协议。DBOwnerAuth 登陆时候需要传入用于登陆的控件显示的父 `UIViewController`，调用方法如下：

### 1、 引入类库 DBOwnerLib

```
using DBOwnerLib;
```

### 2、 在类中声明类库对象

```
//声明参数
DBOwnerSDK dbOwnerSDK = App.dbOwnerSDK;
```

### 3、 在需要登录的地方添加以下代码

```
dbOwnerSDK.DBOAuthLogin(this, DBOwner_LoginCompletedHandler);
```

其中 `DBOwner_LoginCompletedHandler` 是登录成功之后的回调函数，可以像下面代码处理

```
/// <summary>
/// 登录成功后处理
/// </summary>
public void DBOwner_LoginCompletedHandler(object sender, AuthLoginCompletedEventArgs e)
{
    if (e.Error == null)
    {
        NavigationService.Navigate(new Uri("/ViewModel/MainView.xaml",
UriKind.Relative));
    }
}
```

```
else
{
    MessageBox.Show(e.Error.Message);
}
}
```

#### 4、DBOwner 中关于登录需要说明的方法

IsAccessTokenValid: 是否已经授权过

GetAccessToken: 返回 access\_token 值

GetRefreshToken: 返回 refresh\_token 值

GetUserID: 返回 user\_id 值

其他获取用户相关方法，请参考 demo 中的例子

## 四、 实现 View 统计

### 1、 引入 DBOwnerLib

```
using DBOwnerLib;
```

### 2、 在类中声明类库对象

```
//声明参数  
DBOwnerSDK dbOwnerSDK = App.dbOwnerSDK;
```

### 3、 在类的构造方法中加入统计代码

```
dbOwnerSDK.DBOwnerPushViewPage("ViewCode");
```

其中 ViewCode 即代表当前页面 code 值

## 五、 实现广告投放

DBOwnerAD 支持全屏、插屏、横条 ( Top、Bottom、Left、Right ) 多种广告类型 , 并支持自动转屏重置。

### 1、 引入 DBOwnerLib

```
using DBOwnerLib;
```

### 2、 在类中声明类库对象

```
//声明参数  
DBOwnerSDK dbOwnerSDK = App.dbOwnerSDK;
```

### 3、 广告使用方法

```
全屏广告: dbOwnerSDK.addAdFullScreen(this);  
插屏广告: dbOwnerSDK.addAdInsertScreen(this);  
顶部横幅广告: dbOwnerSDK.addAdBanner(this, "up");  
底部横幅广告: dbOwnerSDK.addAdBanner(this, "down");
```