

亿动智道应用内置广告开发包使用说明

SmartMad In-App Ad SDK for Windows Phone

**V3.0.1
Madhouse Inc.**

目录

目录	1
1. 适用范围	3
2. 名词解释	3
3. 广告类型	4
3.1. 固定广告位	4
3.1.1. 旗帜广告	4
3.1.2. 全屏广告	5
3.1.3. 炫动广告	5
3.1.4. 视频广告	6
3.2. 浮动广告位	6
3.2.1. 插页广告	6
3.2.2. 富媒体插页广告	7
4. 广告植入说明	7
4.1. 准备	7
4.2. 建立工程项目	8
4.2.1. 建立新的Windows Phone 应用程序	8
4.2.2. 添加SmartMad SDK 库文件	9
4.2.3. 建立库文件与项目的引用关系	9
4.3. 添加必要权限	10
4.3.1. 目标Windows Phone OS 版本7.1 (选择)	10
4.3.2. 目标Windows Phone OS 版本8.0 (选择)	10
4.3.3. 权限说明	11
4.4. 设置全局性配置信息	11
4.4.1. 通过应用资源设置信息 (选择)	11
4.4.2. 通过编写代码设置信息 (选择)	12
4.4.3. 广告请求间隔时间设置规则	12
4.5. 添加固定位旗帜广告	13
4.5.1. Design 方式：通过Toolbox 控件添加广告 (选择)	13
4.5.2. XAML 方式：通过修改XAML 配置添加广告 (选择)	14
4.5.3. Coding 方式：通过C# 代码添加广告 (选择)	15
4.5.4. 固定位旗帜广告事件说明 (选择)	16

4.6.	添加浮动位插页广告	16
4.6.1.	Coding 方式：通过C# 代码添加广告	16
4.6.2.	浮动位插页广告事件说明（选择）	17
4.7.	广告必要参数规范（重要）	17
4.7.1.	广告位标识：AdSpaceId	17
4.7.2.	广告尺寸类型：AdSize	18
4.7.3.	广告动画类型：AdAnimationType	18
5.	广告植入高级技巧	19
5.1.	广告预缓存机制应用	19
5.1.1.	Coding 方式：通过C# 代码调用广告预缓存机制	19
5.1.2.	广告预缓存必要参数说明	20
5.1.3.	广告预缓存事件说明	20
5.2.	提供信息辅助广告精确定向	20
5.2.1.	提供终端用户信息	20
5.2.2.	提供关键字信息	21
5.3.	混淆注意事项（重要）	21
5.4.	广告事件(SMAdEventCode)说明	22

1. 适用范围

本文档主要介绍“亿动智道”移动广告平台(SmartMad)为开发者和第三方合作伙伴提供的基于 Windows Phone 移动手机操作系统（7.5/7.8 及 8.0 版本）的移动应用程序中广告植入的方法和规范，并简单的描述了投放广告的类型和展示形式，有助于开发人员快速的植入广告到移动应用程序中。本文档主要面向具有一定 Windows Phone 开发经验的人员编写，关于 Windows Phone 编程技术的介绍不属于本文档的介绍范围。

2. 名词解释

名词	全称	释义
WP	Windows Phone	微软移动设备操作系统。
Silverlight	Microsoft Silverlight	微软基于 Web 前端应用程序的跨浏览器和操作系统的开发框架，是微软 Rich Internet Application 策略的主要构成之一。
XNA	Windows Xbox Next-generation Architecture	XNA 是基于 DirectX 的游戏开发环境（开发框架+开发工具），是微软对于 Managed DirectX 的修正及扩充版本。
XAML	eXtensible Application Markup Language	“可扩展应用程序标记语言”，它是微软为构建应用程序用户界面而创建的一种描述性语言，是一种解析性的语言。
VS2010(VS2012类同)	Microsoft Visual Studio 2010	微软提供的支持 WP 开发的 IDE 工具，在 WP 开发包中提供的是 Express 版本。
SmartMad	SmartMad Advertising Platform	“亿动智道”移动广告平台。
Ad SDK	Advertising System Development kit	SmartMad提供的用于移动平台应用程序的广告系统植入开发包。
AppId	Application Identity	SmartMad中注册的用于标识移动平台应用程序的标识码。
AdSpaceId	Advertising Position Identity	SmartMad中根据AppId分配的用于标识应用程序中广告位置的标识码。
App Package Name	Application Package Name	SmartMad中和AppId、AdSpaceId进行绑定，用于校验独立移动应用程序的唯一标识。
Refresh Interval	Advertising Request Interval	广告请求间隔，开发者可以在SDK接口中设置或在SmartMad广告平台网站中设置。

名词	全称	释义
Ad Size	Advertising Size	广告规格尺寸，SmartMad中定义的广告位的尺寸规则。
Banner Ad Animation	Banner Advertising Animation Type	Banner广告过场动画，SmartMad中定义的广告切换之间执行的动画类型。
Interstitial Ad Animation	Interstitial Advertising Animation Type	Interstitial广告过场动画。SmartMad中定义的广告切换之间执行的动画类型。
Debug	Advertising Debug Model	Debug模式，也称作广告调试模式。开发者在Debug模式下可以获得所有类型的测试广告，并且在IDE的控制台中能获取有益的状态或错误提示。注意：测试模式下无法计费。
App	Application	Mobile Application，即移动应用程序，在下文中泛指所有基于 Windows Phone 建立的“移动应用程序”。

3. 广告类型

目前 SmartMad 为 WP 移动应用程序提供了 2 种布局接入方式 (In-Style 和 Float Window)，共计 6 种广告展示形式，支持 MRAID 1.0 和 ORMMA Level1-2 开放式富媒体接口，为富媒体广告创意和富媒体广告交换提供了强有力的支撑。

3.1.固定广告位

固定广告位基于应用程序布局中，位置相对固定，也被称为样式内广告位 (In-Style) 或内联式广告位 (Inline)，具体包含以下广告形式：

3.1.1. 旗帜广告

旗帜广告，即 Banner 广告。分为图形 Banner 广告、文字链 Banner 广告和富媒体 Banner 广告，可以将该广告视图放于任何可以容纳的位置（容纳的容器或位置必须在屏幕内并且尺寸等于或大于广告视图，推荐放在界面的顶部或底部）。针对每个 App，系统都提供 2 个广告位，且在同一屏幕仅可同时使用 2 个不同的广告位。



3.1.2. 全屏广告

全屏广告，即 Full Screen 广告。是一种基于 Banner 广告形式，使用 MRAID/ORMMA 中 Expand 方式扩展到整个屏幕进行广告展示的广告类型。全屏广告当前会在首次调用时展示，使用在 App 启动时调用可类似于开屏广告（Splash）进行展示。



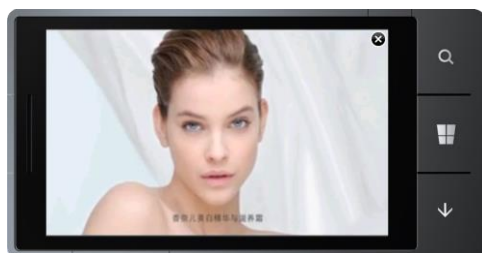
3.1.3. 炫动广告

炫动广告，也被称为 Crazy Ad。是一种基于 Banner 广告，使用 MRAID/ORMMA 中 Expand 方式扩展到整个屏幕并动态展示广告的富媒体广告形式（Rich Media Ad）。



3.1.4. 视频广告

视频广告，是一种基于 Banner 广告，使用 MRAID/ORMMA 中 Expand 方式扩展到半屏或全屏，使用流媒体视频在移动终端展示的富媒体广告形式。



3.2. 浮动广告位

浮动位广告位以窗口方式浮动于应用程序布局之上，因为无需对应用布局进行调整，也被称为浮动窗广告位（float window），具体包含以下广告形式：

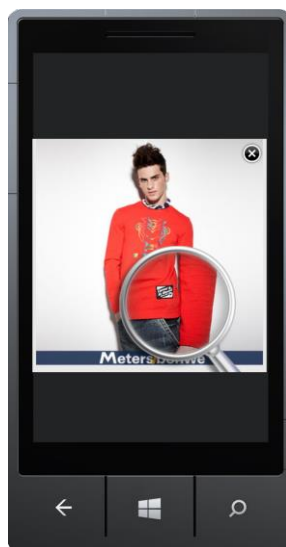
3.2.1. 插页广告

插页广告，即 Interstitial Ad。是一种开发者可以在界面过场时请求并根据需要进行展示的间质性广告形式。插页广告有区别于固定位旗帜广告的 API，调用形式灵活可控。



3.2.2. 富媒体插页广告

富媒体插页广告，是一种插页广告的富媒体版本。其展示内容更据视觉效果，且含有丰富的交互行为，有利于广告的转化和效果。







4. 广告植入说明

应用程序进行广告植入将添加 SmartMad Ad SDK 所包含的代码和文件，以下开发示例说明将假设在以 Windows 7 和 Windows Phone SDK 7.1(包含 Visual Studio 2010 Express、Windows Phone Emulator、Silverlight 和 XNA Game Studio 4.0)或者 Windows 8 和 Windows Phone SDK 8.0 搭建的开发环境下，使用 Silverlight (C#) 开发框架实现。

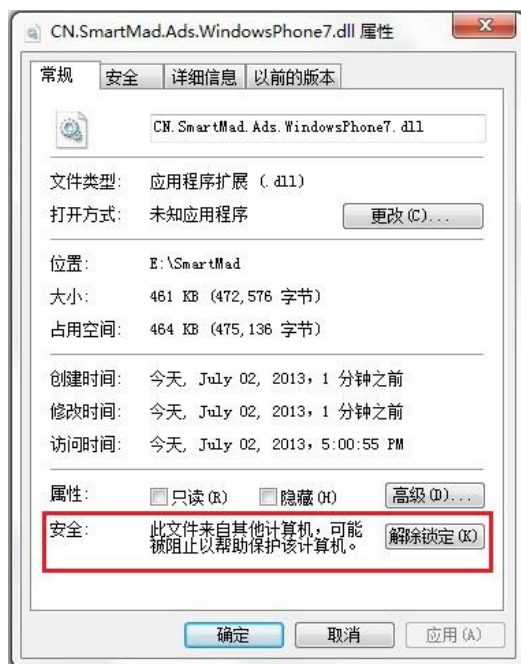
注意:SmartMad Ad SDK 3.0 根据“目标 Windows Phone OS 版本”的不同,分为支持 Windows Phone OS 7.1 及以上版本应用程序的 Universal 版本和仅支持 Windows Phone OS 8.0 应用程序的 Only WP8 版本 (更好的支持 MRAID)。两个 SDK 库文件使用相同的名称空间,不同的文件名称(Universal 的文件名为: CN.SmartMad.Ads.WindowsPhone7.dll ; Only WP8 的文件名为 CN.SmartMad.Ads.WindowsPhone8.dll), 在一个工程中仅能导入其中一个 SDK 文件。

4.1.准备

在亿动智道平台网站下载 SmartMad In-App Ad SDK for WP 发布版本压缩包，解压缩后将会在 SmartMad 目录中看到以下文件：

名称	类型
 CN.SmartMad.Ads.WindowsPhone7.dll	应用程序扩展
 CN.SmartMad.Ads.WindowsPhone8.dll	应用程序扩展
 Newtonsoft.Json.dll	应用程序扩展
 SharpZipLib.WindowsPhone7.dll	应用程序扩展

目录中除了 SmartMad In-App Ad SDK 的两版本的主库文件外，还包括主库正常工作时所必须的“Newtonsoft.Json.dll”和“SharpZipLib.WindowsPhone7.dll”两个第三方库文件。需要特别注意的是：使用这些库文件前，需要检查这四个文件的属性中安全状态是否锁定，如果处于锁定状态请“解除锁定”后再使用，具体如下图：



4.2. 建立工程项目

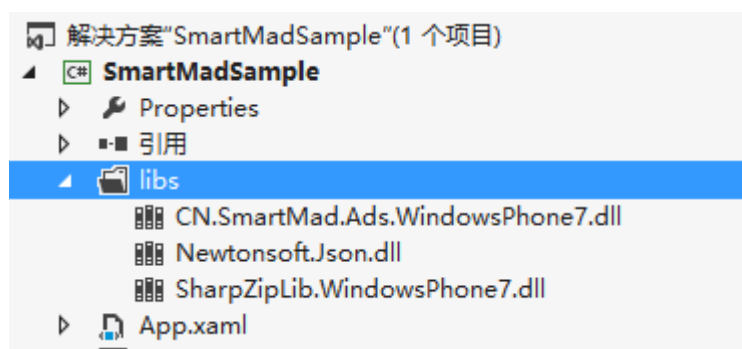
4.2.1. 建立新的 Windows Phone 应用程序

使用 Microsoft Visual Studio(Express)建立新的项目并选择模板，在弹出的对话框中选择应用程序需要的目标 Windows Phone OS 版本，具体如下图：



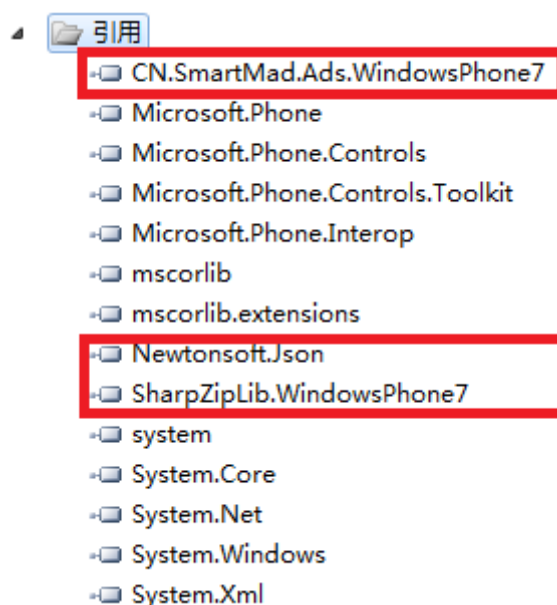
4.2.2. 添加 SmartMad SDK 库文件

在“解决方案资源管理器”窗口中工程项目名称下建立“libs”目录，通过右键菜单中“添加”项中的“现有项...”子项，在弹出的窗口中选中解压缩后的 SmartMad 库文件（根据 4.2.1 步骤中所选目标 Windows Phone OS 版本选择主库文件）和必须的第三方库文件，确认添加到此目录中。结果具体如下图：



4.2.3. 建立库文件与项目的引用关系

在“解决方案资源管理器”窗口中工程项目名称下的“引用”（References），通过右键菜单中“添加引用...”项，在弹出窗口中选中已添加到“libs”目录的库文件，依次确认将主库文件和第三方库文件和工程项目建立引用。结果具体如下图：



4.3. 添加必要权限

SmartMad SDK 中功能的正常运行，需要工程项目中配置相应的能力权限，具体方法如下：

4.3.1. 目标 Windows Phone OS 版本 7.1（选择）

在工程项目的 Properties 文件夹下找到 “WMAppManifest.xml” 文件，双击打开并在 Capabilities 项中检查是否有如下必要权限：

```
<Capabilities>
  <!--必选权限 -->
  <Capability Name="ID_CAP_NETWORKING"/>
  <Capability Name="ID_CAP_IDENTITY_DEVICE"/>
  <Capability Name="ID_CAP_IDENTITY_USER"/>
  <Capability Name="ID_CAP_WEBBROWSERCOMPONENT"/>
  <Capability Name="ID_CAP_MEDIALIB"/>
  <!--可选权限 -->
  <Capability Name="ID_CAP_LOCATION"/>
  <Capability Name="ID_CAP_PHONEDIALER"/>
  <Capability Name="ID_CAP_SENSORS"/>
</Capabilities>
```

注意：在基于目标 Windows Phone OS 版本 7.1 下，所有使用到的权限都应该已经添加到工程项目中，如果有缺少的必选权限，需要手动添加补全。

4.3.2. 目标 Windows Phone OS 版本 8.0（选择）

在工程项目的 Properties 文件夹下找到 “WMAppManifest.xml” 文件，双击打开并在 Capabilities 项中检查是否有如下必要权限：

```
<Capabilities>
  <!--必选权限 -->
  <Capability Name="ID_CAP_NETWORKING"/>
  <Capability Name="ID_CAP_IDENTITY_DEVICE"/>
  <Capability Name="ID_CAP_IDENTITY_USER"/>
  <Capability Name="ID_CAP_WEBBROWSERCOMPONENT"/>
  <!--可选权限 -->
  <Capability Name="ID_CAP_LOCATION"/>
  <Capability Name="ID_CAP_APPOINTMENTS"/>
  <Capability Name="ID_CAP_PHONEDIALER"/>
  <Capability Name="ID_CAP_SENSORS"/>
</Capabilities>
```

注意：在基于目标 Windows Phone OS 版本 8.0 下，新建工程项目中仅有基本权限，如果有缺少的必选权限，需要手动添加补全。

4.3.3. 权限说明

关于 SmartMad SDK 使用到的权限和用途，见下表：

权限名称	用途说明
ID_CAP_NETWORKING	当应用从 Visual Studio 部署到 Windows Phone 或 Windows Phone 模拟器时，将自动包含网络功能。如果您的应用需要连接网络，那么在向 商店 提交应用时，您必须在应用清单文件中指定该功能。如果您没有指定网络连接功能，那么应用将在用户手机上进行安装时失败。
ID_CAP_IDENTITY_DEVICE	提供对设备特定信息(如，唯一设备 ID、制造商或模型名称) 的访问。
ID_CAP_IDENTITY_USER	给予应用使用匿名 Microsoft 账户以识别用户的功能。
ID_CAP_WEBBROWSERCOMPONENT	提供对 Web 浏览器组件的访问。
ID_CAP_MEDIALIB	提供对媒体库的访问。
ID_CAP_LOCATION	提供对位置服务的访问。
ID_CAP_SENSORS	提供对任意 Windows Phone 传感器的访问。
ID_CAP_PHONEDIALER	提供使用PhoneCallTask API 的能力。
ID_CAP_APPOINTMENTS	提供对约会数据的访问。

提示：关于 Windows Phone 应用的功能和硬件要求的更多说明内容，可以从微软官方网站中获取：

<http://technet.microsoft.com/zh-cn/subscriptions/jj206936.aspx>

4.4. 设置全局性配置信息

添加广告容器之前，需要设置一些全局性配置信息以便正常进行广告投放。具体配置方法如下：

4.4.1. 通过应用资源设置信息（选择）

在“解决方案资源管理器”窗口中工程项目名称下，找到并双击“App.xaml”文件，在文件中的“Application.Resources”标签内添加以下内容（粗体字部分）：

```
<Application
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:sys="clr-namespace:System;assembly=microsoft.windows.common-base-1.0"
  <Application.Resources>
    <sys:String x:Key="SM_APP_ID">Application ID</sys:String>
    <sys:Double x:Key="SM_AD_REFRESH_INTERVAL">Interval value</sys:Double>
  </Application.Resources>
</Application>
```

“SM_APP_ID” 标签中的“Application ID”需要替换为在亿动智道平台中所申请媒体的“应用程序/网站 ID”；“SM_AD_REFRESH_INTERVAL”标签中的数字则需要设置广告请求(广告刷新)间隔时间，具体见“广告请求间隔时间设置规则”所述内容。

4.4.2. 通过编写代码设置信息（选择）

除了可以在应用程序资源配置文件中添加广告全局配置信息外，还可以在编写代码的时候进行全局配置信息的设置。找到并打开“App.xaml.cs”文件，在“Application_Launching”方法中添加如下代码（粗体字部分）：

```
// 应用程序启动(例如，从“开始”菜单启动)时执行的代码，此代码在重新激活应用程序时不执行
private void Application_Launching(object sender, LaunchingEventArgs e)
{
    // 设置亿动智道平台中获取的应用程序/网站ID
    SManager.SetApplicationId("Application ID");

    // 设置广告请求(广告刷新)的间隔时间
    SManager.SetAdRefreshInterval(int);

    // 设置开发(Debug)模式
    SManager.SetDebugMode(boolean);
}
```

注意：设置应用程序 ID、广告请求间隔时间和开发者模式需要在实例化广告视图之前。开发模式仅可以通过编写代码方式进行设置，默认值为“false”表示正式计费投放；值为“true”表示开发模式，将会展示 Debug 广告并在 IDE 的控制台中显示提示信息或调试信息，但是开发模式不会进行计费。

4.4.3. 广告请求间隔时间设置规则

广告请求间隔时间即广告刷新时间，该值直接影响广告的展示频率。SmartMad 对该值的定义如下：

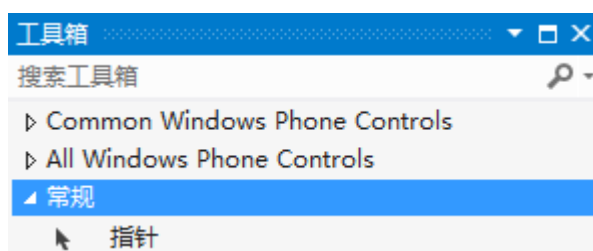
类型	数值（单位:秒）	说明
特殊值	0	默认值，以系统的默认时间为准
	1	仅请求 1 次广告，适合于聚合平台使用
区间值	20~600	最小 20，最大 600，如果取值超过此区间则自动使用默认值 0

提示：广告刷新时间根据经验并不是越小越好，展示时间过短会影响广告浏览者对广告内容产生兴趣的概率，一般取值在 45-60 秒之间时广告被点击的概率会上升到最大。

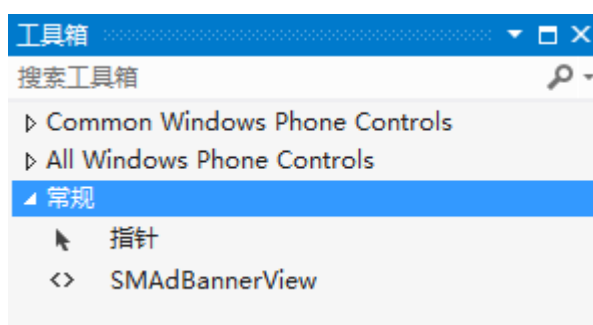
4.5.添加固定位旗帜广告

4.5.1. Design 方式：通过 Toolbox 控件添加广告（选择）

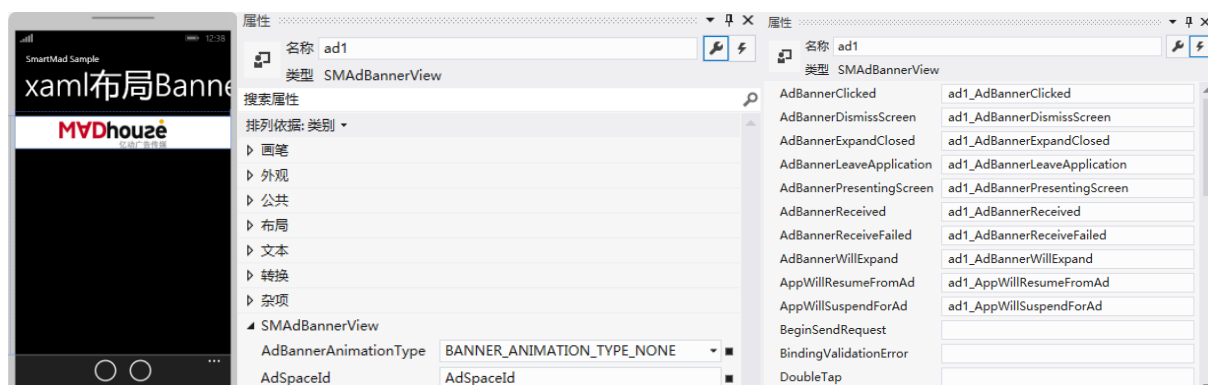
在“工具箱”的“常规”分支项点击右键并激活菜单中的“选择项”，如下图：



在弹出的“选择工具箱项”对话框中，选中“Windows Phone 组件”标签，然后点击“浏览”按钮找到引入的 SmartMad SDK 库文件并“确定”加入，具体如下图：



在 Design 视图中，将“SMAdBannerView”控件从 Toolbox 中拖入相应的位置后，通过“设置”窗口中设置相关属性值和事件，具体如下图：



注意：“AdSpaceId”中填入亿动智道平台中建立媒体时所分配的“广告位”，关于广告动画和广告事件的描述将在之后章节介绍。

4.5.2. XAML 方式：通过修改 XAML 配置添加广告（选择）

打开需要添加广告的 XAML 文件，将广告代码添加入布局适当位置，具体如下（粗体字部分）：

```
<!--在此添加名称空间信息-->
<phone:PhoneApplicationPage
    x:Class="Test.AdTestPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="PortraitOrLandscape" Orientation="Portrait"
    mc:Ignorable="d" d:DesignHeight="696" d:DesignWidth="480"
    shell:SystemTray.IsVisible="True"
    xmlns:SmartMad="clr-namespace:CN.SmartMad.Ads.WindowsPhone.WPF;assembly=CN.SmartMad.Ads.WindowsPhone7">

<!--广告控件内容-->
<Grid x:Name="ContentPanel" Grid.Row="1" >
    <SmartMad:SMAdBannerView Name="adbanner" VerticalAlignment="Top"
        AdSpaceId="AdSpaceId"
        AdBannerAnimationType="BANNER_ANIMATION_TYPE_NONE"
        AdBannerClicked="ad1_AdBannerClicked"
        AdBannerDismissScreen="ad1_AdBannerDismissScreen"
        AdBannerExpandClosed="ad1_AdBannerExpandClosed"
        AdBannerLeaveApplication="ad1_AdBannerLeaveApplication"
        AdBannerPresentingScreen="ad1_AdBannerPresentingScreen"
        AdBannerReceived="ad1_AdBannerReceived"
        AdBannerReceiveFailed="ad1_AdBannerReceiveFailed"
        AdBannerWillExpand="ad1_AdBannerWillExpand"
        AppWillResumeFromAd="ad1_AppWillResumeFromAd"
        AppWillSuspendForAd="ad1_AppWillSuspendForAd" />
</Grid>
</Grid>
```

注意：设置名称空间时，“assembly”内容和所选“目标 Windows Phone OS 版本”有关（主库文件名称）；而“AdSpaceId”内容请使用亿动智道平台建立媒体时所分配的“广告位”，关于广告动画和广告事件的描述将在之后章节介绍。

4.5.3. Coding 方式：通过 C#代码添加广告（选择）

打开需要添加广告的“cs”文件，将广告代码添加适当位置，具体如下（粗体字部分）：

```
namespace SmartMadSample {  
    public partial class BannerAd : PhoneApplicationPage {  
        public BannerAd() {  
            InitializeComponent();  
            // 实例化SmartMad Banner容器  
            SMAdBannerView banner = new SMAdBannerView("AdSpaceId");  
            // 设置Banner广告刷新过场动画类型(可选)  
            banner.AdBannerAnimationType =  
                SMAdBannerAnimationType.BANNER_ANIMATION_TYPE_FADEINOUT;  
            // 设置Banner广告事件  
            banner.AdBannerClicked += new EventHandler(banner_AdBannerClicked);  
            banner.AdBannerWillExpand += new EventHandler(banner_AdBannerWillExpand);  
            banner.AdBannerExpandClosed += new  
                EventHandler(banner_AdBannerExpandClosed);  
            banner.AdBannerReceived += new EventHandler(banner_AdBannerReceived);  
            banner.AdBannerReceiveFailed += new  
                EventHandler<SMAdEventCode>( banner_AdBannerReceiveFailed);  
            banner.AdBannerPresentingScreen += new  
                EventHandler<SMAdEventCode>( banner_AdBannerPresentingScreen);  
            banner.AdBannerLeaveApplication += new  
                EventHandler(banner_AdBannerLeaveApplication);  
            banner.AppWillSuspendForAd += new  
                EventHandler(banner_AppWillSuspendForAd);  
            banner.AppWillResumeFromAd += new  
                EventHandler(banner_AppWillResumeFromAd);  
            banner.AdBannerDismissScreen += new  
                EventHandler(banner_AdBannerDismissScreen);  
            // 将Banner容器加入界面  
            ContentPanel.Children.Add(banner);  
        }  
    }  
}
```

注意：“AdSpaceId”内容请使用亿动智道平台建立媒体时所分配的“广告位”，关于广告动画和广告事件的说明将在之后章节介绍。

4.5.4. 固定位旗帜广告事件说明（选择）

添加固定位旗帜广告时，可以通过一些事件接收广告当前状态，关于事件定义具体说明如下表：

事件	说明
AdBannerReceived	旗帜广告接收成功
AdBannerReceiveFailed	旗帜广告接收失败
AdBannerPresentingScreen	旗帜广告将要添加到当前屏幕上
AdBannerDismissScreen	旗帜广告将要当前屏幕上移除
AdBannerLeaveApplication	旗帜广告被点击后 app 将会被切换到后台,例如：调用系统浏览器
AppWillSuspendForAd	旗帜广告被点击后 app 被遮挡，例如：调用内置浏览器
AppWillResumeFromAd	旗帜广告行为遮挡 app 的状态解除，例如：关闭了内置浏览器
AdBannerClicked	旗帜广告被点击
AdBannerWillExpand	旗帜广告将要 expand
AdBannerExpandClosed	旗帜广告 expand 状态恢复

建议：在应用程序被广告展示遮挡的事件中，暂停当前应用的逻辑执行，直到终端用户关闭了广告窗口并解除了遮挡状态，再恢复当前应用的逻辑执行。

4.6. 添加浮动位插页广告

4.6.1. Coding 方式：通过 C# 代码添加广告

浮动位插页广告仅能通过 C# 代码的形式添加，具体操作分为以下两个步骤（粗体字部分）：

```
namespace SmartMadSample {  
    public partial class InterstitialAd : PhoneApplicationPage {  
        /* 第一步： 实例化插页广告并建立事件关联，然后请求广告 */  
        // 声明插页对象句柄  
        SMAdInterstitial interstitial;  
  
        public InterstitialAd() {  
            InitializeComponent();  
            // 实例化 SmartMad Interstitial 广告  
            interstitial = new SMAdInterstitial("AdSpaceId");  
            // 设置插页广告显示关闭动画类型  
            interstitial.AdInterstitialAnimationType =  
            SMAdInterstitialAnimationType.INTERSTITIAL_ANIMATION_TYPE_FADEINOUT;  
            // 设置插页广告接收事件  
            interstitial.AdInterstitialReceived += new EventHandler(ad_AdInterstitialReceived);  
            // 开始请求插页广告  
  
            interstitial.RequestAd();  
        }  
    }  
}
```

```
/* 第二步： 插页广告接收事件成功后，调用展示插页广告 */  
private void ad_AdInterstitialReceived(object sender, EventArgs e) {  
    // 展示插页广告  
    interstitial.Show();  
}  
}
```

注意：插页广告的展示需要配合插页广告请求事件，未请求成功时调用插页广告展示将无法响应。

4.6.2. 浮动位插页广告事件说明（选择）

添加浮动位插页广告时，可以通过一些事件接收广告当前状态，关于事件定义具体说明如下表：

事件	说明
AdInterstitialReceived	插页广告接收成功
AdInterstitialReceiveFailed	插页广告接收失败
AdInterstitialPresentingScreen	插页广告将要添加到当前屏幕上
AdInterstitialDismissScreen	插页广告将从当前屏幕上移除
AdInterstitialLeaveApplication	插页广告被点击后 app 将会被切换到后台 例如：调用系统浏览器
AdInterstitialClicked	插页广告被点击

建议：插页广告是浮动位广告形式，当添加到屏幕上后，会遮挡当前运行的应用程序，开发者应暂停当前应用的逻辑执行，直到终端用户关闭了广告窗口，插页广告被从屏幕移除之后，再恢复应用的逻辑执行。

4.7. 广告必要参数规范（重要）

4.7.1. 广告位标识：AdSpaceId

- 广告位标识是开发者在亿动智道广告平台中建立媒体时分配的一个 8 位字符串型数字。广告位标识与“应用程序/网站 ID”和“包名称”（Marketplace Release Product Id）互为绑定关系，故一款应用程序仅能使用申请时对应的广告位，否则广告请求将无效。
- 广告位使用时还要遵循以下原则：相同的广告位在同一屏幕（页面）中仅允许使用一次；不相同的广告位在同一屏幕中使用不能超过两个。

4.7.2. 广告尺寸类型：AdSize

- 因为 WP 不存在平板设备，所以固定位旗帜广告的广告尺寸类型统一使用自动类型进行处理。
- 浮动位插页广告的广告尺寸类型分类如下表：

枚举类型(SMAdInterstitialSizeType)	说明
AD_INTERSTITIAL_MEASURE_AUTO	插页广告尺寸类型由广告投放系统根据终端设备屏幕尺寸自动决定，此选项为默认设定
AD_INTERSTITIAL_MEASURE_UMAP	插页广告尺寸类型使用 “MMA Universal Mobile Ad Package 2.0 - UMAP” 标准，仅使用 300x250 的插页尺寸。

4.7.3. 广告动画类型：AdAnimationType

广告切换时，为了能更好的吸引终端用户的眼球，SmartMad SDK 提供了一系列的动画类型供开发者选用，具体动画类型说明如下：

- 固定位旗帜广告过场动画类型

枚举类型(SMAdBannerAnimationType)	说明
BANNER_ANIMATION_TYPE_NONE	旗帜广告切换无过场动画效果
BANNER_ANIMATION_TYPE_RANDOM	旗帜广告切换随机选择过场动画效果
BANNER_ANIMATION_TYPE_FADEINOUT	旗帜广告切换使用淡入淡出动画效果
BANNER_ANIMATION_TYPE_FLIPFROMLEFT	旗帜广告切换使用从左向右翻转动画效果
BANNER_ANIMATION_TYPE_FLIPFROMRIGHT	旗帜广告切换使用从右向左翻转动画效果
BANNER_ANIMATION_TYPE_CURLUP	旗帜广告切换使用向上卷起动画效果
BANNER_ANIMATION_TYPE_CURLDOWN	旗帜广告切换使用向下卷起动画效果
BANNER_ANIMATION_TYPE_SLIDEFROMLEFT	旗帜广告切换使用从左向右滑动动画效果
BANNER_ANIMATION_TYPE_SLIDEFROMRIGHT	旗帜广告切换使用从右向左滑动动画效果

- 浮动位插页广告显示关闭动画类型

枚举类型(SMAdInterstitialAnimationType)	说明
INTERSTITIAL_ANIMATION_TYPE_NONE	插页广告显示关闭无动画效果
INTERSTITIAL_ANIMATION_TYPE_POPUP	插页广告显示关闭使用弹出动画效果
INTERSTITIAL_ANIMATION_TYPE_FADEINOUT	插页广告显示关闭使用淡入淡出动画效果

5. 广告植入高级技巧

5.1. 广告预缓存机制应用

广告预缓存机制是 SmartMad SDK 为开发者提供的一种在应用程序启动时根据自身需要，调用获取广告缓存物料加速正式广告投放的广告通用型技术。

5.1.1. Coding 方式：通过 C# 代码调用广告预缓存机制

使用广告预缓存功能必须在实例化广告容器之前完成，具体代码如下（粗体字部分）：

```
namespace SmartMadSample {  
    public partial class App : Application {  
        public PhoneApplicationFrame RootFrame { get; private set; }  
        public App() {  
            // Constructor for the Application object.  
        }  
        private void Application_Launching(object sender, LaunchingEventArgs e){  
            SManager.SetApplicationId(settings.AppId);  
            // 请求广告预缓存  
            SManager.EnableAdPrecache("AdSpaceId",  
SAdUnitType.BANNER_AD_UNIT_TYPE);  
            // 建立事件监听  
            SManager.AdPrecacheStarted += new  
                SAdPrecacheEventHandler(AdPrecache_Started);  
            SManager.AdPrecacheError += new  
                SAdPrecacheEventHandler<SAdEventCode>(AdPrecache_Error);  
            SManager.AdPrecacheCompleted += new  
                SAdPrecacheEventHandler(AdPrecache_Completed);  
        }  
        private void AdPrecache_Started() { // TODO: }  
        private void AdPrecache_Error(SAdEventCode code) { // TODO: }  
        private void AdPrecache_Completed() { // TODO: }  
    }  
}
```

注意：广告预缓存功能当前仅支持应用程序启动后在请求正式广告前执行一次，且请求必须指定需要预缓存的广告位和广告类型。广告预缓存功能仅对广告物料尺寸较大的广告生效（例如：富媒体广告），且处于节省终端用户流量的考虑，本功能仅在 WIFI 请求时才能正常激活。

5.1.2. 广告预缓存必要参数说明

- 广告位(SMAdSpaceId)
亿动智道平台中申请应用程序媒体时分配的广告位。预缓存使用的广告位必须和正式广告请求的广告位相同，这样预缓存加速才会有效果。
- 广告位类型(SMAdUnitType)
亿动智道平台中的广告位当前是包含固定位旗帜广告和浮动位插页广告两种类型，且开发者可以根据自己的情况进行选择，所以调用广告预缓存时开发者也必须根据在平台中的设置传入需要相应的类型参数，具体参数设置见下表：

枚举类型	说明
BANNER_AD_UNIT_TYPE	请求广告位中的旗帜广告预缓存单元
INTERSTITIAL_AD_UNIT_TYPE	请求广告位中的插页广告预缓存单元

5.1.3. 广告预缓存事件说明

SmartMad SDK 提供了广告预缓存功能执行时状态事件，方便开发者实时的掌握广告预缓存的动态信息，具体事件说明见下表：

事件	说明
SMAdPrecacheEventHandler.AdPrecacheStarted	广告预缓存已经开始
SMAdPrecacheEventHandler.AdPrecacheError	广告预缓存错误，代码见广告事件
SMAdPrecacheEventHandler.AdPrecacheCompleted	广告预缓存完成

建议：开发者可以根据广告预缓存完成的状态开始正式广告的请求。

5.2. 提供信息辅助广告精确定向

5.2.1. 提供终端用户信息

开发者如果希望提高广告的转化率，可以将一些仅用于广告投放服务的终端用户信息提交到亿动智道广告平台，我们将根据所提供的信息为你进行具有优化效果的广告精准定向投放。这些信息使用 JSON 格式进行传输，具体调用代码如下（粗体字部分）：

```
namespace SmartMadSample {  
    public partial class App : Application {  
        private void Application_Launching(object sender, LaunchingEventArgs e){  
            SMAdManager.setUserInfomation("{\"age\":\"25\",\"gender\":\"true\",\"work\":\"engineer\"}");  
        }  
    }  
}
```

具体可以提交的用户信息内容如下表：

内容项	说明
{"age": "string"}	终端用户年龄信息
{"gender": boolean}	终端用户性别信息，false代表女性，true代表男性
{"birthday": "string"}	终端用户生日信息
{"hobby": "string"}	终端用户爱好信息
{"city": "string"}	终端用户所在城市信息
{"zipcode": "string"}	终端用户所在地邮政编码信息
{"work": "string"}	终端用户工作类型信息

注意：用户信息内容的提交调用需要在正式广告请求调用之前完成。

5.2.2. 提供关键字信息

开发者可以参与并提高广告转化率的另一种方法是将应用的类型或者用于应用程序的关键字提交到亿动智道广告平台，我们将根据所提供的关键字为你进行具有优化效果的广告精准定向投放。具体调用代码如下（粗体字部分）：

```
namespace SmartMadSample {  
    public partial class App : Application {  
        private void Application_Launching(object sender, LaunchingEventArgs e) {  
            // 设置关键字  
            SMAdManager.SetKeywords("angrybirds,game");  
        }  
    }  
}
```

注意：关键字的提交调用需要在正式广告请求调用之前完成，如果有多个关键字使用“,”进行分隔。

5.3. 混淆注意事项（重要）

根据微软 MSDN 中的说明，通过 Visual Studio 开发 Windows Phone 8 Only，编译生成的 xap 文件是 Machine Independent Common Intermediate Language (CIL) 类型的文件，等同于 MSIL。将生成的发布包提交到 Windows Phone Store 时，应用商店会将 xap 文件重新编译 (Compile in the cloud) 成为 Machine Dependent Intermediate Language (MDIL) 类型的文件提供 Windows Phone 8 终端用户下载 (Windows Phone 7 用户仍然提供 JIT 类型文件)，增加了运行效率并减小了被反编译的风险，所以开发者没有必要对提交的发布包进行混淆。

如果开发者添加 SmartMad SDK 后进行发布包编译时，认为必须进行混淆 (微软 WP 默认推荐混淆工具：Dotfuscator)，请在混淆工具的配置中保留 SmartMad SDK 主库文件和另外两个第三方库文件不进行混淆。

5.4.广告事件(SMAdEventCode)说明

SmartMad SDK 在广告正式请求或广告预缓存请求时,通过定义的事件接口会返回事件状态代码,这些状态代码的具体定义说明如下表:

枚举类型	枚举值	说明
SUCCEED	0	执行成功
INVALID_ID	1	唯一标识校验没有通过(例如:广告位)
AD_TOO_MUCH_ON_SCREEN	2	使用了过多的广告位(例如:超过2个)
PERMISSION_INCOMPLETE	3	缺少SDK运行的必要权限
INVALID_AREA_OR_BE_COVERD	4	广告视图加载到了窗口之外
OS_VERSION_TOO_LOW	5	系统版本过低
NETWORK_ERROR	6	网络错误(例如:移动设备无法检测到网络)
INVALID_REQUEST	7	无效的广告请求
NO_FILL	8	有效的广告请求,但无广告填充
INTERNAL_ERROR	9	内部错误