Windows Phone 7.1 mOcean C# SDK

Developer Guide

V 2.9.0

Table of Contents

[Overview 3](#_Toc293481719)

[System requirements: 3](#_Toc293481720)

[SDK contents: 3](#_Toc293481721)

[Installation instructions: 3](#_Toc293481722)

[SDK API reference: 4](#_Toc293481723)

[AdView class members 4](#_Toc293481724)

[Interstitial properties 7](#_Toc293481725)

[Sample usage 8](#_Toc293481726)

## Overview

This document provides a description of the following:

* System requirements
* Installation Instructions
* SDK contents
* SDK API reference
* Sample of usage

### System requirements:

* Microsoft Windows Vista SP2 or Microsoft Windows 7.
* Windows Phone Developer Tools (<http://go.microsoft.com/?linkid=9772716>)
* Microsoft Visual Studio 2010
* 2 Mb free disk space

### SDK contents:

Release version of library

Debug version of library in Debug directory

Samples in Samples directory

### Installation instructions:

*Installing Windows Phone Developer Tools*

Download and install *Windows Phone Developer Tools* (<http://go.microsoft.com/?linkid=9772716>).

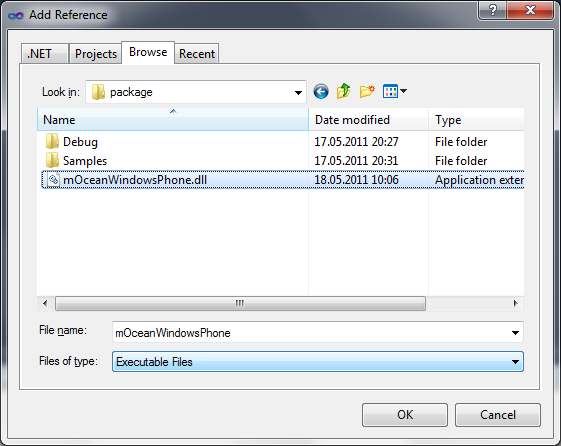
Follow installation instructions provided by Windows Phone Developer Tools. Once Tools has been installed follow to the next step to install Windows Phone mOcean C# SDK.

*Installing Windows Phone 7.1 C# SDK*

The SDK has been coded as a class library.

To add the library to a project, the developer must configure the project properties to add library reference.

Select References in Solution Explorer. Right click and choose Add Reference, locate the mOceanWindowsPhone.dll library and add it.



## SDK API reference:

### AdView class members

#### Setting properties of AdView

##### int Site

The id of the publisher site.

##### int Zone

The id of the zone of publisher site.

##### bool Test

Setting is test mode where, if the ad code is true, the ad response is "Test MODE". (default = false)

##### int Premium

Filter ads by premium status (0 - non-premium, 1 - premium only, 2 - both). Can be used only by premium publishers. (default = 0)

##### String Keywords

Keywords to search ad delimited by commas.

##### int AdsType

Type of ads (1 - text only, 2 - image only, 3 - image and text, 6 - SMS ad). SMS will be ONLY returned in XML and should be used along with key=3. DEPRECATED

##### int Type

Type of ads to be returned (1 - text, 2 - image, 4 - richmedia ad). You can set different combinations with these values. For example, 3 = 1 + 2 (text + image), 7 = 1 + 2 + 4 (text + image + richmedia) and so on. By default parameter is not sent.

##### int MinSizeX

Minimal width of the ad banner to be shown.

It is recommended to set minimum width of control to 320 px according Windows Phone specifications.

##### int MinSizeY

Minimal height of the ad banner to be shown.

It is recommended to set minimum height of control to 480 px according Windows Phone specifications.

##### int MaxSizeX

Maximal width of the ad banner to be shown.

##### int MaxSizeY

Maximal height of the ad banner to be shown.

##### Color BackgroundColor

Background color. Background color of control used if property not setted and background color of control is solid.

##### Color TextColor

Determine the color of links in HEX.

##### String CustomParameters

Custom request parameters (name,value comma separated).

##### String AdServerUrl

Url of ad server (default to <http://ads.mocean.mobi/ad>).

##### Image DefaultImage

Image which will be shown during advertising loading if there is no advertising in a cache.

##### bool InternalBrowser

The flag which operates advertising opening. False - Ad opens in an external browser. True - Ad opening at internal form. (default = false)

##### int AdvertiserId

Advertiser id (will be provided by mocean) (if both AdvertiserId and GroupCode are specified then install notification is enabled)

##### String GroupCode

Group code (will be provided by mocean) (if both AdvertiserId and GroupCode are specified then install notification is enabled)

##### int UpdateTime

Banner refresh interval (in seconds). (Banner refresh interval (in seconds). 0 - disable update.)

##### String Latitude

User location latitude value (given in degrees.decimal degrees). It’s used together with ’long’ parameter. null - enable autodetect(default). "" - don't send coords to server.

##### String Longitude

User location longtitude value (given in degrees.decimal degrees). It’s used together with ’lat’ parameter. null - enable autodetect(default). "" - don't send coords to server.

##### String Country

Country of visitor; it overrides country detected by IP. It is ISO 3166 to be used for specifying country code.

##### String Region

Region of visitor. ISO 3166-2 is used for United States and Canada and FIBS 10-4 is used for other countries.

##### String City

City of the device user (with state). For US only.

##### String Area

Area code of a user. For US only.

##### String Metro

Metro code of a user. For US only.

##### String Zip

Zip/Postal code of user (note: parameter is all caps). For US only.

##### String Carrier

Carrier name.

##### bool Track

allows for client side tracking per ad request (default false)

##### Logger Logger

Logging object

##### Logger.LogLevel LogLevel

Log level (0 - disabled, 1 - errors only (default), 2 - errors and information messages, 3 - errors, information messages and HTTP traffic)

##### ContentAlignment

aligns ad content

#### Interstitial properties

##### uint AutoCloseInterstitialTime

auto-close interstitial time

##### uint ShowCloseButtonTime

show close button after delay

##### Button CloseButton

own close button

#### Methods

##### void Update()

Immediately update banner contents

#### Events

##### EventHandler<NavigatingEventArgs> AdNavigateBanner

This event fired after user clicks the banner (only if InternalBrowser property has been set to True).

url - location proceeding to.

Args property Cancel should be setted to True if you are handled the event and do not want the control to show internal popup window.

##### EventHandler AdDownloadBegin

This event is fired before banner download begins.

EventHandler AdDownloadEnd  
This event is fired after banner content fully downloaded.

##### EventHandler<DownloadErrorEventArgs> AdDownloadError

This event is fired after fail to download content.

##### EventHandler<WebViewClosingEventArgs> AdWebViewClosing

This event is fired after internal web browser closed.

##### EventHandler<ExternalCampaignEventArgs> AdExternalCampaign

This event is fired when ad is third party.

##### EventHandler AdClose

This event is fired when interstitial ad closed.

## Sample usage

See Sample directory in SDK contents.

To run this sample you need Windows Phone Emulator from Windows Phone Developer Tools.

*Create banner control:*

MainPage.xaml

|  |
| --- |
| <phone:PhoneApplicationPage xmlns:my="clr-namespace:mOceanWindowsPhone;assembly=mOceanWindowsPhone">  <my:AdView Name="AdView" HorizontalAlignment="Left" VerticalAlignment="Top" Height="240" Width="480" />  …  </phone:PhoneApplicationPage> |

MainPage.xaml.cs

|  |
| --- |
| public MainPage()  {  InitializeComponent();  adView.Site = 8061;  adView.Zone = 20249;  adView.AdvertiserId = 1234;  adView.GroupCode = "code";  adView.UpdateTime = 50;  } |
|  |

Ad View Control should be child of the LayoutRoot of PhoneApplicationPage with HorizontalAlignment="Left" and VerticalAlignment="Top" for correct working according Ormma specifications.

-- END OF DOCUMENT --