WINDOWS 8 AD ROTATOR FOR UNITY SDK

version 1.0.0.0

Contents

[Figures 2](#_Toc359248124)

[Introduction 3](#_Toc359248125)

[Features 3](#_Toc359248126)

[Pre-Requisites / Supported versions 3](#_Toc359248127)

[SDK Installation 4](#_Toc359248128)

[SDK Unity Design Time Experience 5](#_Toc359248129)

[Menus 5](#_Toc359248130)

[Install Ad Rotator Menu ITem 5](#_Toc359248131)

[Editors 5](#_Toc359248132)

[Configure Ad Rotator 5](#_Toc359248133)

[SDK Unity Scripts API Documentation 8](#_Toc359248134)

[AdRotatorUnitySDK.Assets.Plugins.AdRotatorManagement 8](#_Toc359248135)

[AdRotatorUnitySDK.Assets.Plugins.AdSettings 9](#_Toc359248136)

[AdRotatorUnitySDK.Assets.Plugins.AppSettings 10](#_Toc359248137)

[Building and Configuring Ad Rotator for Your Game - Step By Step 11](#_Toc359248138)

[Generating Windows Store App Solution 11](#_Toc359248139)

[Importing the SDK Unity Package 11](#_Toc359248140)

[Creating the Ad Rotator Management Game Object 12](#_Toc359248141)

[Configuring the Ad Rotator 12](#_Toc359248142)

[Building the Unity Project 12](#_Toc359248143)

[Configuring newly generated Visual Studio solution 13](#_Toc359248144)

[Adding AdRotator for Windows 8 with Nuget Manager and Building the Game 13](#_Toc359248145)

[Demo Installation and Execution 15](#_Toc359248146)

[Limitations / Restrictions / Known Issues 16](#_Toc359248147)

# Figures

[Figure 1 - AdRotatorUnitySDK.unitypackage - Importing Package 4](#_Toc359248148)

[Figure 2 - Ad Rotator Management Settings 5](#_Toc359248149)

[Figure 3 - AdRotatorManagement Sample Code 8](#_Toc359248150)

[Figure 4 - Importing Unity Package 11](#_Toc359248151)

[Figure 5 - Unity Build Settings - Windows Store Apps 12](#_Toc359248152)

[Figure 6 - Manage Nuget Packages Menu 13](#_Toc359248153)

[Figure 7 - Manage Nuget Packages Window 13](#_Toc359248154)

[Figure 8 - Nuget Select Projects Window 13](#_Toc359248155)

[Figure 9 - Nuget Installation Window 14](#_Toc359248156)

[Figure 10 - Visual Studio Configuration Manager 14](#_Toc359248157)

# Introduction

Ad Rotator for Windows 8 is a control library used to generate revenue from your applications by displaying ads relevant to your users. Ad Rotator has the following major features:

* Supports multiple Ad Providers:
  + AdDuplex: <http://www.adduplex.com/>
  + Microsoft pubCenter: <http://adsinapps.microsoft.com/pubcenter>
  + Custom House Ads: design and implement your own ads
* Ability to support decide which Ad Provider is used when:
  + Ad Provider distribution and configuration selection based on the culture
  + Distribution of Ad displayed based on a provider weight system
* Support for local and remote Ad Provider configuration:
  + Ability to change your ad configuration remotely without publishing your application again
* Sliding Ads that appear and disappear in order to draw more attention to them

For a complete list of Ad Rotator for Windows 8 features or for any issue with the control library, please visit the following CodePlex project: <http://wp7adrotator.codeplex.com/>

The aim of this SDK is to enable Unity developers to use Ad Rotator transparently without leaving the environment they know well, using the API they are used to work with.

# Features

* Ad Rotator Unity design time experience:
  + Ad Provider settings (default App providers, remote and local configuration...)
  + Ad Layout display settings used at game startup (position, size, sliding or not)
  + Ad Layout settings can be overridden later on at runtime by accessing the Ad Rotator Unity SDK APIs through Unity scripts.
  + When building project, code and project is adjusted to ease Ad Rotator usage and configuration
* Ad Rotator component is shared between Unity scenes:
  + Need to configure Ad Rotator only once in the initially loaded scene.
  + Unity scripts can be used to control Ad Rotator in any scene of the game.
  + Ads can be shown while changing scene.
* Ad Rotator can be dynamically moved inside the game screen and resized to adjust to game content
  + See Limitations/Restrictions section for resizing limitations.
* Ad Rotator can be dynamically enabled or disabled depending on the developer needs

# Pre-Requisites / Supported versions

* Pre-requisites:
  + Unity version 4.2.0b3 with Metro Support for Editor 4.2.0.8573.
    - <http://unity3d.com/beta/windowsstoreapps>
* Microsoft Advertising SDK for Windows 8 (Xaml) version 6.1.0.0
  + - <http://go.microsoft.com/?linkid=9815330>
* Supported versions:
  + Ad Rotator for Windows 8 version 1.0.10
    - <http://nuget.org/packages/AdRotatorWin8/>

Sdk

# SDK Installation

Please refer to Unity documentation to know how to import a Unity package into your project. Otherwise, refer to section *Building and Configuring Ad Rotator for Your Game - Step By Step* to see the details of the importation steps.

Ad Rotator for Unity SDK is provided as a unity package. The name of the package is: *AdRotatorUnitySDK.unitypackage*

While importing, please select all Items in order to have the complete SDK included in your own project:

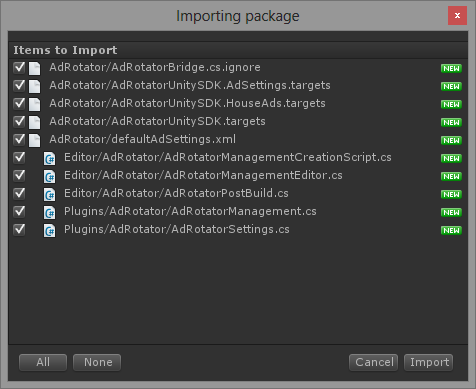


Figure 1 - AdRotatorUnitySDK.unitypackage - Importing Package

Once importation is completed, the Ad Rotator design time experience will be enabled.

# SDK Unity Design Time Experience

## Menus

### Install Ad Rotator Menu ITem

This menu is used in order to configure your scene to use the Ad Rotator for Unity SDK. Using this menu item will trigger the following operations on your behalf:

* Create a new Game Object in the current scene named *AdRotatorManagement*
  + **Note that this Game Object shall not be renamed.**
* Configure Unity Player settings (Edit🡪Project Settings🡪Player, then Capabilities) to enable Internet Client capability in order to have Ad Rotator properly working.
* Select the newly created *AdRotatorManagement* Game Object, trigger the related Editor to be displayed in the *Inspector* window

This menu is available from two different locations:

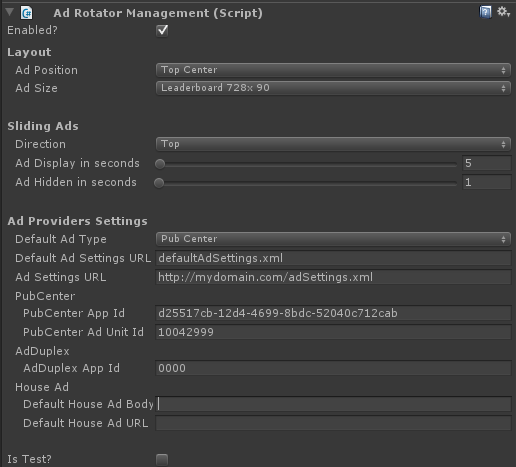
* From the main menu:
  + GameObject🡪Create Other🡪Ad Rotator
* From the *Hierarchy* window:
  + Create🡪Ad Rotator

## Editors

This section presents the different visual editors available in the design time experience. Refer to the *SDK Unity Scripts API Documentation* for more information about the APIs stated in this section.

### Configure Ad Rotator

This editor is used to configure the Ad Layout settings and the Ad Provider settings. The editor has the following fields available. Note that each field document can be seen through a tool tip on the field name. Note that the settings presented in this editor and directly taken from Ad Rotator supported settings, please refer to Ad Rotator for detailed information.



Ad Providers Settings

Layout Settings

Figure 2 - Ad Rotator Management Settings

#### Layout Settings

Note that the Ad Layout settings will be automatically configured when your Windows Store App will be initially starting and can be changed later on while the game is running through the APIs.

* Enabled?
  + Check to enable Ad Rotator. Ads will not be displayed if disabled.
  + Note that even if Ad Rotator is disabled, you can still configure Layout settings which will be used as the pre-configured value when the application is starting.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.IsEnabled*
* Layout / Ad Position
  + Screen position of the Ad banner. The following positions relative to the scene are supported:
    - Top Left Corner
    - Top Right Corner
    - Bottom Left Corner
    - Bottom Right Corner
    - Top Center
    - Bottom Center
    - Left Center
    - Right Center
    - Center
  + Note that Ad will be displayed in front of the scene, potentially hiding scene content. Select position accordingly.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.Position*
* Layout / Ad Size
  + Size of the Ad Banner. Only the following sizes are supported in Windows 8:
    - Medium Rectangle 300x250
    - Leaderboard 728x90
    - Wide Skyscraper 160x600
    - Square Pop Up 250x250
    - Split View Banner 500x130
    - Snap View Banner 292x60
    - Half Tile 250x125
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.Size*
* Sliding Ad / Direction
  + Set *Sliding Ad Direction* to either Left, Right, Bottom or Top to have the ad slide in, stay for *Sliding Ad Display in Seconds*, slide out and stay hidden for *Sliding Ad Hidden in Seconds*. If *Sliding Ad Direction* is set to *None*, this behavior does not take place, the ad remains static.
  + Note that even if value is *None*, you can configure the Display and Hidden delays to be used as default value once application is started.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.SlidingAdDirection*
* Sliding Ad / Ad Display in Seconds
  + Display time of the ad, in seconds. See Sliding Ad🡪Direction.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.SlidingAdHiddenSeconds*
* Sliding Ad / Ad Hidden in Seconds
  + Hidden time of the ad, in seconds. See Sliding Ad🡪Direction.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AdSettings.SlidingAdHiddenSeconds*

#### Ad Providers Settings

Note that the Ad Providers settings will be automatically configured when your Windows Store App will be initially starting and can be changed later on while the game is running through the APIs.

* Ad Provider Settings / Default Ad Type
  + Ad type that should be shown if either the ad settings file could not be loaded or other ad providers have failed to load.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.DefaultAdType*
* Ad Provider Settings / Default Ad Settings URL
  + URI to a local XML file that will be used if the remote file specified with Ad Settings Url could not be loaded. Example: defaultAdSettings.xml
  + Should use defaultAdSettings.xml value to have automatic integration with the build.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.DefaultSettingsFileUri*
* Ad Provider Settings / Ad Settings URL
  + URL to the remote XML file that controls the probability of ad providers shown. Strongly advised to set this property. Example: http://mydomain.com/myAdSettings.xml
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.SettingsUrl*
* Ad Provider Settings / PubCenter / PubCenter App Id
  + App ID to use with Microsoft PubCenter Ad Provider.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.PubCenterAppId*
* Ad Provider Settings / PubCenter / PubCenter Ad Unit Id
  + Ad unit ID to use with Microsoft PubCenter Ad Provider.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.PubCenterAdUnitId*
* Ad Provider Settings / AdDuplex / AdDuplex App Id
  + App ID to use with AdDuplex Ad Provider.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.AdDuplexAppId*
* Ad Provider Settings / House Ad / Default House Ad Body
  + Name of the XAML control to use (format: <namespace>.<object name>), need to be inside your application assembly or referenced by your project. See AdRotator documentation for more information. Example: AdRotatorExample.MyDefaultAd
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.DefaultHouseAdBody*
* Ad Provider Settings / House Ad / Default House Ad URL
  + URL to the remote XAML file to use as the House Ad. See AdRotator documentation for more information. Example: http://mydomain.com/myHouseAd.xaml
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.DefaultHouseAdUri*
* Ad Provider Settings / Is Test?
  + Enable or disable ad Test mode. See AdRotator documentation for more information.
  + See also API *AdRotatorUnitySDK.Assets.Editor.AdRotatorManagement.AppSettings.IsTest*

# SDK Unity Scripts API Documentation

The SDK can be used to control at runtime the Ad Rotator appearance and display.

## AdRotatorUnitySDK.Assets.Plugins.AdRotatorManagement

Component responsible of managing interactions and settings of Ad Rotator. This component is available of the GameObject named *AdRotatorManagement*. The code below shows how to modify and update the Ad Layout settings.

|  |
| --- |
| var ads = GameObject.Find("AdRotatorManagement");  if (ads != null)  {  var adMgt = ads.GetComponent<AdRotatorManagement>();  adMgt.AdSettings.IsEnabled = false;  adMgt.AdSettings.Position = AdPosition.TopCenter;  adMgt.UpdateAd();  } |

Figure 3 - AdRotatorManagement Sample Code

|  |  |
| --- | --- |
| Class Variables |  |
| AdSettings | Ad Layout Settings. See *AdRotatorUnitySDK.Assets.Plugins.AdSettings* for more info. |
| AppSettings | Ad Provider Settings. See *AdRotatorUnitySDK.Assets.Plugins.AppSettings* for more info. |
| OnSettingsUpdated | This variable is primarily intended for internal use only to communicate with Windows 8. Do not modify this variable. |

|  |  |
| --- | --- |
| Class Functions |  |
| UpdateAd | Notify the UI Shell that Ad Rotator layout settings need to be updated. Prior calling this method, *AdSettings* and/or *AppSettings* values needs to be updated. See sample code above for example. |

|  |  |
| --- | --- |
| Inner Members |  |
| OnSettingsUpdatedHandler | This delegate is primarily intended for internal use only to communicate with Windows 8. Do not use. |

## AdRotatorUnitySDK.Assets.Plugins.AdSettings

Currently showing Ad settings.

|  |  |
| --- | --- |
| Class Variables |  |
| IsEnabled : bool | Use to enable Ad Rotator. Ads will not be displayed if disabled. |
| Position : AdPosition | Screen position of the Ad banner.  The *AdPosition* enumeration from the same namespace as the class supports the following values :   * TopLeftCorner * TopRightCorner * BottomLeftCorner * BottomRightCorner * TopCenter * BottomCenter * LeftCenter * RightCenter * Center |
| Size : AdSize | Size of the Ad Banner.  The *AdSize* enumeration from the same namespace as the class supports the following values :   * MediumRectangle300x250 * Leaderboard728x90 * WideSkyscraper160x600 * SquarePopUp250x250 * SplitViewBanner500x130 * SnapViewBanner292x60 * HalfTile250x125 |
| SlidingAdDirection : SlidingAdDirection | Set Sliding Ad Direction to either Left, Right, Bottom or Top to have the ad slide in, stay for Sliding Ad Display in Seconds, slide out and stay hidden for Sliding Ad Hidden in Seconds. If Sliding Ad Direction is set to None, this behaviour does not take place, the ad remains static.  The *SlidingAdDirection* enumeration from the same namespace as the class supports the following values :   * None * Bottom * Left * Right * Top |
| SlidingAdHiddenSeconds : int | Display time of the ad, in seconds. See *SlidingAdDirection* for more information. |
| SlidingAdDisplaySeconds : int | Hidden time of the ad, in seconds. See *SlidingAdDirection* for more information. |

## AdRotatorUnitySDK.Assets.Plugins.AppSettings

Application settings of Ad Rotator.

|  |  |
| --- | --- |
| Class Variables |  |
| SettingsUrl : string | URL to the remote XML file that controls the probability of ad providers shown. Strongly advised to set this property. Example: http://mydomain.com/myAdSettings.xml |
| DefaultSettingsFileUri : string | URI to a local XML file that will be used if the remote file specified with Ad Settings Url could not be loaded. Example: defaultAdSettings.xml |
| DefaultAdType : AdProvider | Ad type that should be shown if either the ad settings file could not be loaded or other ad providers have failed to load.  The *AdProvider* enumeration from the same namespace as the class supports the following values :   * None * PubCenter * AdDuplex * DefaultHouseAd |
| AdDuplexAppId : string | App ID to use with AdDuplex Ad Provider. |
| PubCenterAppId : string | App ID to use with Microsoft PubCenter Ad Provider. |
| PubCenterAdUnitId : string | Ad unit ID to use with Microsoft PubCenter Ad Provider. |
| DefaultHouseAdBody : string | Name of the XAML control to use (format: <assemblyname>.<classname>), need to be inside your application assembly or referenced by your project. See AdRotator documentation for more information. Example: AdRotatorExample.MyDefaultAd |
| DefaultHouseAdUri : string | URL to the remote XAML file to use as the House Ad. See AdRotator documentation for more information. Example: http://mydomain.com/myHouseAd.xaml |
| IsTest : bool | Enable or disable ad Test mode. See AdRotator documentation for more information. |

# Building and Configuring Ad Rotator for Your Game - Step By Step

The following section will show you how to configure your Unity project to use the Ad Rotator SDK and building it for the first time.

## Generating Windows Store App Solution

### Importing the SDK Unity Package

1. Make sure that the *Microsoft Advertising SDK for Windows 8 (Xaml)* pre-requisite is installed.
2. Start Unity and open your project.
3. Go into the *Project* window and right click on the *Asset* folder
4. Select menu item Import Package🡪Custom Package…

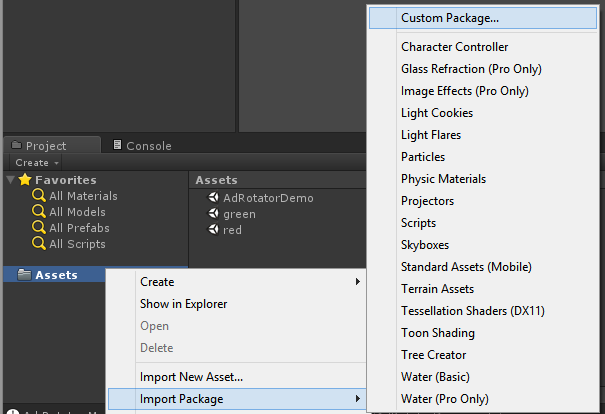


Figure 4 - Importing Unity Package

1. Browse and open the Ad Rotator for Unity SDK unity package named *AdRotatorUnitySDK.unitypackage*
2. Select all items from the package and press *Import*.
3. The following assets will be deployed:
   1. **Assets\AdRotator\AdRotatorUnitySDK.targets**: Contains files necessary to integrate the Unity to AdRotator bridge. This file is also including the references to *AdRotator* and *AdDuplex*, pointing to a specific version.
   2. **Assets\AdRotator\AdRotatorUnitySDK.AdSettings.targets**: File modified every time a Unity build is done to include all XML files from the **Assets/AdRotator** folder.
   3. **Assets\AdRotator\AdRotatorUnitySDK.HouseAds.targets**: File modified every time a Unity build is done to include all XAML files from the **Assets/AdRotator** folder.
   4. **Assets\AdRotator\defaultAdSettings.xml**: Ad Rotator settings file that will be deployed with your Windows Store App. Edit this file to align with your own Ad Provider settings. See Ad Rotator documentation for file format.
      1. Make sure that the Ad Rotator Management (Ad Provider Settings / Default Ad Settings URL) is configured with the name defaultAdSettings.xml.
   5. **Assets\AdRotator\AdRotatorBridge.cs.ignore**: Code used to hook Unity and the Windows 8 application. Do not modify.
   6. **Assets\Editor\AdRotator\AdRotatorManagementCreationScript.cs**: Unity script used to install AdRotatorManagement component in the current scene. Trigger by the menu GameObject🡪Create Other🡪Ad Rotator.
   7. **Assets\Editor\AdRotator\AdRotatorManagementEditor.cs**: Unity script containing the editor used to edit the AdRotatorManagement component.
   8. **Assets\Editor\AdRotator\AdRotatorPostBuild.cs:** Unity script containing the post build step used to modify the output project of the build.
   9. **Assets\Plugins\AdRotator\AdRotatorManagement.cs**: Unity script containing the component used to manage settings and communication between Unity and the application.
   10. **Assets\Plugins\AdRotator\AdRotatorSettings.cs**: Source code of the AdRotator settings and associated enumeration values.
4. **Note** that any additional XML and XAML files added to **the Assets\AdRotator** will be automatically imported in the Windows Store App generated solution.
   1. XML files will be added as *Content* to be deployed with the application.
   2. XAML files will be added as *Page* to be compiled and included in the Visual Studio project, used for House Ads.

### Creating the Ad Rotator Management Game Object

1. Open the scene being initially loaded in your project
2. Go to the *Hierarchy* window.
3. Select Create🡪Ad Rotator
   1. A new Game Object named *AdRotatorManagement* has been added and selected in your hierarchy. Do **NOT** rename it.

### Configuring the Ad Rotator

1. Go to the *Inspector* window in the *Ad Rotator Management* section. Refer to section *This section presents* the different visual editors available in the design time experience. Refer to the *SDK Unity Scripts API Documentation* for more information about the APIs stated in this section.
2. Configure Ad Rotator to learn how to populate the values properly.

### Building the Unity Project

1. Go to the menu File🡪Build Settings…
2. Select the Platform *Windows Store Apps*.
3. Select the build type *XAML C# Solution*.
   1. Note that this is the only supported build type.

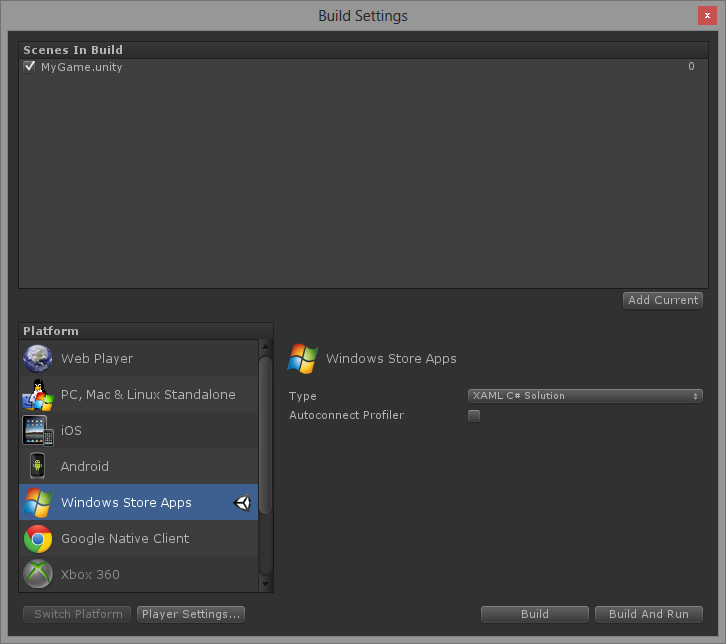


Figure 5 - Unity Build Settings - Windows Store Apps

1. Make sure the initial scene with the *AdRotatorManagement* game object is the first in the list.
2. Press the Build button.
3. Select the target folder and press Select Folder.
4. Wait for the build to complete. Check there are not errors.

## Configuring newly generated Visual Studio solution

### Adding AdRotator for Windows 8 with Nuget Manager and Building the Game

1. Go to the target folder where the build output was generated.
2. Open the solution containing the generated Windows Store App project.
   1. Solution and project files are usually named with the same name as your Unity project.
3. In the *Solution Explorer* window, right-click on the solution.
4. Select menu item *Manage Nuget Packages…*.

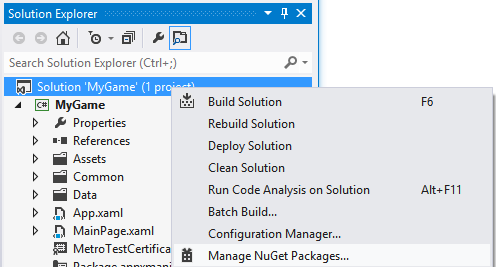


Figure 6 - Manage Nuget Packages Menu

1. Once in the *Manage Nuget Packages* window, search for the AdRotator package:

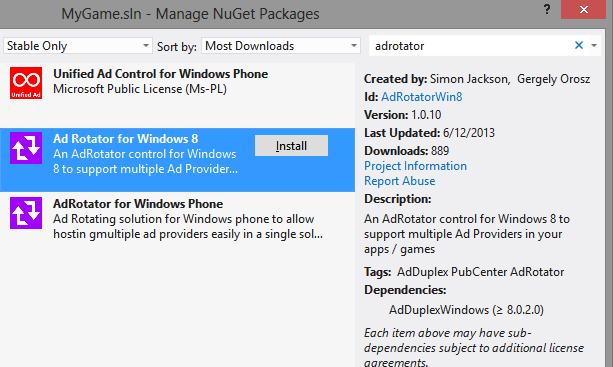


Figure 7 - Manage Nuget Packages Window

1. Select *Ad Rotator for Windows 8* and click *Install*. When prompted to select the project, make you’re your Windows Store App project is selected and press OK:

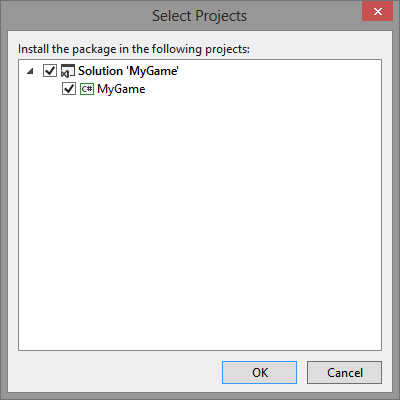


Figure 8 - Nuget Select Projects Window

1. An installation window will appear, check there are no errors. Note that the AdDuplex dependency will be installed at the same time.

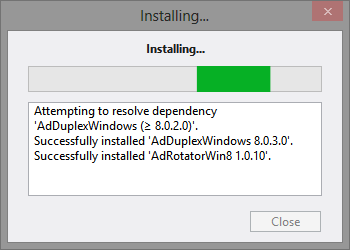


Figure 9 - Nuget Installation Window

1. Close the *Manage Nuget Packages* window.
2. Go back in the *Solution Explorer* window, find the *app.config* file that was added by Nuget and remove it. Not removing the file will prevent your game from executing.
3. Expand the *References* node of the solution, check that the following references are correctly listed without a yellow warning icon, they are mandatory:
   1. References:
      1. *AdRotatorWin8*
      2. *AdDuplex.Windows*
      3. *Microsoft Advertising SDK for Windows 8 (Xaml)*
   2. **Note**: If any of these 3 mandatory references is showing the warning icon (example: ), AdRotator will not be able to load correctly and that will result in either a compilation error or a crash of the application when starting.
   3. **Note**: By default, the references configured in the project are using the supported version of AdRotator as documented in this document. If a newer version of AdRotator is required, simply modify the file *AdRotatorUnitySDK.targets* found in the *Assets/AdRotator* in the Unity projet, and update the reference folder name accordingly.
4. Change the build target to target x86 to validate that the application is correctly working locally.

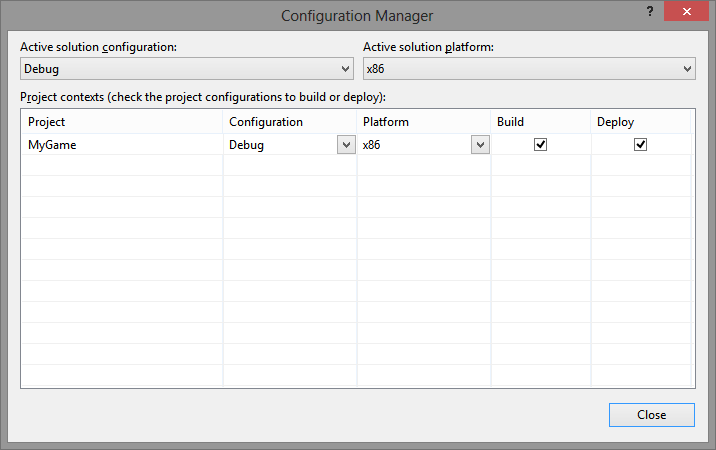


Figure 10 - Visual Studio Configuration Manager

1. Build and execute the game.
2. The game should start using the configured Ad settings.
   1. Note that Ad Rotator logs will be forwarded to the Unity debug logs to ease troubleshooting.
   2. Ad Rotator logs can also be seen in *Debug Output* window of Visual Studio.

sdk DEMO

# Demo Installation and Execution

The following section will show you how to configure your Unity project to use the Ad Rotator SDK and building it for the first time.

1. Create a new Unity project with a scene and save it.
2. Follow the Building and Configuring Ad Rotator for Your Game - Step By Step, but:
   1. After importing the *AdRotatorUnitySDK.unitypackage* package, import the *AdRotatorUnitySDKDemo.unitypackage* and select all items for that package as well.
      1. The demo package contains the following assets:
         1. **Assets/Editor/ConfigureDemoScript.cs:** Contains Unity script used to configure the scenes to correctly display the demo.
         2. **Assets/Plugins/DisplayDemo.cs**: Contains Unity script to display onscreen menu and related operations to modify Ad Rotator configuration. Example of how to use this SDK.
         3. **Assets/AdRotator/HouseAdDemo.xaml**: Custom House Ad to be added to the project when building.
         4. **Assets/AdRotator/defaultAdSettingsDemo.xaml**: Ad Rotator settings file that will be deployed with your Windows Store App. Demo specific values.
         5. **Asssets/AdRotatorDemo.unity**: Initial scene of the Demo, where Ad Rotator will be configured.
         6. **Asssets/green.unity**: Supplementary scene used with the Demo.
         7. **Asssets/red.unity**: Supplementary scene used with the Demo.
   2. Instead of executing section Configuring the Ad Rotator, go to the menu:
      1. Edit🡪Configure Ad Rotator Demo
3. Finish the procedure and run the game.
4. Use the displayed button to move the Ad around or change its behavior.
   1. Use the *Switch Scene* button to change the scene currently displayed.
   2. Use the *Disable Ad* button to stop displaying ads.

# Limitations / Restrictions / Known Issues

* Unity Build:
  + Only Windows Store Apps with “XAML C# Solution” build type is supported.
  + Windows Store Apps Visual Studio solution generated by Unity needs to be configured through “Manage NuGet Packages” in order to have AdRotator for Windows 8 installed.
  + Need to install Microsoft Advertising SDK for Windows 8 to use Ad Rotator
  + Need to install AdDuplex for Windows 8 to use Ad Rotator
    - Installed automatically with AdRotator for Windows 8 Nuget package
* Same restrictions as Ad Rotator, as documented on their project page, including the following:
  + Windows App Visual Studio Project needs to have reference to all supported Ad Providers, even if specific Ad Provider is not used
  + Only PubCenter, AdDuplex and House Ads are supported for Windows 8
* PubCenter Ads require a different set of settings to be resized dynamically:
  + This is a limitation of the configuration file of AdRotator for AdUnitId which are pre-determined in the configuration and not attached to the current Ad display size.
    - AdDuplex seems however to correctly handle resizing out of the box.
  + For proper behavior when both PubCenter and AdDuplex are used, an easy solution is to select only one supported size, and only change position while keeping the same size (showing top or bottom, for example).
  + When resizing a campaign containing PubCenter ads, a workaround is to do the following to the Ad Provider settings each time dimension is changed:
    - Change the PubCenter settings for AdUnitId to use a proper value for the new size (see Microsoft documentation for AdUnitId values): <http://msdn.microsoft.com/en-us/library/advertising-windows-test-mode-values(v=msads.10).aspx>)
    - To do so, change the “defaultAdSettings.xml” setting and URL to use a specific one with the new size. For example:
      * Change *Ad Provider Settings / Default Ad Settings URL* with a local file pointing to *defaultAdSettings\_728x90.xml* containing appropriate AdUnitId for that size.
        + Simply add the new file in the Unity project in Assets\AdRotator, it will be repackaged a new Unity build.
      * Change *Ad Provider Settings / Ad Settings URL* to point to a URL with a remote file pointing to defaultAdSettings\_728x90.xml containing appropriate AdUnitId for that size.
* AdRotator seems to have some issues with House Ads:
  + AdRotator seems to have an issue with local House Ads which are not being loaded correctly.
    - Do not define any local House Ads to work around the problem (*Ad Provider Settings / House Ad / Default House Ad Body* should be empty)
  + In some cases, AdRotator will display two ads over each other (House Ad over AdDuplex for example).
    - No workaround found.
* Note that AdRotator is using a caching mechanism internally for configuration that could cache values that were previously used but that were changed.
* Depending on your Unity project setup, it might be mandatory to interact with this SDK through Unity plugins written in C#.
* Ad Rotator for Unity SDK (*AdRotatorManagement*) Game Object needs to be configured and present in the initial scene of the game to allow the Windows Store App to hook with Unity correctly.
* Ad Rotator will be displayed over the Unity rendered scene, as an overlay.
  + It is the responsibility of the game developer to make sure the Ads are not blocking the game (game scene resizing, moving the Ad in the screen…)
* Cannot specify exact position (X, Y) of AdRotator on the screen but rather an absolute pre-defined position relative to the screen (top, bottom…).