

# UWP - Xbox Live Programming Guide

---

Updated: April 15, 2017

**This document is confidential** and provided to you under a Microsoft non-disclosure agreement. While we have tried to ensure the accuracy of this document, we provide no express warranties or guarantees regarding the information. The information is subject to change. Microsoft may have intellectual property rights in the subject matter of this paper. This document doesn't grant you a license to those rights—it's for informational purposes only.

## Abstract

This guide provides Xbox Live features and service components that enable you to use Xbox Live Achievement and Leaderboard services from your Universal Windows App.

## Contents

Environment.....	1
Xbox Live SDK & Docs.....	1
1. Xbox Live Service API (Known as Xbox Live SDK) .....	1
2. Xbox Live SDK Documentation.....	1
3. Xbox Live Tools package .....	2
4. Optional XDK Downloads Page .....	2
Pre-coding Preparation .....	2
1. Apply ID@Xbox / Register UDC.....	2

2. Preparation on UDC .....	3
1) Create a new app in dashboard.....	3
2) Retrieve PFN .....	3
3) Retrieve MSA Client id .....	3
3. Preparation on XDP.....	4
1) Create Product Group under Publisher .....	4
2) Create Product under Product Group .....	5
3) Create Dev Account: LINK.....	7
4) Access Management for Group.....	8
5) Create sandbox and set access permission for an access group (XDP portal->Manage->Sandboxes).....	9
4. Associate UDC product with the XDP title .....	10
1) Application ID Setup in XDP .....	10
2) Dev Center Binding in XDP.....	10
Dev Guidance .....	11
1. Configuring your development PC (Sandbox Setup) .....	11
1) Xbox Live Account Tool.....	11
2) Verify Dev Environment.....	11
2. Troubleshooting issues when integrating Xbox Live .....	11
3. Make integration works in UWP app (Unity to UWP) .....	12
4. Achievement configuration on XDP.....	18
1) Achievement configuration in sandbox.....	18
2) After Achievement configuration on XDP, please publish to make changes work.....	18
5. Leaderboard configuration on XDP .....	19

1) Leaderboard configuration in sandbox .....	19
2) After Achievement configuration on XDP, please publish to make changes work.....	19
6. The additional code work in converted UWP project as follows (If using XboxLiveIntegration class library) .....	19
1) Initialize and Sign in Xbox Live.....	19
2) Using AchievementManager class to write game event.....	20
3) Using LeaderboardManager class to retrieve Leaderboard data .....	21
Scenario Sample.....	21
Goal: First Win achievement for one game character and leaderboard implementation.....	21
Step 1: Achievement Design .....	21
Step 2: XDP Configuration.....	21
1) Event & Stat Rules .....	21
2) Achievement.....	25
3) Leaderboard.....	28
4) Publish Service Configuration.....	29
Step 3: Development work .....	30
1) Initialize and Sign in Xbox Live.....	30
2) Handle Keyboard/Gamepad/Joystick Input .....	30
3) Using AchievementManager class to write game event.....	30
4) Using LeaderboardManager class to retrieve Leaderboard data .....	33
Step 4: Debug and verify Achievement & Leaderboard data .....	33

## Environment

- 1) **Unity Version:** Unity 5.3+
- 2) **Windows Version:** Windows 10.10586 or later version, Windows 10 Anniversary Edition (10.0; Build 14393) is recommended
- 3) **Visual Studio Version:** Visual Studio 2015 Update 3 or later version
- 4) **Xbox Live SDK Version:** 1611 or later version
- 5) **Publishing platform location:** [XDP/UDC](#), which used to allow developer to do Live Service Configuration and publish title.

## Xbox Live SDK & Docs

### 1. Xbox Live Service API (Known as Xbox Live SDK)

<https://github.com/Microsoft/xbox-live-api>

Preview SDK: <https://developer.xboxlive.com/en-us/live/development/Pages/Downloads.aspx>

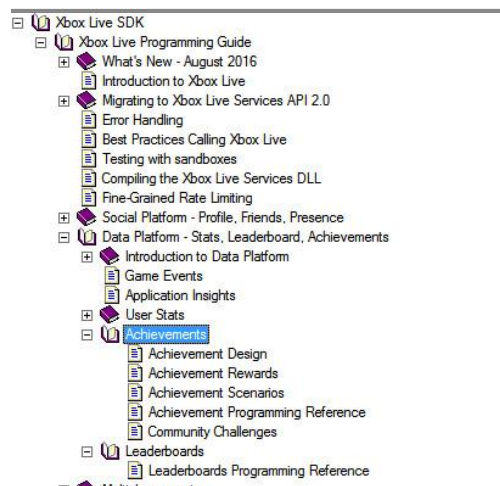
We need to download **Xbox Live SDK & Xbox Live Platform Extensions SDK**

### 2. Xbox Live SDK Documentation

<https://aka.ms/xbldocs>

<https://developer.xboxlive.com/en-us/platform/development/documentation/Pages/home.aspx>

**Xbox Live SDK Documentation (CHM)**, see **Achievement & Leaderboard** part



### 3. Xbox Live Tools package

<https://aka.ms/xboxliveuwptools>

### 4. Optional XDK Downloads Page

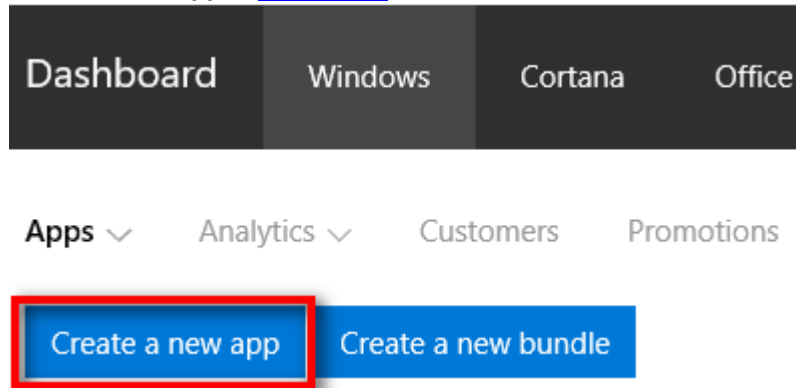
<https://developer.xboxlive.com/en-us/platform/development/downloads/Pages/optional-xdk-downloads.aspx>

## Pre-coding Preparation

### 1. Apply ID@Xbox / Register UDC

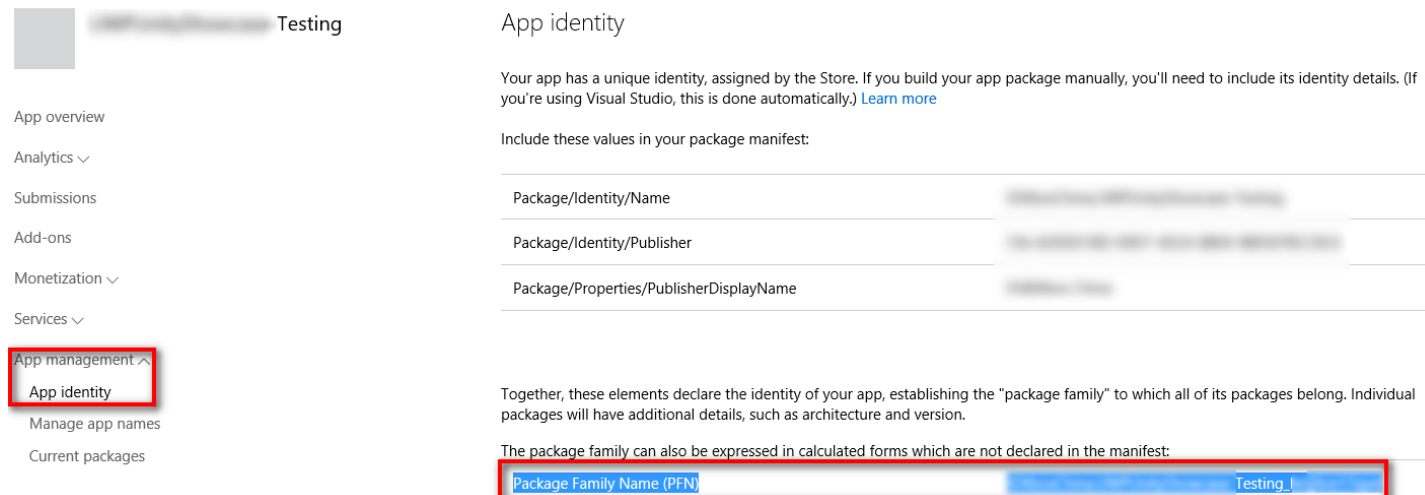
## 2. Preparation on UDC

### 1) Create a new app in [dashboard](#)



### 2) Retrieve PFN

App management->App identity-> Package Family Name (PFN)




### 3) Retrieve MSA Client id

- Open [Windows Live application management site](#)

- ii. Login with **UDC** account
- iii. Find the client id for your UDC product in “**Live SDK applications**” list

Live SDK applications [Learn More](#) Add an app

Name	App ID / Client Id	
 Testing	00000000-0000-0000-0000-0000000009	Delete

### 3. Preparation on XDP

#### 1) Create Product Group under Publisher

*Please working with your developer account manager, or seeking support from [idsetup@xbox.com](mailto:idsetup@xbox.com)*

#### Publishers

Show 10 entries

Search:

Publisher	Product Groups	Products
Microsoft 	1	2
Microsoft 	3	4

Showing 1 to 2 of 2 entries

◀ 1 ▶

#### Product Groups

Show 10 entries

Search:

Name	Products	Action
 Testing Product	2	Rename

Showing 1 to 1 of 1 entries

◀ 1 ▶

2) Create Product under Product Group

- i. Under Product Group page, click “Add New Product”

Microsoft Testing Product

Add New Product

Bulk Publish DLC

Show 10 entries

Search:

- ii. Fill the setup table

CONFIDENTIAL



# Product Setup

Give your new creation a name, you can change it anytime prior to certification.

Product Name

Product Type

Sandbox

Platform Type

- ☒ Xbox One
- ☒ Windows 10 PC
- ☐ Windows Mobile
- ☐ Android
- ☐ iOS

Resource Access

- ☐ Shared Resource Access (SRA)
- ☒ Exclusive Resource Access (ERA)

What is this product intended for?

- ☐ Physical and digital media
- ☒ Digital media only

Create Product

Cancel

- ❖ Please notice that the sandbox of your product can't be reset, otherwise you create a new product
- ❖ For Platform Type, please check "Windows 10 PC"

### iii. Retrieve TitleId and PrimaryServiceConfigId

Product Setup -> Product Details:

Product Details	Product Name	Test
Application ID Setup	Product Type	Game
Dev Center Binding	Platform Type	Xbox One, Windows 10 PC
Platform Type	Title ID	7
	Title ID (decimal)	95
	Scid	894 df7

Scid is PrimaryServiceConfigId

### 3) Create Dev Account: [LINK](#)

Profile	Manage Development Accounts	<a href="#">+ Add Development Accounts</a>
Access	batch active	
Development Accounts		
Email Notifications		

Batch Name ^	Email	Creation Date (UTC)	Modified Date (UTC)	Account Security	Processing / Succeeded / Requested

After creating Dev Account, please wait a while. Ensure "Processing/Succeeded/Requested" column shows 0/1/1 and the new created account is listed in the active page

Batch Name ▲	Email	Creation Date (UTC)	Modified Date (UTC)	Account Security	Processing / Succeeded / Requested	
test	franklin	2/6/2017 6:45 AM	2/6/2017 6:45 AM	Protected	0/1/1	<a href="#">Remove</a>
test2	ft	3/6/2017 8:28 AM	3/6/2017 8:28 AM	Protected	0/1/1	<a href="#">Remove</a>
test3	ft1	3/7/2017 6:12 AM	3/7/2017 6:12 AM	Protected	1/0/1	

Available

Page Size: 10 Showing 1 - 3 of 3



batch active

Batch Name ▼

Search

Batch Name ▲	Account Email Address	Creation Date (UTC)	Account Type	Account Security	XUID	
test	@xboxtest.com	2/6/2017	Gold	Protected		<a href="#">Remove</a>
test2	@xboxtest.com	3/6/2017	Gold	Protected		<a href="#">Remove</a>
test3	@xboxtest.com	3/7/2017	Gold	Protected		<a href="#">Remove</a>

#### 4) Access Management for Group

Manage ▼

Access Management

Approval Dashboard

Console Conversion

Sandboxes

Web Services

Add your XDP login account to an access group, for example: Administrators

Groups <sup>?</sup>

---

MY GROUPS

---

Add Group <sup>+</sup>

---

CIUS.Administrators <sup>→</sup>

5) Create sandbox and set access permission for an access group (XDP portal->Manage->Sandboxes)

## Sandboxes

Add new sandbox

Show  entries

Search:

Name <sup>↑</sup>	Description <sup>↑</sup>	Groups <sup>↑</sup>
M...0	Default	<div>1</div>
M...1		

Showing 1 to 2 of 2 entries

Groups Access

☒ Administrators

#### 4. Associate UDC product with the XDP title

##### 1) Application ID Setup in XDP

In the **Application ID Setup** page, fill the Application Id with [MSA Client ID](#). We just need to fill Application ID part, click “Submit Application ID Setup” button

Product Details →

Application ID Setup →

Dev Center Binding →

Platform Type →

Sample AppXManifest.xml Identity Node

```
<Identity Name="SampleName"
  ProcessorArchitecture="neutral"
  Publisher="CN=Sample Publisher Corporation, O=
    S=Hawaii, C=US"
  Version="1.0.0.0"
  ResourceId="NorthAmerica" />
```

Please enter Name and Publisher attributes below as it appears in the Identity

AppManifest Identity Name attribute:

AppManifest Identity Publisher attribute:

Application ID:  [Get Application ID](#)

**Submit Application ID Setup** ?

##### 2) Dev Center Binding in XDP

In the **Dev Center Binding** page, fill in [PFN](#)

Product Details →

Application ID Setup →

Dev Center Binding →

Platform Type →

If this XDP product is sharing Xbox Live configuration with a Dev Center game (UWP), please enter the PFN for your Dev Center game below as it appears within the Dev Center UI. This value doesn't go live until the Service Configuration is published for the desired sandbox.

x

**Save**

## Dev Guidance

### 1. Configuring your development PC (Sandbox Setup)

#### 1) Xbox Live Account Tool

Login using [Dev account](#) and change sandbox. File Location: `{*SDK source root*}\Tools\XboxLiveAccountTool.exe`

#### 2) Verify Dev Environment

Try to login using Xbox App on Win10 PC using Dev account to check if the sandbox is already changed

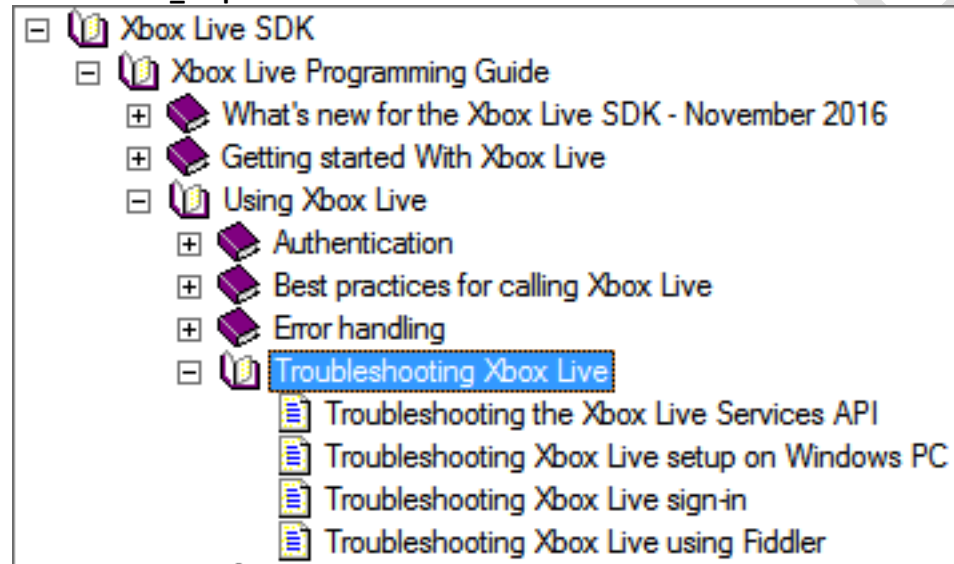
**If you can't login using dev account, please run this script from admin command prompt:**

`{*SDK source root*}\Tools\SwitchSandbox.cmd <your sandbox ID>`

Sample: `SwitchSandbox.cmd XDKS.1`

### 2. Troubleshooting issues when integrating Xbox Live

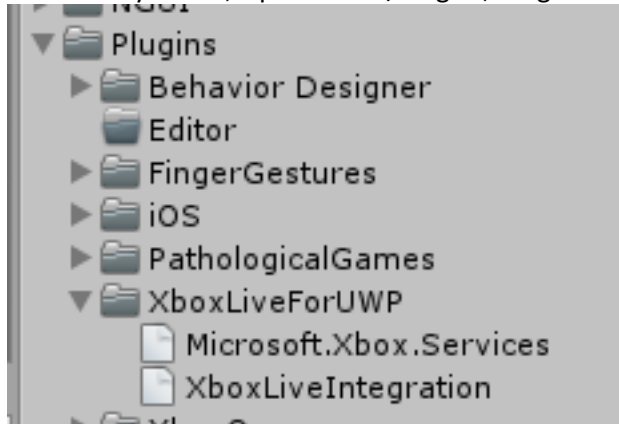
XboxLiveSDK\_uwp.chm:



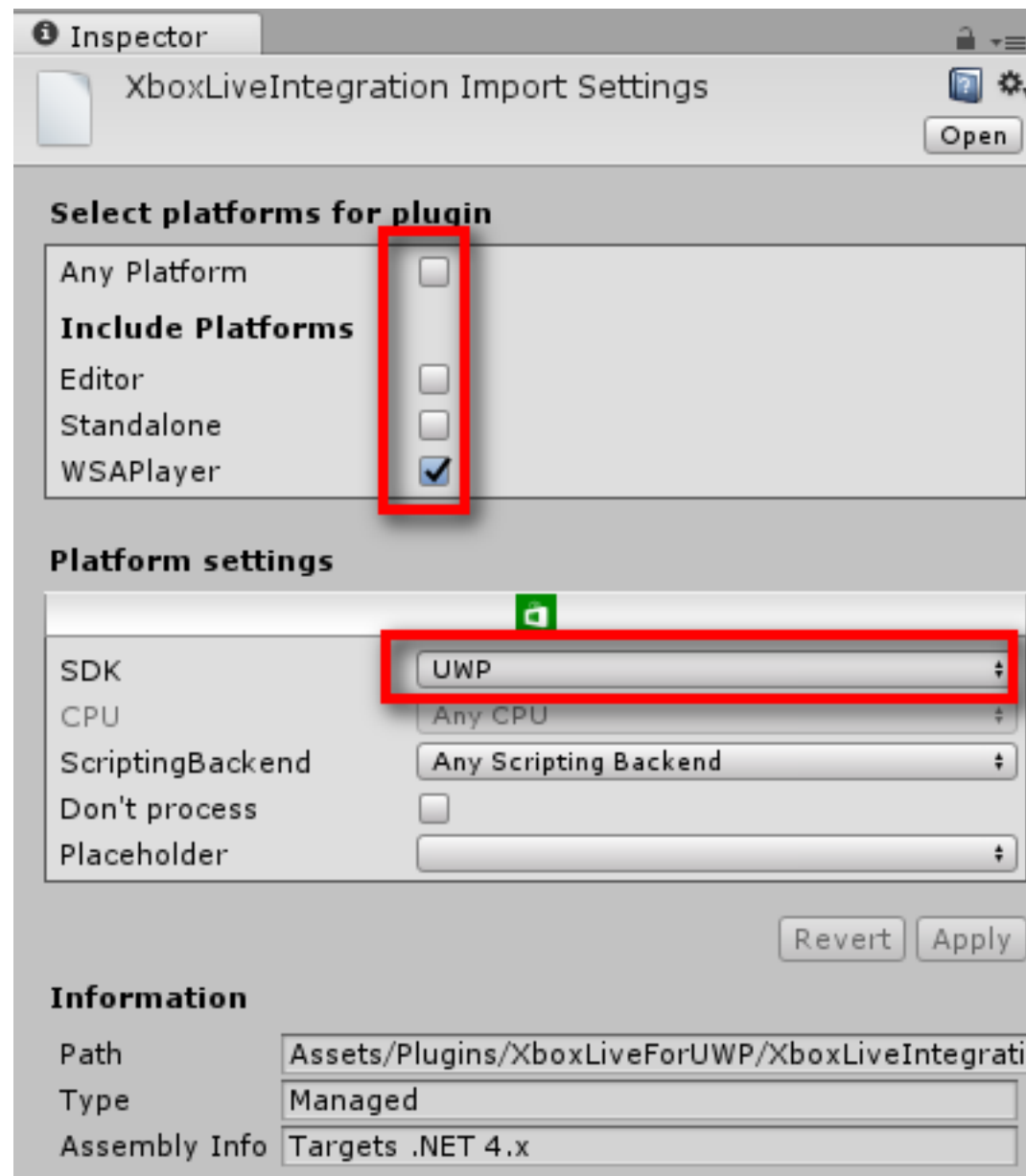
### 3. Make integration works in UWP app (Unity to UWP)

We can either use the standard API to implement the integration by following the API document, or utilize the **XboxLiveIntegration** class library in Unity project and convert to UWP project. Download the latest release in [here](#)

- 1) In the Unity Editor, Open Assets/Plugins, drag **Microsoft.Xbox.Services.winmd** and **XboxLiveIntegration.dll** into to Assets hierarchy

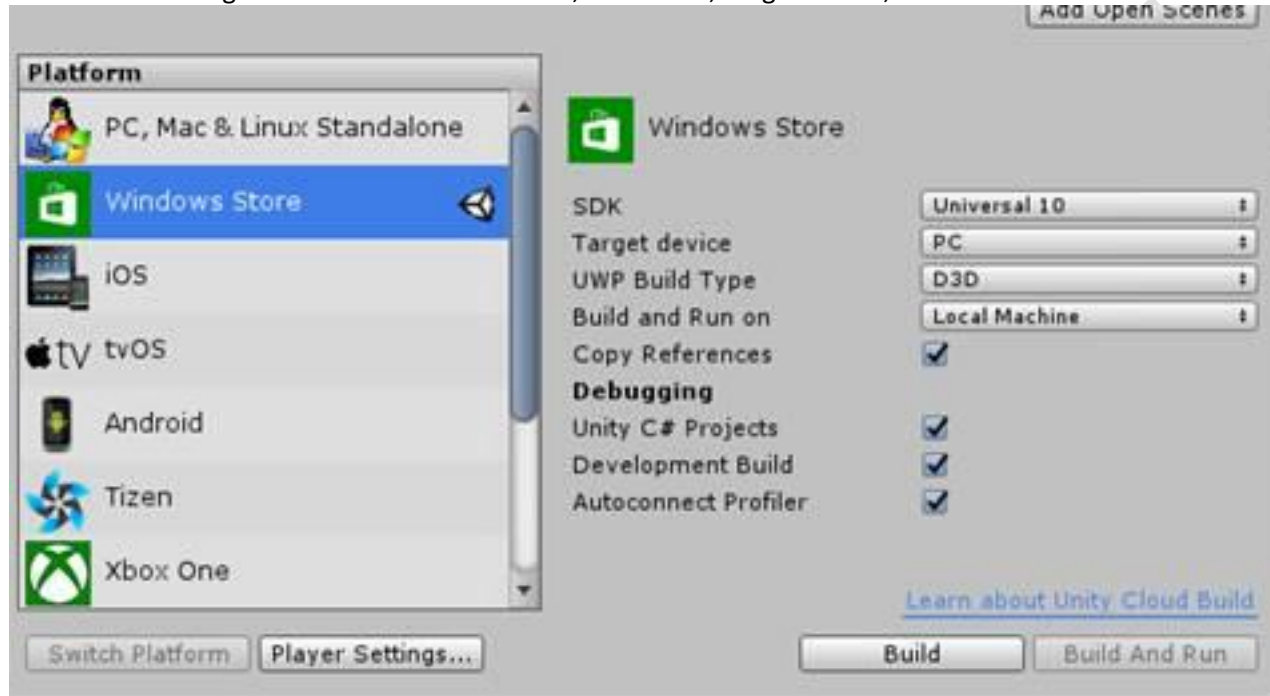


- 2) Change Importing settings for **XboxLiveIntegration.dll**. Select this item and change setting as follows

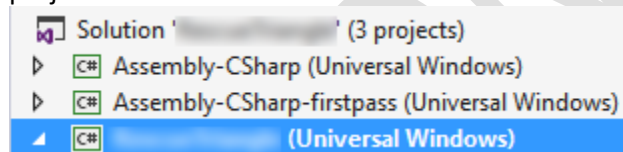




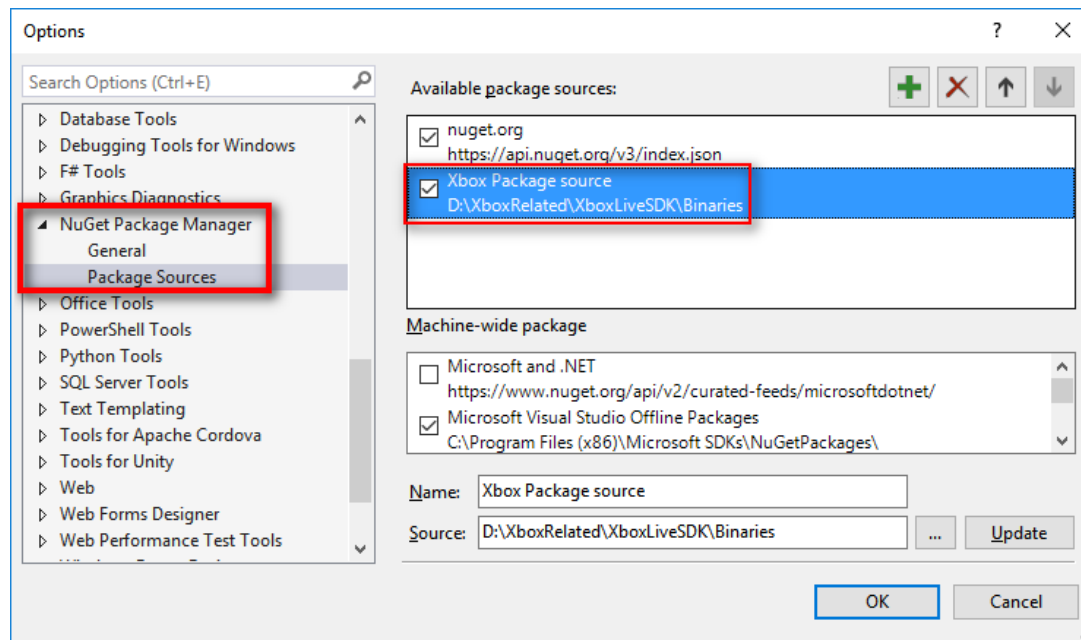
- 3) File -> Build settings->Choose “Windows Store”, select SDK, Target device, Build and Run on as follows:



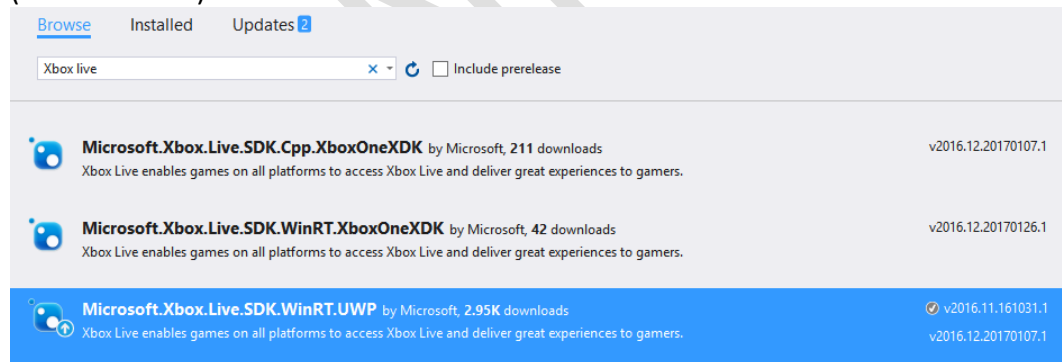
- 4) Open the converted UWP project in Visual Studio, the first two projects are auto-generated, we need to do some changes in the third UWP project



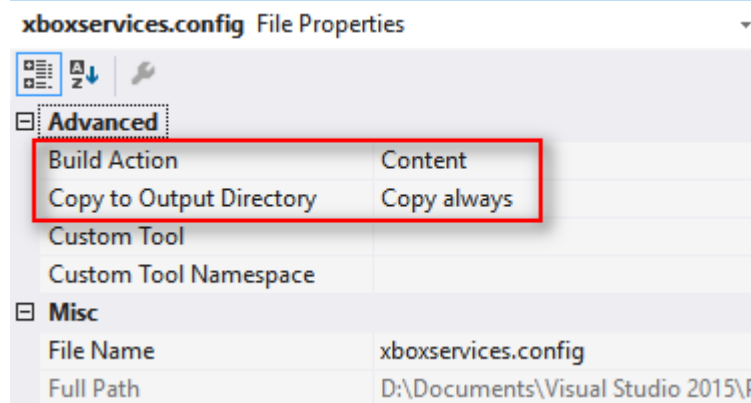
- 5) Install XBL Nuget package for your UWP project
- Method 1:** Create customized Nuget Source and point to the 'Xbox Live SDK ->Binaries' directory. Choose your source in the “Manage NuGet Packages” page and select the appropriate API set (C++ or WinRT)



- ii. **Method 2 (Recommended):** Search for “xbox live” in the “Manage NuGet Packages” page. Select the appropriate API set (C++ or WinRT) and then click on “Install”



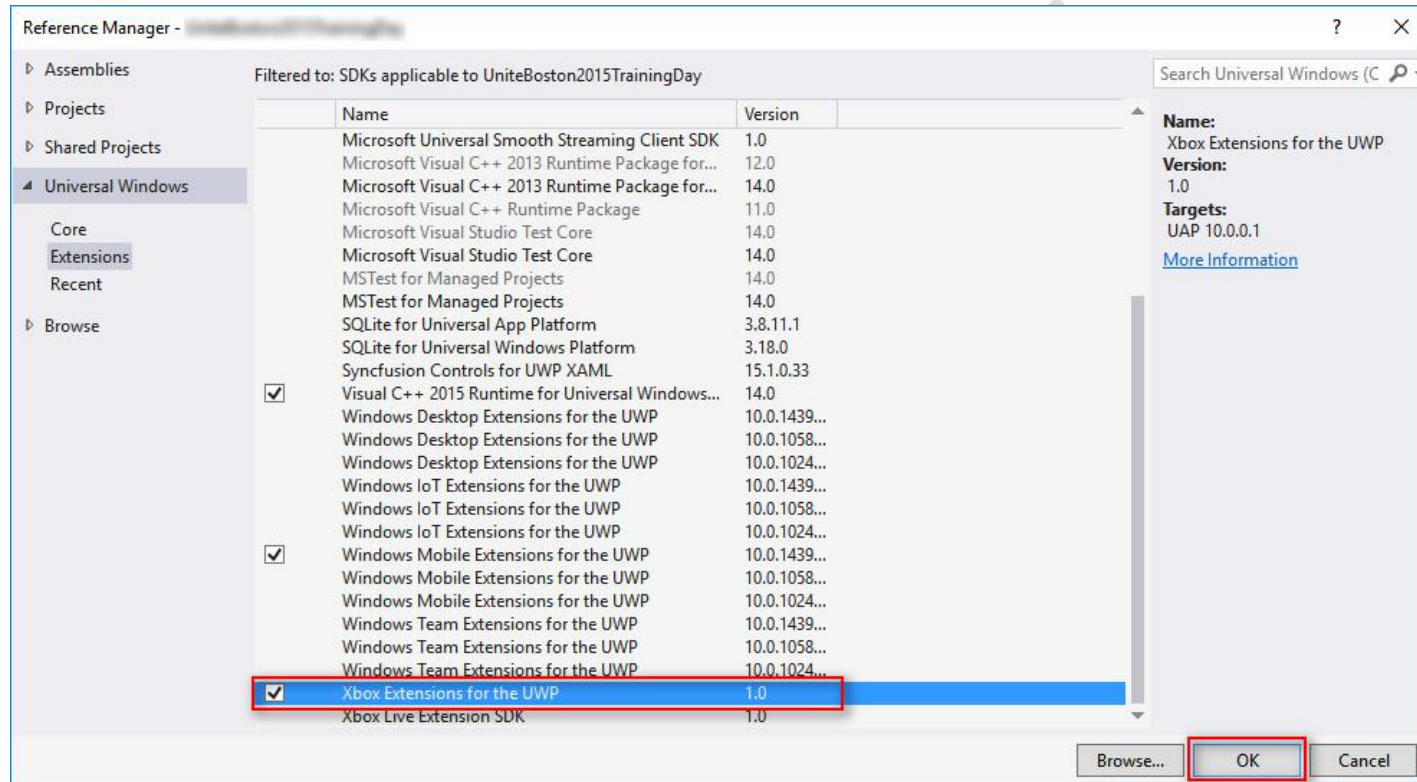
- 6) Associate your Visual Studio project with your Xbox Live enabled title information
  - i. Create a JSON file and name it **xboxservices.config**
  - ii. Add the JSON file to your primary UWP project (the StartUp Project)
  - iii. Right click on the file, select **Properties** and ensure that **Build Action** is set to **Content** and set **Copy always** for **Copy to Output Directory**. This will ensure the file is copied correctly in the AppX folder.



- iv. You can leave the Item Type to Does not participate in build
  - v. Edit the JSON file with the following template, and replace the [TitleId](#), [PrimaryServiceConfigId](#) with the values you get from Windows Dev Center(UDC)  

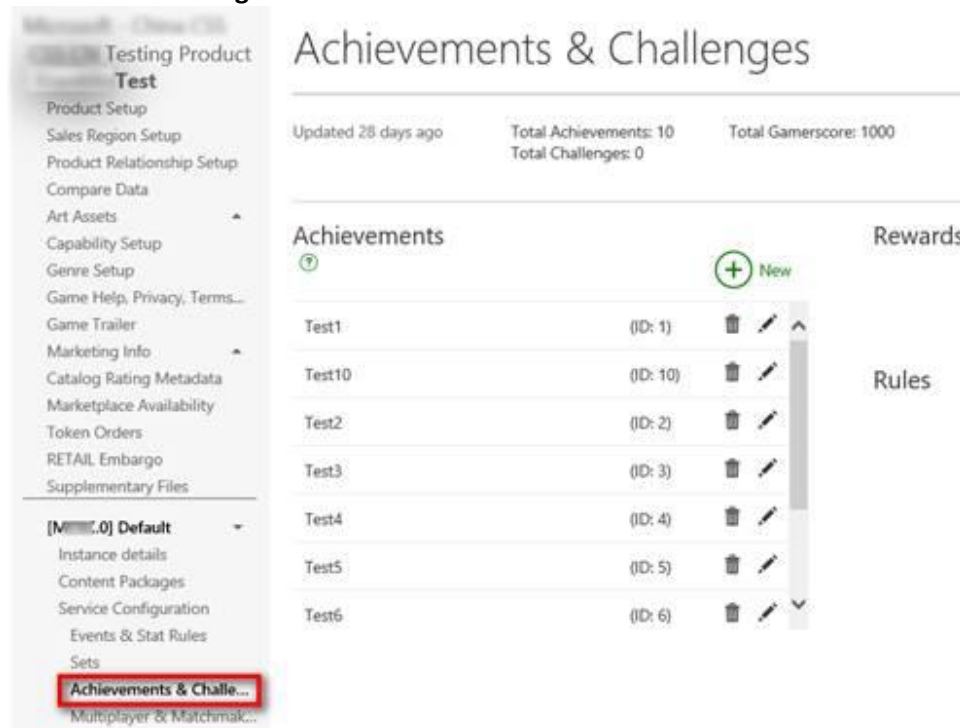
```
{"TitleId": xxxxxx, "PrimaryServiceConfigId": "exxxx-7xxx-4xxx-axxx-3609xxx" }
```
- 7) Associate publisher information with your UWP App
  - i. Open your project in Visual Studio 2015
  - ii. Right click the primary UWP project (the StartUp Project), click **Store** -> **Associate App with the Store...**
  - iii. Sign-in with the Windows Developer account used for creating the app if asked
  - iv. On the next page, select the app you just created, confirm the information, and click **Associate**
- 8) Enable **Internet (Client)** capability
  - i. Double click on the **package.appxmanifest** file in Visual Studio 2015 to open the Manifest Designer.
  - ii. Click on the **Capabilities** tab
  - iii. Click on **Internet (Client)**
  - iv. Close the file and save the changes.
- 9) Add **Xbox Extensions for the UWP**

- i. Right-Click project, Add -> Reference...
- ii. Universal Windows -> Extensions -> Select “Xbox Extensions for the UWP”



#### 4. Achievement configuration on XDP

##### 1) Achievement configuration in sandbox



Microsoft Xbox Live  
Testing Product  
Test

Product Setup  
Sales Region Setup  
Product Relationship Setup  
Compare Data  
Art Assets  
Capability Setup  
Genre Setup  
Game Help, Privacy, Terms...  
Game Trailer  
Marketing Info  
Catalog Rating Metadata  
Marketplace Availability  
Token Orders  
RETAIL Embargo  
Supplementary Files

[MCSC.0] Default  
Instance details  
Content Packages  
Service Configuration  
Events & Stat Rules  
Sets  
**Achievements & Challe...**  
Multiplayer & Matchmak...

## Achievements & Challenges

Updated 28 days ago    Total Achievements: 10    Total Gamerscore: 1000  
Total Challenges: 0

### Achievements

+ New

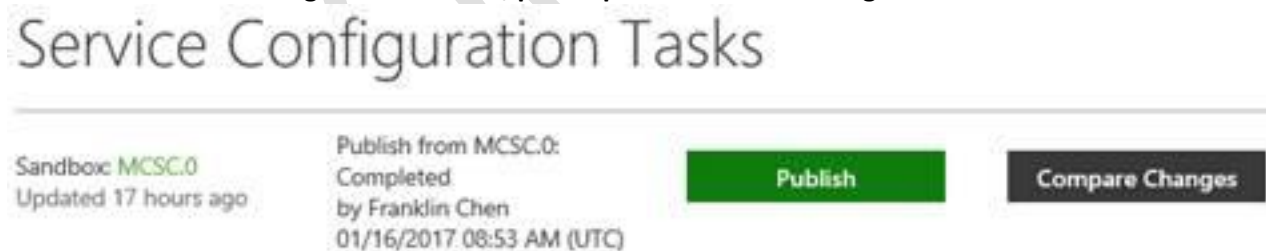
Test1	(ID: 1)	🗑️ ✎️ ⬆️
Test10	(ID: 10)	🗑️ ✎️ ⬆️
Test2	(ID: 2)	🗑️ ✎️ ⬆️
Test3	(ID: 3)	🗑️ ✎️ ⬆️
Test4	(ID: 4)	🗑️ ✎️ ⬆️
Test5	(ID: 5)	🗑️ ✎️ ⬆️
Test6	(ID: 6)	🗑️ ✎️ ⬆️

### Rewards

### Rules

Note: For one product, we must set at least **10** achievements with total **1000** Gamerscore reward

##### 2) After Achievement configuration on XDP, please publish to make changes work



## Service Configuration Tasks

Sandbox: **MCSC.0**  
Updated 17 hours ago

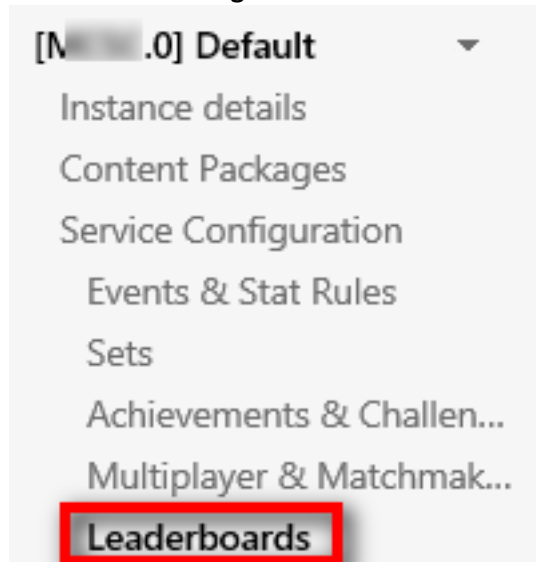
Publish from MCSC.0:  
Completed  
by Franklin Chen  
01/16/2017 08:53 AM (UTC)

**Publish**    **Compare Changes**

The changes will take effect about 10-30 minutes later

## 5. Leaderboard configuration on XDP

### 1) Leaderboard configuration in sandbox



### 2) After Achievement configuration on XDP, please publish to make changes work

The changes will take effect about 10-30 minutes later

## 6. The additional code work in converted UWP project as follows (If using [XboxLiveIntegration class library](#))

### 1) Initialize and Sign in Xbox Live

- i. When app start, please call  

```
//Init and Sign in silently  
XboxLiveIntegration.LiveResources.GetInstance();
```

**to initialize and sign in silently.**

- ii. Handle Keyboard/Gamepad/Joystick Input

```

//Handle Keyboard and Joystick input
if (Input.GetKeyDown("t") || Input.GetKeyDown("joystick button 2"))
{
    XboxLiveIntegration.LiveResource.GetInstance().SignIn();
}
if (Input.GetKeyDown("y") || Input.GetKeyDown("joystick button 3"))
{
    XboxLiveIntegration.LiveResource.GetInstance().SwitchAccount();
}

```

## 2) Using AchievementManager class to write game event

```

/// <summary>
/// Unlock First Win Achievement
/// </summary>
/// <param name="totalWinCount">TotalWinCount</param>
void UnlockFirstWinAchievement(int totalWinCount)
{
    var liveResource = XboxLiveIntegration.LiveResources.GetInstance();
    var measurements = new Windows.Foundation.Collections.PropertySet();
    measurements.Add("TotalWinCount", totalWinCount);

    var dimensions = new Windows.Foundation.Collections.PropertySet();
    dimensions.Add("UserId", liveResource.User.XboxUserId);

    AchievementManager.GetInstance().WriteGameEvent("TotalDataUpdate", dimensions, measurements);
}

#if NETFX_CORE
    var liveResource = XboxLiveIntegration.LiveResources.GetInstance();
    if (liveResource.IsSignedIn)
        this.UnlockFirstWinAchievement(1); //Pass total win count number, 1 as example
    else
        System.Diagnostics.Debug.WriteLine("Please sign in to Xbox Live first");
#endif

```

### 3) Using LeaderboardManager class to retrieve Leaderboard data

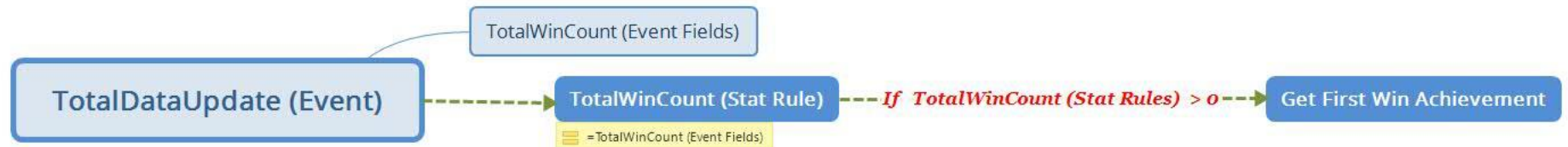
```
/// <summary>
/// Get Xbox Live Leaderboard by Leaderboard name
/// </summary>
/// <param name="name"></param>
async void GetLeaderboardByName(string name)
{
    var result = await LeaderboardManager.GetInstance().GetLeaderboardAsync(name);
}
```

## Scenario Sample

Goal: First Win achievement for one game character and leaderboard implementation

### Step 1: Achievement Design

Event->Stat Rules->Achievement



### Step 2: XDP Configuration

#### 1) Event & Stat Rules

Create new event and add event field



Add New Event

Base Event ?

Base

Use as event name

Event Name ?

TotalDataUpdate

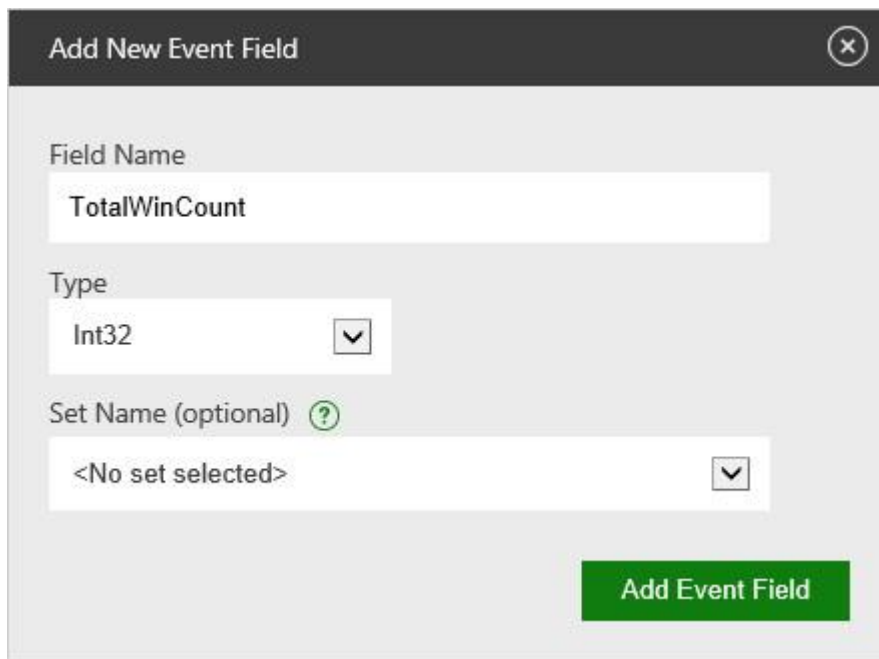
Description ?

Event Fields

+ Add Event Field ?

Field	Data Type	Set Name
UserId	Unicode...	
PlayerSessionId	GUID	

Add Event



**Add New Event Field**

Field Name  
TotalWinCount

Type  
Int32

Set Name (optional) ?  
<No set selected>

**Add Event Field**

Select the Event we just created and create Stat Rule

TotalDataUpdate: Add Stat Rule

Stat Rule Name ?☐ Open read access to any title ?  
TotalWinCount

Fields added to the stat rule name: ?  

+

 Add Event Field

Operator ?  
Replace

Parameter ?  
TotalWinCount

Fields added to the stat metadata: ?  


+

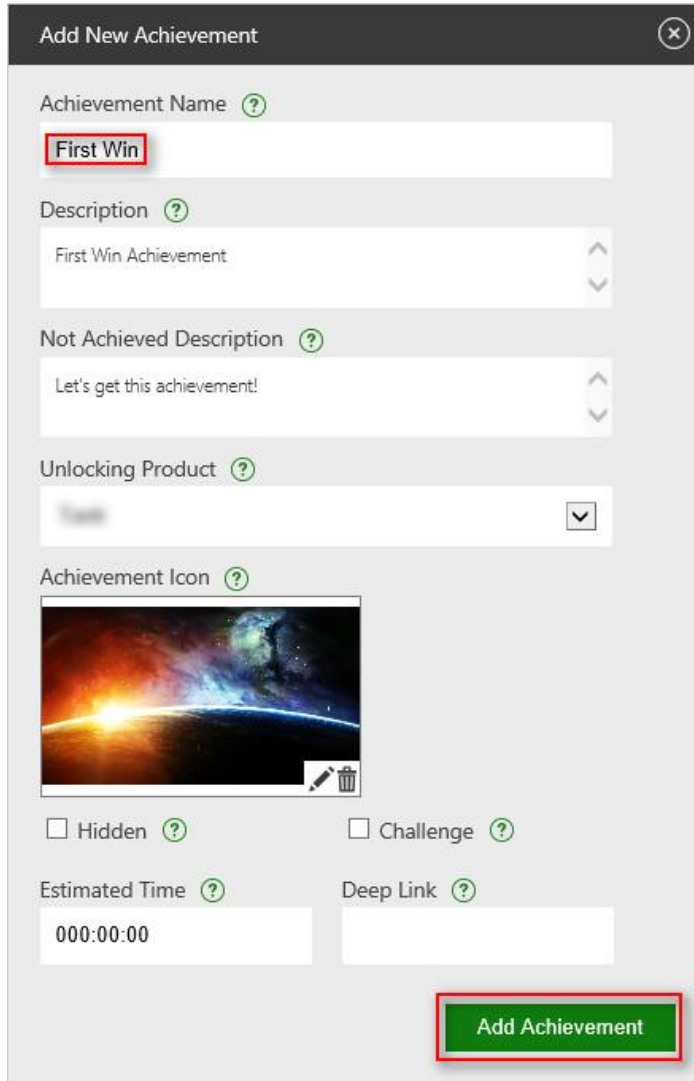
 Add Event Field


☐ Only allow changes to this stat from a server event. Client events will be ignored. ?


Add Stat Rule


## 2) Achievement


Check the tooltip  and document if you have difficult to fill the form  
Create new achievement










**Add New Achievement** 



Achievement Name   
**First Win**



Description   
First Win Achievement

Not Achieved Description   
Let's get this achievement!

Unlocking Product   
 

Achievement Icon   
  

☐ Hidden  ☐ Challenge 

Estimated Time  000:00:00 Deep Link 

**Add Achievement**

Set Reward and unlock rules, using the stat rule we created in [the last step](#)



Reward Type ?

Gamerscore

Value ?

200

Save Reward

Add New Rule

Rule Name ?

TotalWinCount

Type ?

Basic

Action ?

Max

Threshold ?

0

Data Type

Assigned after rule is saved

Add Stat Instance to Rule ?

Select an Instance

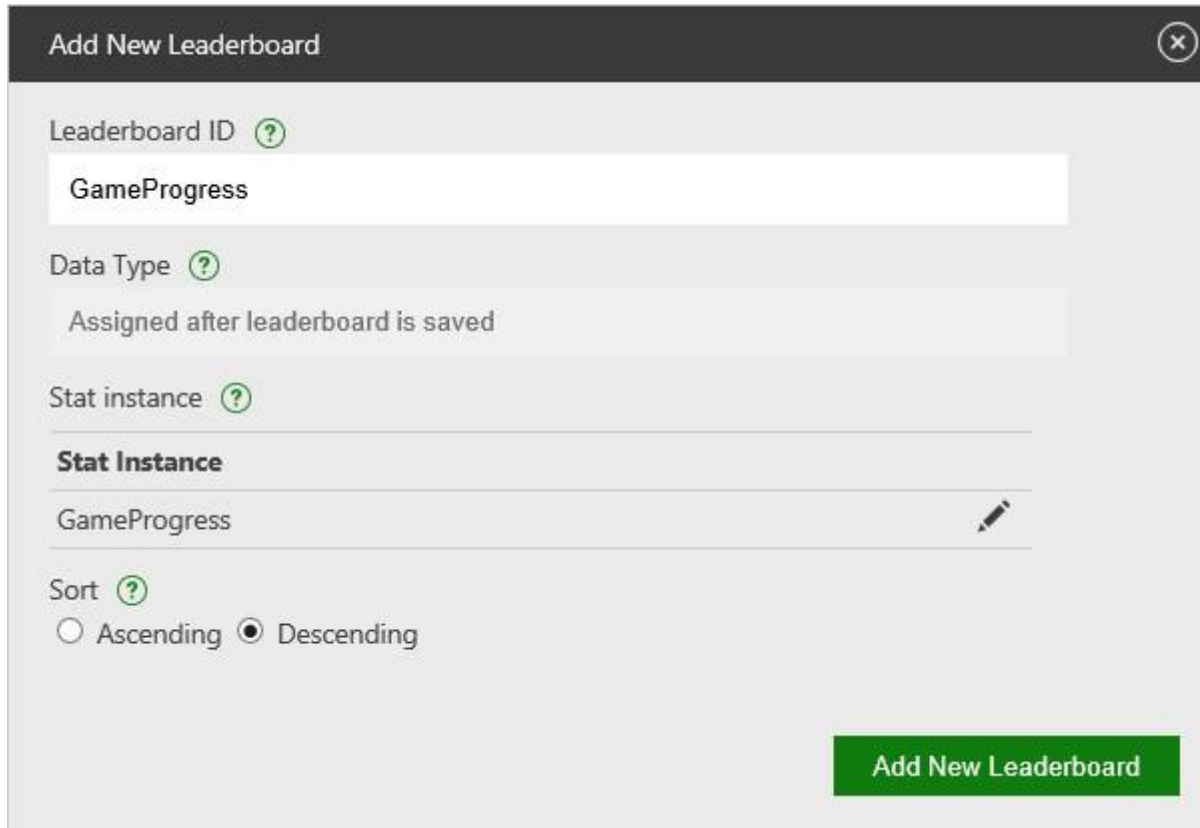
Stat Instance(s)

TotalWinCount

Add Rule

Note: For one product, we must set at least **10** achievements with total 1000 Gamerscore reward

### 3) Leaderboard



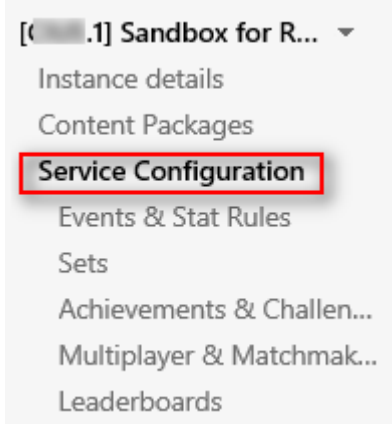
The screenshot shows a web form titled "Add New Leaderboard" with a close button (X) in the top right corner. The form contains the following fields and options:

- Leaderboard ID** (with a green question mark tooltip icon): A text input field containing the value "GameProgress".
- Data Type** (with a green question mark tooltip icon): A dropdown menu with the selected value "Assigned after leaderboard is saved".
- Stat instance** (with a green question mark tooltip icon): A section containing a table with one row. The table has a header "Stat Instance" and a value "GameProgress". A pencil icon is visible to the right of the value.
- Sort** (with a green question mark tooltip icon): Radio buttons for "Ascending" and "Descending", with "Descending" selected.

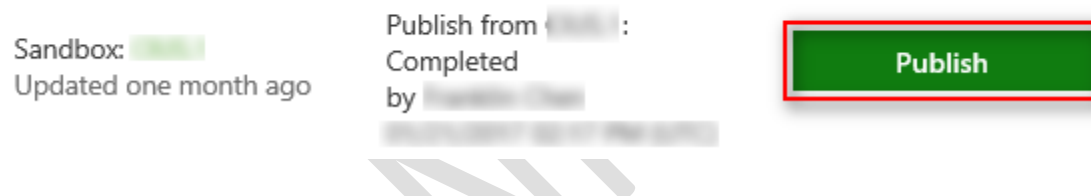
A green button labeled "Add New Leaderboard" is located at the bottom right of the form.

Check the tooltip  and document if you have difficulty filling the form

#### 4) Publish Service Configuration



## Service Configuration Tasks



## Publishing Approval: Product Instance [Product Name].1



If the configuration is passed the validation, the changes will take effect about **10-30 minutes later**



### Step 3: Development work

Please follow [Dev Guidance](#)

Here we will use [Tanks](#) project as example:

#### 1) Initialize and Sign in Xbox Live

Position: [\\_Completed-Assets\Scripts\Managers\GameManager.cs](#) -> [Start\(\)](#) method

When app start, please call

```
//Init and Sign in silently
```

```
XboxLiveIntegration.LiveResources.GetInstance();
```

to initialize and sign in silently.

#### 2) Handle Keyboard/Gamepad/Joystick Input

Position: [\\_Completed-Assets\Scripts\Managers\GameManager.cs](#) -> [Update\(\)](#) method

```
//Handle Keyboard and Joystick input
```

```
if (Input.GetKeyDown("t") || Input.GetKeyDown("joystick button 2"))
```

```
{
```

```
    XboxLiveIntegration.LiveResource.GetInstance().SignIn();
```

```
}
```

```
if (Input.GetKeyDown("y") || Input.GetKeyDown("joystick button 3"))
```

```
{
```

```
    XboxLiveIntegration.LiveResource.GetInstance().SwitchAccount();
```

```
}
```

#### 3) Using AchievementManager class to write game event

Position: [\\_Completed-Assets\Scripts\Managers\GameManager.cs](#)

```

/// <summary>
/// Unlock First Win Achievement
/// </summary>
/// <param name="totalWinCount">TotalWinCount</param>
void UnlockFirstWinAchievement(int totalWinCount)
{
    var liveResource = XboxLiveIntegration.LiveResources.GetInstance();
    var measurements = new Windows.Foundation.Collections.PropertySet();
    measurements.Add("TotalWinCount", totalWinCount);

    var dimensions = new Windows.Foundation.Collections.PropertySet();
    dimensions.Add("UserId", liveResource.User.XboxUserId);

    AchievementManager.GetInstance().WriteGameEvent("TotalDataUpdate", dimensions, measurements);
}

/// <summary>
/// Get Xbox Live Achievement by Id
/// </summary>
/// <param name="id"></param>
async void GetAchievementById(int id)
{
    var ach = await AchievementManager.GetInstance().GetAchievement(id.ToString());
    if (ach.ProgressState ==
Microsoft.Xbox.Services.Achievements.AchievementProgressState.Achieved)
    {
        var msg = $"You achieved {ach.Name}";
        System.Diagnostics.Debug.WriteLine(msg);
    }
}

IEnumerator ExecuteAfterTime(float time)
{
    yield return new WaitForSeconds(time);

    GetAchievementById(1);
}

```

Position: `_Completed-Assets\Scripts\Managers\GameManager.cs` -> `GetGameWinner ()` method:

```
private TankManager GetGameWinner()
{
    // Go through all the tanks...
    for (int i = 0; i < m_Tanks.Length; i++)
    {
        // ... and if one of them has enough rounds to win the game, return it.
        if (m_Tanks[i].m_Wins == m_NumRoundsToWin)
        {
            #if NETFX_CORE
                var liveResource = XboxLiveIntegration.LiveResources.GetInstance();
                if (liveResource.IsSignedIn)
                {
                    this.UnlockFirstWinAchievement(1);
                }
                else
                {
                    System.Diagnostics.Debug.WriteLine("Please sign in to Xbox Live first");
                }

                if (liveResource.IsSignedIn)
                {
                    StartCoroutine(ExecuteAfterTime(2));
                }
            #endif

            return m_Tanks[i];
        }
    }

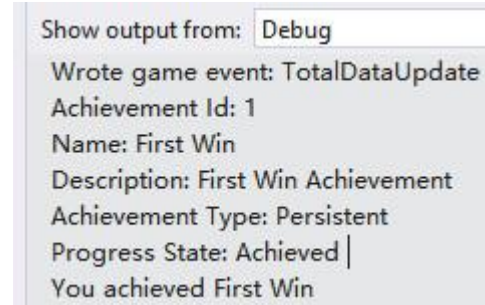
    // If no tanks have enough rounds to win, return null.
    return null;
}
```

#### 4) Using LeaderboardManager class to retrieve Leaderboard data

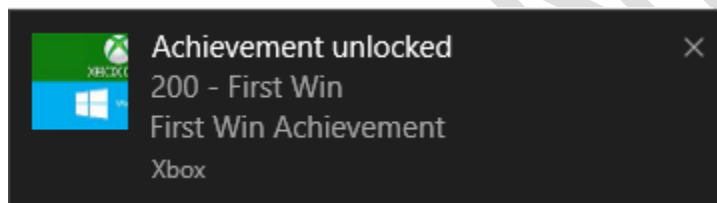
```
/// <summary>
/// Get Xbox Live Leaderboard by Leaderboard name
/// </summary>
/// <param name="name"></param>
async void GetLeaderboardByName(string name)
{
    var result = await LeaderboardManager.GetInstance().GetLeaderboardAsync(name);
}
```

#### Step 4: Debug and verify Achievement & Leaderboard data

Check the message in the Visual Studio output windows



The toast notification from Xbox UWP app:



If we get the Achievement, we should also can see the status in the built-in Xbox UWP app

