

Instrumenting Your Game

Using in-game analytics to track player engagement and find balance issues.



The instructions below are from an earlier version of Unreal Engine. iOS Apsalar is no longer available in UE5. However, these instructions should still be applicable to other analytics frameworks.

The first step in capturing player retention data is to register an analytics provider for your game. This is done via your project's `DefaultEngine.ini` file. You must register a default provider. Optionally, you can register different providers and account details for different build types of your game (development, testing, and production). The sections below are an example of configuring the Apsalar plugin:

```
1 [Analytics]
2 ProviderModuleName=IOSApsalar
3 ApiKey=YourAnalyticsKey1
4 ApiSecret=YourAnalyticsSecret1
5 SendInterval=60
6
7 [AnalyticsDevelopment]
8 ApiKey=YourAnalyticsKey2
```

```
9  ApiSecret=YourAnalyticsSecret2
10 SendInterval=60
11
12 [AnalyticsTest]
13 ApiKey=YourAnalyticsKey3
14 ApiSecret=YourAnalyticsSecret4
15 SendInterval=60
```

 Copy full snippet

The `[Analytics]` section is the default one used and is where you should set the name of the default provider module. In the case above, it is set to the IOSapsalar plugin that is part of the 4.5 release. The `ApiKey` and `ApiSecret` fields come from the Apsalar website. Once you create an account, they will give you a key and secret to use.

Once you have it configured for your project, you are ready to start recording analytics events. To get just the basic player retention data, you need to create a session when the game starts up and end it when it is no longer in the foreground. This can be done using the lines of code shown below, or the [Blueprint analytics plugin](#) provided in versions 4.5 and later:

```
1 FAnalytics::Get().GetDefaultConfiguredProvider()->StartSession();
2 FAnalytics::Get().GetDefaultConfiguredProvider()->EndSession();
```

 Copy full snippet

With those calls as part of your game, you will automatically start gathering player retention data. After getting basic player retention data, you can start adding more events to tell you even more about player behavior in your game.