



GOOGLE ANALYTICS

PLUGIN FOR UE4 SECOND EDITION



Contents

1. License	3
2. Plugin updates.....	4
a. Update from previous versions to 2.9.0	4
3. Introduction.....	5
4. Getting started	6
5. Example project	8
6. GitHub Repository.....	8
7. Recording events in Blueprints	9
8. Custom dimensions and metrics	12
9. Supported Analytics Provider Nodes	13
10. Recording events in C++	14
11. GDPR Compliance	15

1. License

Copyright © 2018 gameDNA Ltd. All rights reserved.

gameDNA Ltd grants you a non-exclusive, non-transferable, non-sublicensable license for a single User to use, reproduce, display, perform, and modify the Google Analytics Plugin for Unreal® Engine 4 for any lawful purpose (the "License"). The License becomes effective on the date you buy Google Analytics Plugin for Unreal® Engine 4. The License does not grant you any title or ownership in the Licensed Technology.

You may Distribute the Google Analytics Plugin for Unreal® Engine 4 incorporated in object code format only as an inseparable part of a Product to end users. The Product may not contain any Plugin Content in uncooked source format.

Unreal is a trademark or registered trademark of Epic Games, Inc. in the United States of America and elsewhere. Unreal® Engine, Copyright 1998 – 2017, Epic Games, Inc. All rights reserved.

Google Analytics is a trademark of Google, Inc.

2. Plugin updates

a. Update from previous versions to 2.9.0

Delete all files under *Plugins/GoogleAnalytics* directory and install plugin again as described in [Getting started](#) section.

3. Introduction

Google Analytics Provider integrates out-of-the-box Google Analytics into Unreal Engine project and is compatible with all platforms. It fully implements the Unreal Analytics Provider system so you can use all available features exposed by this, including Analytics Blueprint Library. It's useful for tracking events and screens, real-time sessions tracking, registering in-app purchases, user progression, automatic reporting errors, exceptions, crashes, etc.

Required Unreal Engine 4.10 or above because of Android Plugin support!

Current plugin version: 2.9.1

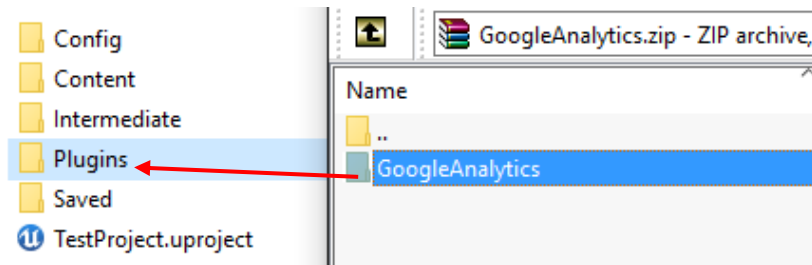
Support: support@gamednastudio.com

FEATURES:

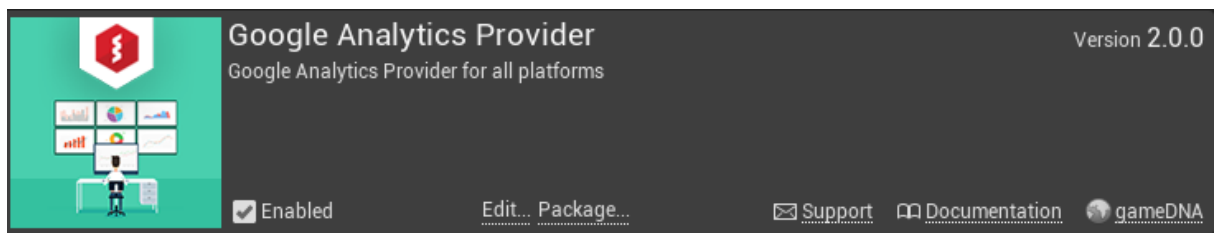
- Four special nodes only for Google Analytics Provider plugin: *Record Google Event*, *Record Google Screen*, *Record Google Social Interaction*, *Record Google User Timing*
- Implements Unreal Analytics Provider system
- All features are exposed to Blueprints
- Real-time sessions tracking (you can see who is currently playing your game!)
- Tracking events and screens
- Registering In-App Purchases
- Registering user progression
- Sending info about age, gender, location and interests
- Automatic errors, exceptions and crash reporting
- Automatic system info tracking (e. g. device, operating system, app version)
- Automatic caching events when user is offline and send them to Google Analytics when user is online
- Campaign measurement
- Social interactions
- User timing
- Custom dimensions and metrics
- And more features provided by Google Analytics
- Full source code
- Compatible with all UE4 platforms (all-in-one)
- Supports gameDNA installer. No more downloading SDKs and creating packages on your own!
- Easy plugin setup
- All future updates
- Support via e-mail (support@gamednastudio.com)

4. Getting started

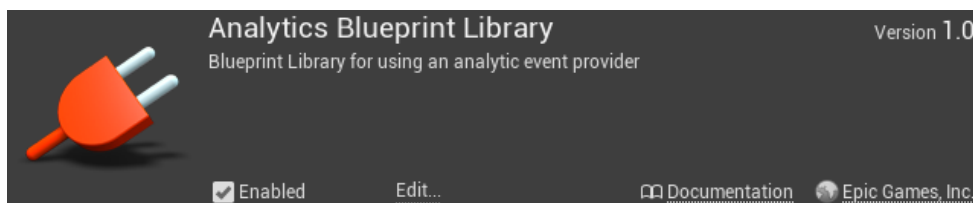
1. Unpack the plugin archive to *Plugins* folder in your UE4 project folder (for project plugins) or *Engine/Plugins/Marketplace* (for engine plugins) and start the editor.



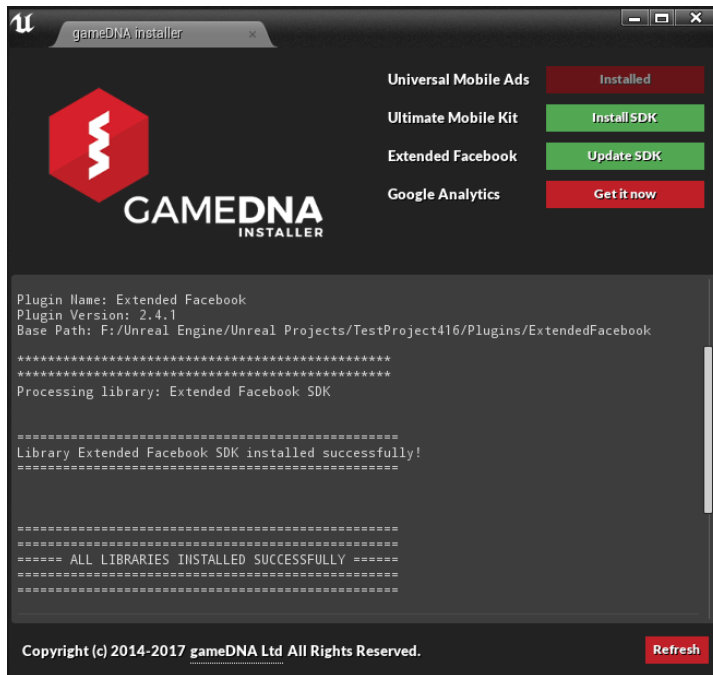
2. Enable *Google Analytics Provider* in *Edit -> Plugins -> Analytics -> Project -> Google Analytics Provider*.



3. If you want to use Blueprint nodes for *Google Analytics Provider*, go to *Edit -> Plugins -> Built-In -> Analytics* and enable *Analytics Blueprint Library*.



4. Download **gameDNA installer** plugin from the *releases* tab on the official GitHub repository <https://github.com/gameDNAstudio/gameDNAinstaller>. Install plugin and SDKs as described in the included documentation.



5. Open *Config/DefaultEngine.ini* file and add following lines:

```
[Analytics]
ProviderModuleName=GoogleAnalytics
TrackingIdAndroid=<ENTER YOUR TRACKING ID FOR ANDROID>
TrackingIdIOS=<ENTER YOUR TRACKING ID FOR IOS>
TrackingIdUniversal=<ENTER YOUR TRACKING ID FOR PLATFORMS OTHER THAN
IOS OR ANDROID>

[AnalyticsDevelopment]
ProviderModuleName=GoogleAnalytics
TrackingIdAndroid=<ENTER YOUR TRACKING ID FOR ANDROID>
TrackingIdIOS=<ENTER YOUR TRACKING ID FOR IOS>
TrackingIdUniversal=<ENTER YOUR TRACKING ID FOR PLATFORMS OTHER THAN
IOS OR ANDROID>

[AnalyticsTest]
ProviderModuleName=GoogleAnalytics
TrackingIdAndroid=<ENTER YOUR TRACKING ID FOR ANDROID>
TrackingIdIOS=<ENTER YOUR TRACKING ID FOR IOS>
TrackingIdUniversal=<ENTER YOUR TRACKING ID FOR PLATFORMS OTHER THAN
IOS OR ANDROID>
```

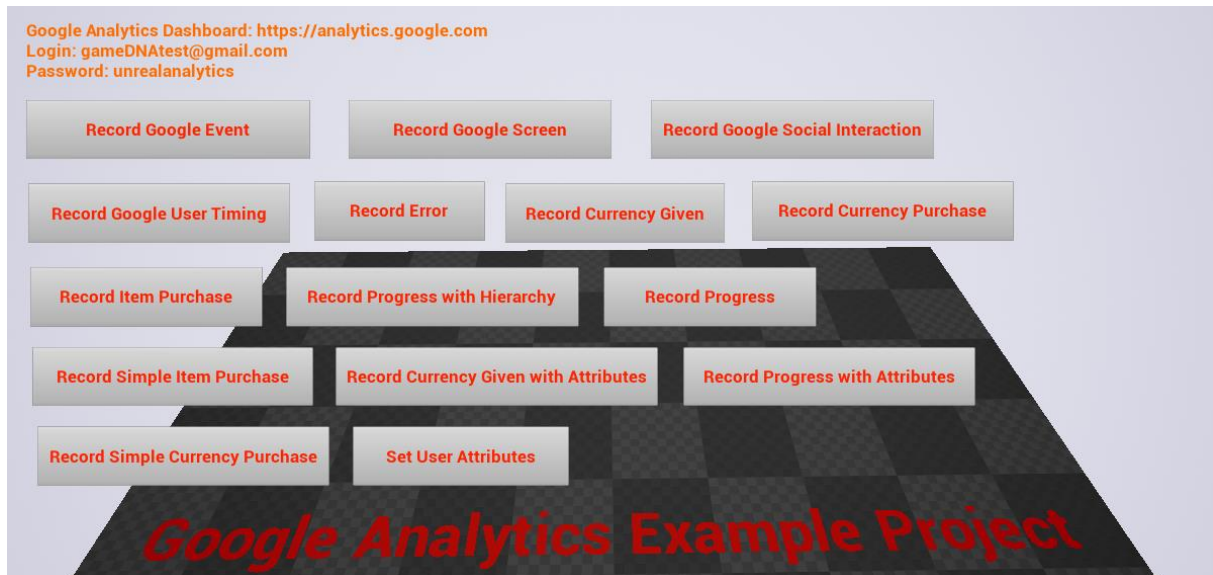
TIP: Google Analytics account for platforms other than iOS & Android should be set as website.

6. That's all! Google Analytics Provider works now! 😊

5. Example project

You can download example project at the following address:

<https://github.com/gameDNastudio/ExampleProjects>



Credentials to Google Analytics Dashboard (<https://analytics.google.com>):

Login: gameDNAtest@gmail.com

Password: unrealanalytics2017

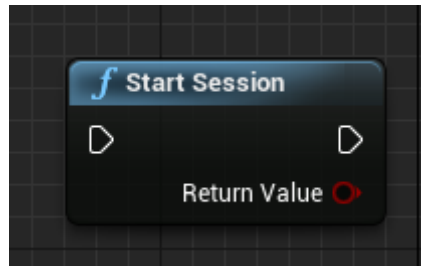
Please don't change password! ;)

6. GitHub Repository

Please email the receipt of this plugin's purchase to support@gamednastudio.com for access to the GitHub repository used to develop this plugin!

7. Recording events in Blueprints

Before recording any events you have to call the *Start Session* function. You should do it before any other Google Analytics function.

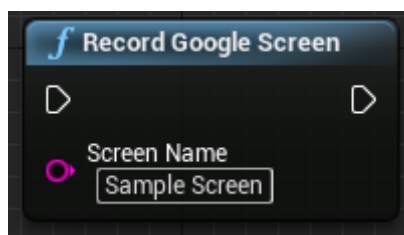


This plugin implements four special nodes only for Google Analytics.

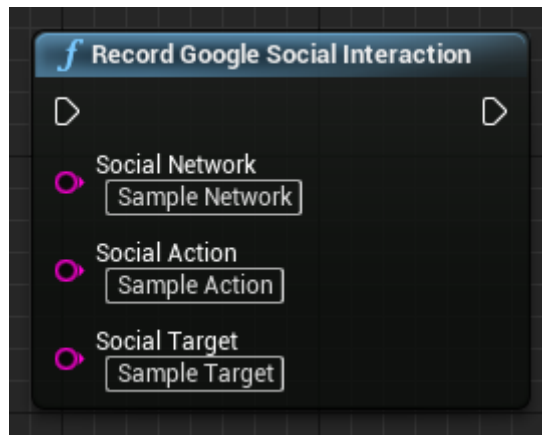
- a) **Record Google Event** - events are a useful way to collect data about a user's interaction with interactive components of your game, like button presses or the use of a particular item in a game.



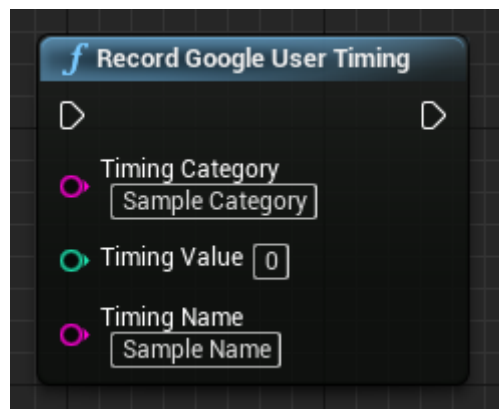
- b) **Record Google Screen** - screens in Google Analytics represent content users are viewing within your game. The equivalent concept in web analytics is a pageview. Measuring screen views allows you to see which content is being viewed most by your users, and how they are navigating between different pieces of content. A screen view consists of a single string field that will be used as the screen name in your Google Analytics reports.



- c) **Record Google Social Interaction** – social interaction measurement allows you to measure a user's interactions with various social network sharing and recommendation widgets embedded in your content.



- d) **Record Google User Timing** – measuring user timings provides a native way to measure a period of time in Google Analytics. This can be useful to measure resource load times, for example.



Google Analytics Provider plugin fully implements Unreal Analytics Provider system so you can use all available features exposed by this (including Analytics Blueprint Library).

You can learn more about Unreal Analytics Provider here:

<https://docs.unrealengine.com/latest/INT/Gameplay/Analytics/index.html>

and about Analytics Blueprint Library here:

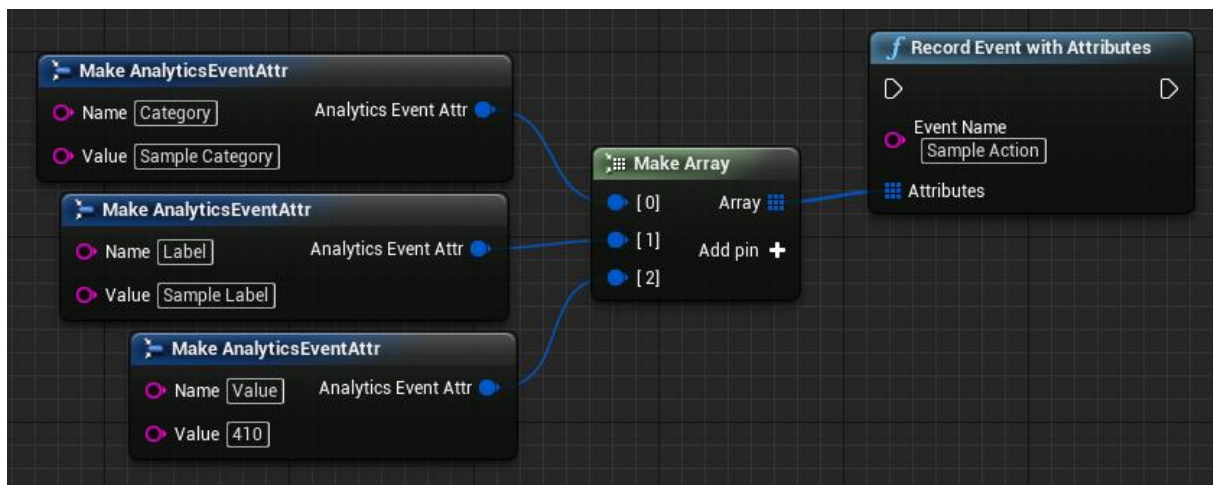
<https://docs.unrealengine.com/latest/INT/Gameplay/Analytics/Blueprints/index.html>

LIMITATIONS:

Google Analytics supports only following attributes for events:

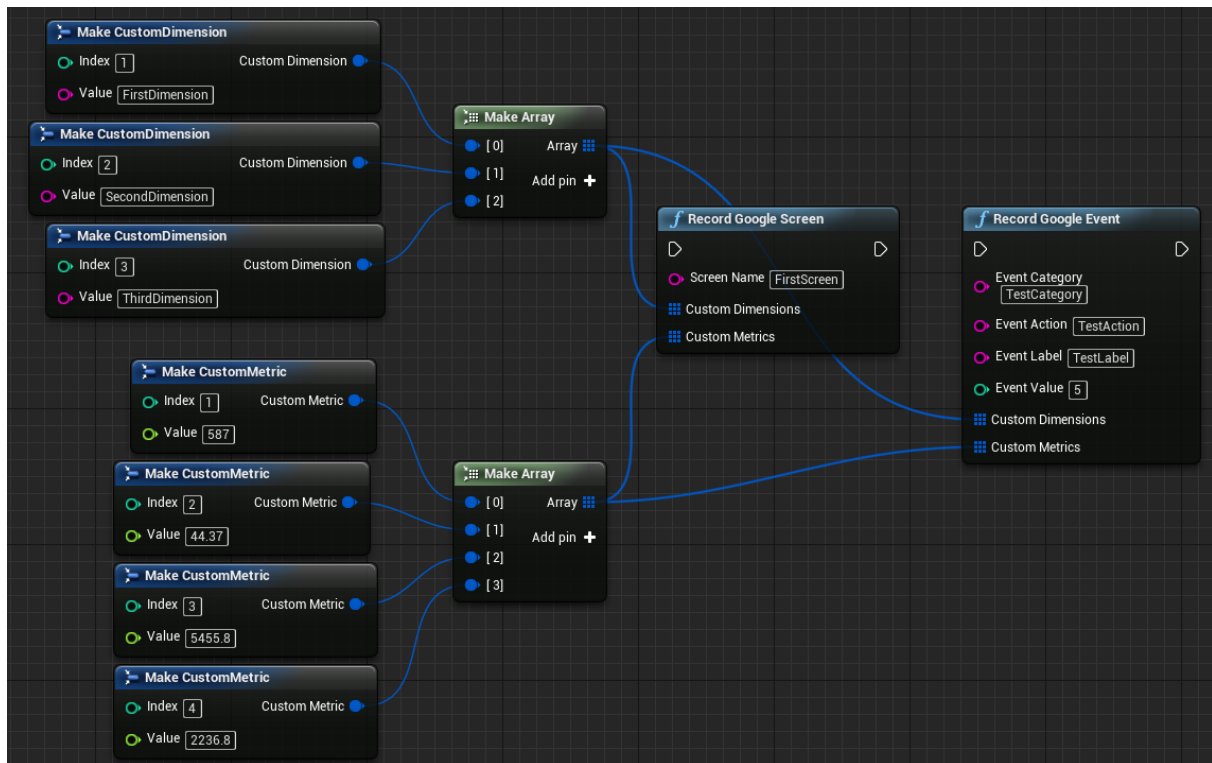
- Event Category (String)
- Event Action (String) – known as Event Name in Unreal Analytics Provider system
- Event Label (String)
- Event Value (Integer)

So if you are using standard Unreal Analytics Provider functions, you can make something like this and that's exactly what special Record Google Event node is.



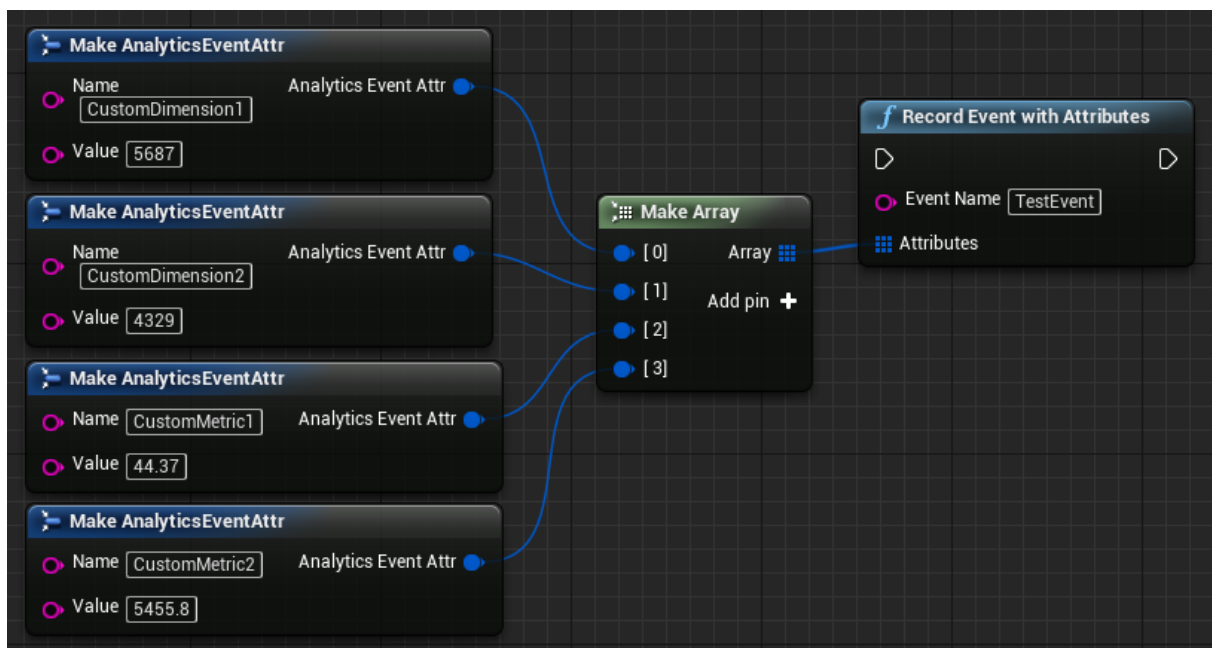
8. Custom dimensions and metrics

You can add custom dimensions and metrics for almost every function. Simply make an array of *CustomDimension* or *CustomMetric* structure and connect to the node.



You can also create array of *AnalyticsEventAttr* with specified:

- *CustomDimension*<DimensionIndex> (for Custom Dimensions)
- *CustomMetric*<MetricIndex> (for Custom Metrics)



9. Supported Analytics Provider Nodes

SUPPORTED ANALYTICS PROVIDER NODES

f Record Google Event
▷ ◁
Event Category
Event Action
Event Label
Event Value

f Record Google Social Interaction
▷ ◁
Social Network
Social Action
Social Target

f Record Google User Timing
▷ ◁
Timing Category
Timing Value
Timing Name

f Record Google Screen
▷ ◁
Screen Name

SPECIAL FOR GOOGLE ANALYTICS PLUGIN

f Start Session
▷ ◁
Return Value

f Get User Id
▷ ◁
Return Value

f Set User Id
▷ ◁
User Id

f Set Age
▷ ◁
Age

f Set Gender
▷ ◁
Gender

f Start Session with Attributes
▷ ◁
Attributes Return Value

f Make Event Attribute
Attribute Name Return Value
Attribute Value

f Set Location
▷ ◁
Location

f Record Error
▷ ◁
Error

f Record Currency Given
▷ ◁
Game Currency Type
Game Currency Amount

f Record Currency Purchase
▷ ◁
Game Currency Type
Game Currency Amount
Real Currency Type
Real Money Cost
Payment Provider

f Record Item Purchase
▷ ◁
Item Id
Currency
Per Item Cost
Item Quantity

f End Session
▷ ◁

f Record Event
▷ ◁
Event Name

f Record Progress with Full Hierarchy and Attributes
▷ ◁
Progress Type
Progress Names
Attributes

f Record Simple Item Purchase
▷ ◁
Item Id
Item Quantity

f Record Progress
▷ ◁
Progress Type
Progress Name

f Record Currency Given with Attributes
▷ ◁
Game Currency Type
Game Currency Amount
Attributes

f Record Progress with Attributes
▷ ◁
Progress Type
Progress Name
Attributes

f Record Simple Currency Purchase
▷ ◁
Game Currency Type
Game Currency Amount

f Record Event with Attribute
▷ ◁
Event Name
Attribute Name
Attribute Value

f Record Event with Attributes
▷ ◁
Event Name
Attributes

10. Recording events in C++

You can use Unreal Analytics Provider for C++ or simply call special functions for Google Analytics Provider plugin.

Add the following to your **.Build.cs* file private dependency module name:

```
PrivateDependencyModuleNames.AddRange(new string[] { "GoogleAnalytics" });
```

Import these headers anywhere in source code project:

```
#include "GoogleAnalyticsBlueprintLibrary.h"  
#include "Runtime/Analytics/Analytics/Public/Analytics.h"  
#include  
"Runtime/Analytics/Analytics/Public/Interfaces/IAalyticsProvider.h"
```

a) Start session:

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

b) Record Google Event:

```
UGoogleAnalyticsBlueprintLibrary::RecordGoogleEvent(TEXT("Event Category"),  
TEXT("Event Action"), TEXT("Event Label"), <INTEGER VALUE>);
```

c) Record Google Social Interaction:

```
UGoogleAnalyticsBlueprintLibrary::RecordGoogleSocialInteraction(TEXT("Social  
Network"), TEXT("Social Action"), TEXT("Social Target"));
```

d) Record Google User Timing:

```
UGoogleAnalyticsBlueprintLibrary::RecordGoogleUserTiming(TEXT("Timing  
Category"), <INTEGER VALUE>, TEXT("Timing Name"));
```

e) Record Google Screen:

```
UGoogleAnalyticsBlueprintLibrary::RecordGoogleScreen(TEXT("Screen Name"));
```

11. GDPR Compliance

Under the Google EU User Consent Policy, you must make certain disclosures to your users in the European Economic Area (EEA) and obtain their consent to use cookies or other local storage, where legally required, and to use personal data. This policy reflects the requirements of the EU ePrivacy Directive and the General Data Protection Regulation (GDPR).

The default behavior of the Google Analytics Provider is to track all user data. If a user has consented to not tracking IP address you can change this by calling *Set Anonymize IP* function.

