# Blueprint Analytics Plugin

A set of Blueprint nodes for communicating with analytics services.

The analytics APIs in the engine were designed to have no UObject dependencies. Because of that, you need a translation layer from Blueprints to the C++ code. We supply this translation layer as a Blueprint library plugin. Once you enable this plugin, you can start making analytics API calls from your Blueprints. The plugin translates the calls and sends them to the default analytics provider that you have registered for your project. If no default provider is registered, you will see a warning in your project's log files.
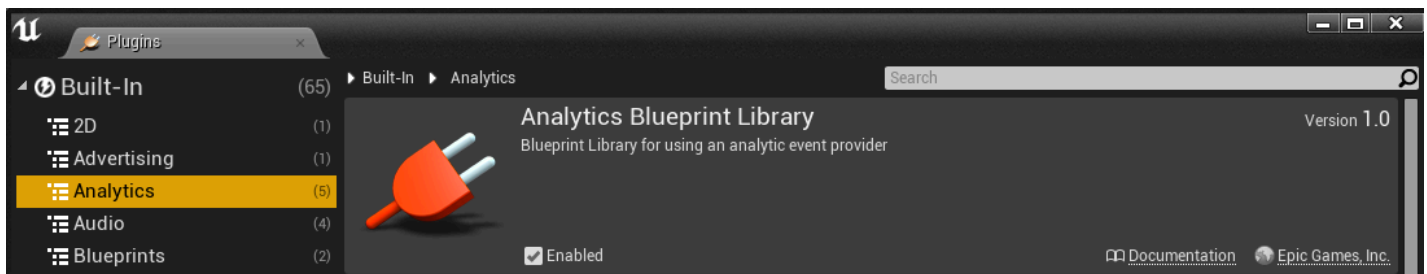
Start Session

This starts a new analytics session for the current player. Each play session should be treated as a unique session, so that your analytics data captures how frequently a user plays your

game and for how long. If a previous session is underway, most providers will automatically close the previous one and start a new one.

Output	Description
Return Value	Whether the default analytics provider was able to create a new session or not.

End Session

This stops recording events for the given session. Some providers will flush the data to their collector.

Record Event

To record an event that has no associated attributes with it, use the **Record Event** node. These types of events are often used to count how many times a user performs a specific option.

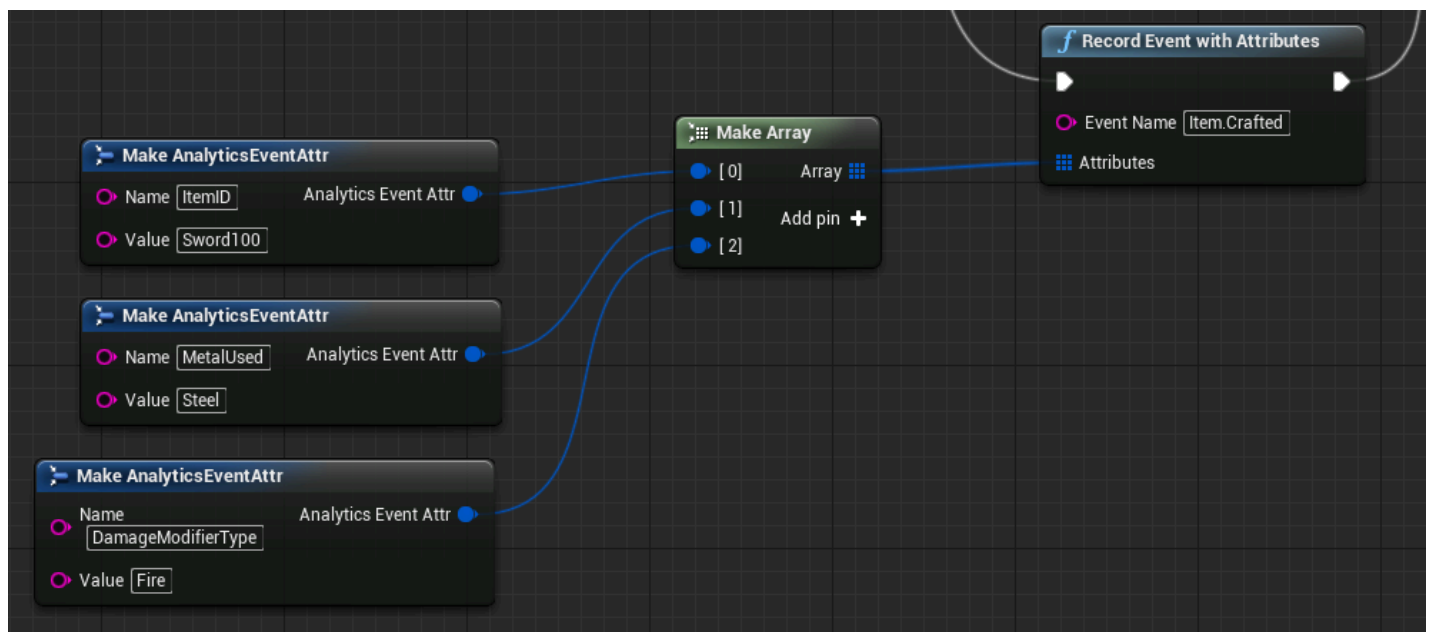| Inputs	Description
Event Name	The name of the event that is being incremented.

Record Event With Attribute

This records an event by name with a single attribute, so that you can compare the frequency of that event across the unique attributes. For instance, the Item.Equipped event below would record how often the user equips a weapon in your game, which will show which items are the most popular with players in your game.

Inputs	Description
Event Name	The name of the event that is being incremented.

Inputs	Description
Attribute Name	The name of the attribute that is customizing this event.
Attribute Value	The value that is used as the unique portion of the customization of the event.

# Record Event With Attributes

To provide even more details about your event, you can use **Record Event with Attributes**. This is used when you want to provide multiple attributes as context for the event. In the example below, the attributes describe the item that was crafted, the material used, and the player's preferred damage modifier. This example can show you player preferences for crafting items and also uncover potential balancing issues.

Inputs	Description
Event Name	The name of the event that is being incremented.
Attributes	The list of attributes that provide more context on the user's action

Make AnalyticsEventAttr

This is an easy way to take two strings and turn them into the structure that the analytics API needs.
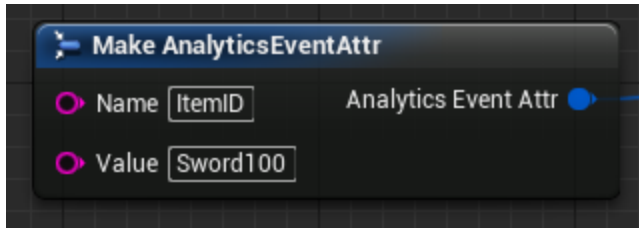
Inputs	Description
Attribute Name	The name of the attribute being created.
Attribute Value	The value of the attribute being created.

Flush Events

Some providers cache collected analytics data locally until a threshold is met or until a session ends. This call tells the provider to commit whatever data it has cached locally.

Record Item Purchase

This records the purchase of an in-game item using in-game currency. This is not for real world money transactions. Instead it tracks which items are purchased using your in-game currencies. This in-game currency can be earned over time or be purchased directly using real world money. To track real world money purchases of in-game currency, use the Record Currency Purchase node.

Inputs	Description
Item Id	This is a unique item descriptor to attribute the purchase to. This ID is used when viewing reports about how popular particular items are.

Inputs	Description
Currency	This is the string that describes which in-game currency was used to purchase the item (gold, chips, gems, etc.)
Per Item Cost	Indicates how much the item that was purchased costs in the in-game currency.
Item Quantity	The number of items that were purchased by the player.

Record Currency Given

This records when the game grants the player some in-game currency. This can be for completing a quest, selling an item, opening a chest, etc. It is important to record this data, so that you can understand how your in-game economy is performing overall.

Inputs	Description
Game Currency Type	This is the string that describes which in-game currency was used awarded to the player (gold, chips, gems, etc.)
Game Currency Amount	The amount of the currency that was granted to the player.

Record Currency Purchase

This records an in-game currency purchase paid for by real world money, a.k.a. an in-app purchase. Some providers will capture this information automatically, e.g. Apsalar. Usually, this behavior can be configured on/off, depending on whether you want to manually capture the information or not.

Inputs	Description
Game Currency Type	This is the string that describes which in-game currency was purchased by the player (gold, chips, gems, etc.)

Inputs	Description
Game Currency Amount	The amount of the currency that was granted to the player.
Real Currency Type	This should be the ISO 4217 3 letter abbreviation for the real world currency used to purchase the in-game currency. See the [ISO guides](#) for more information.
Real Money Cost	The amount of real world money used to purchase the in-game currency.
Payment Provider	The provider that fulfilled the purchase (Apple, Google, PayPal, Steam, etc.)