Macros

Collapsed networks of nodes that can be executed, or called, from another graph.

Blueprint Macros, or **Macros**, are essentially the same as collapsed graphs of nodes. They have an entry point and exit point designated by tunnel nodes. Each tunnel can have any number of execution or data pins which are visible on the macro node when used in other Blueprints and graphs.

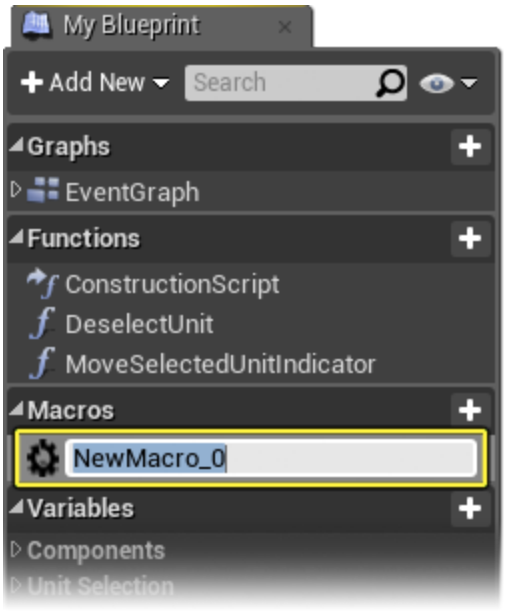
Creating a Blueprint Macro

Blueprint Macros can be created within a Blueprint Class or Level Blueprint, just like Blueprint Functions. They can also be organized into Blueprint Macro Libraries.

To create a Blueprint Macro inside a Blueprint Class, Level Blueprint, or Blueprint Macro Library:

1. In the **My Blueprint** tab create a new macro, by clicking on the **Add Button** on the macros list header 

.
2. Enter a name for your Blueprint Macro.



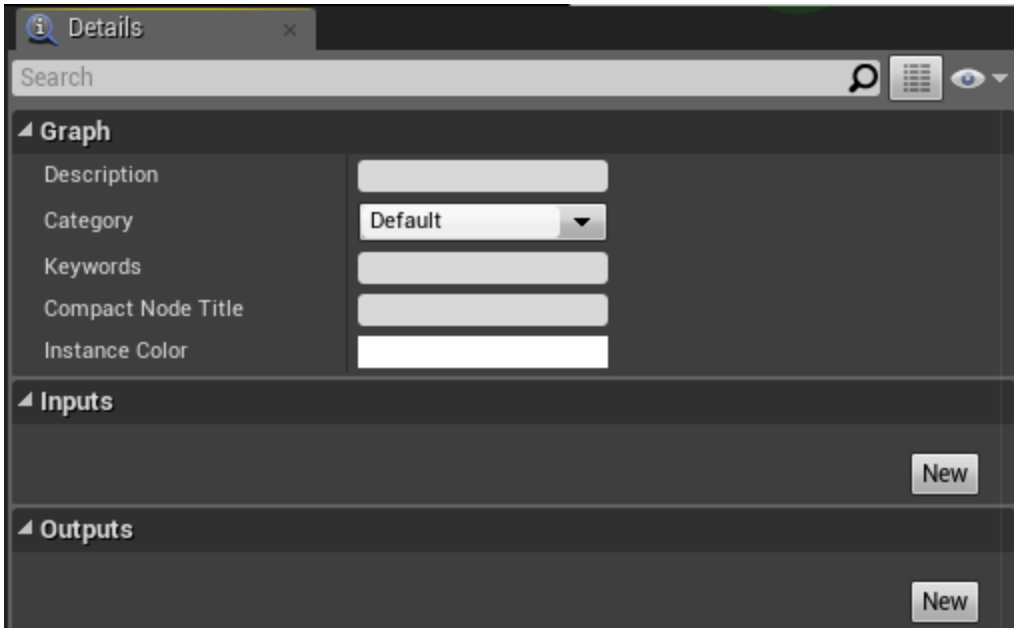
Your Blueprint Macro will open in a new tab in the **Graph** tab of the Blueprint Editor.

You can also create a Blueprint Macro by **Right-clicking** in the **My Blueprint** tab and selecting **Macro**.

Building a Blueprint Macro

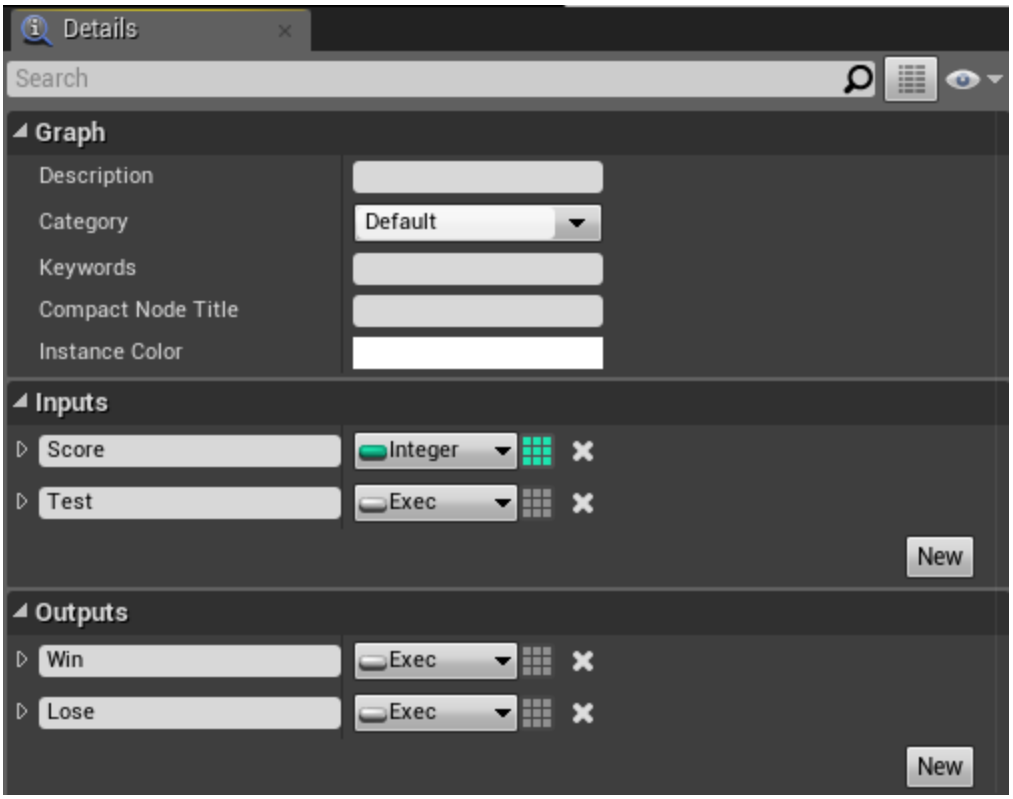
When you first create a Blueprint Macro, a new graph will open containing an **Inputs** tunnel node and an **Outputs** tunnel node.

In the **Details** pane for your Blueprint Macro, you can add input and output execution and data pins. You can also set the **Description**, **Category**, and **Instance Color** of your Blueprint Macro.

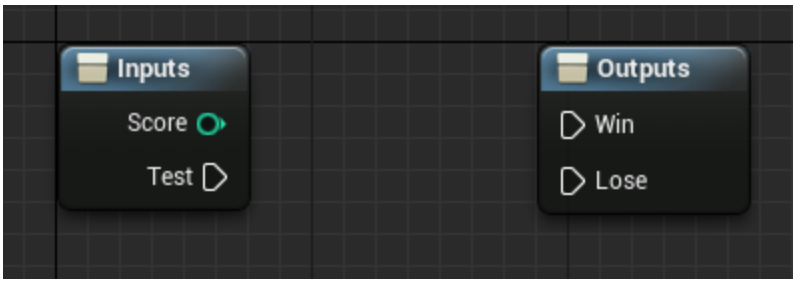


To add input or output parameters:

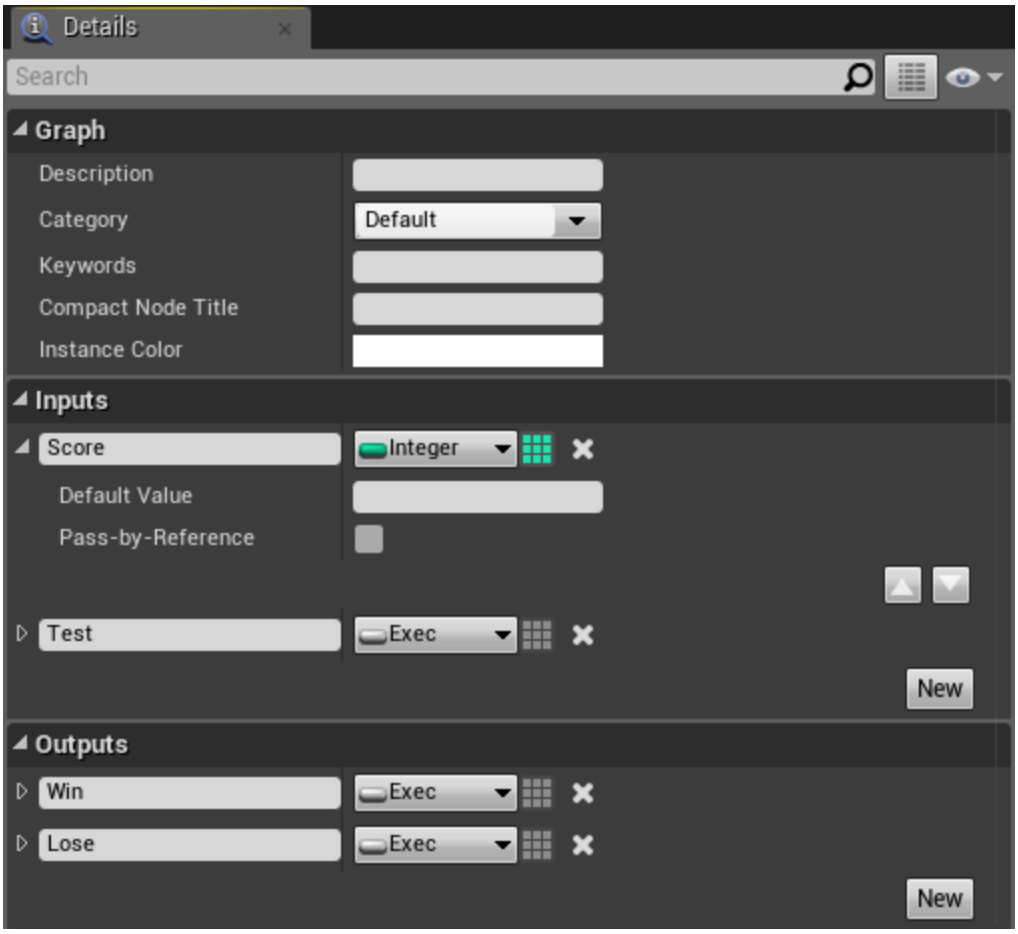
1. Click on the **New** button in either the **Inputs** or **Outputs** section of the **Details** pane.
2. Name the new parameter and set its type using the dropdown menu. In this example, there is an Integer data input parameter named **Score**, an input execution pin named **Test**, and two output execution pins named **Win** and **Lose**.



The tunnel nodes in the Blueprint Macro graph will automatically update with the correct pins.

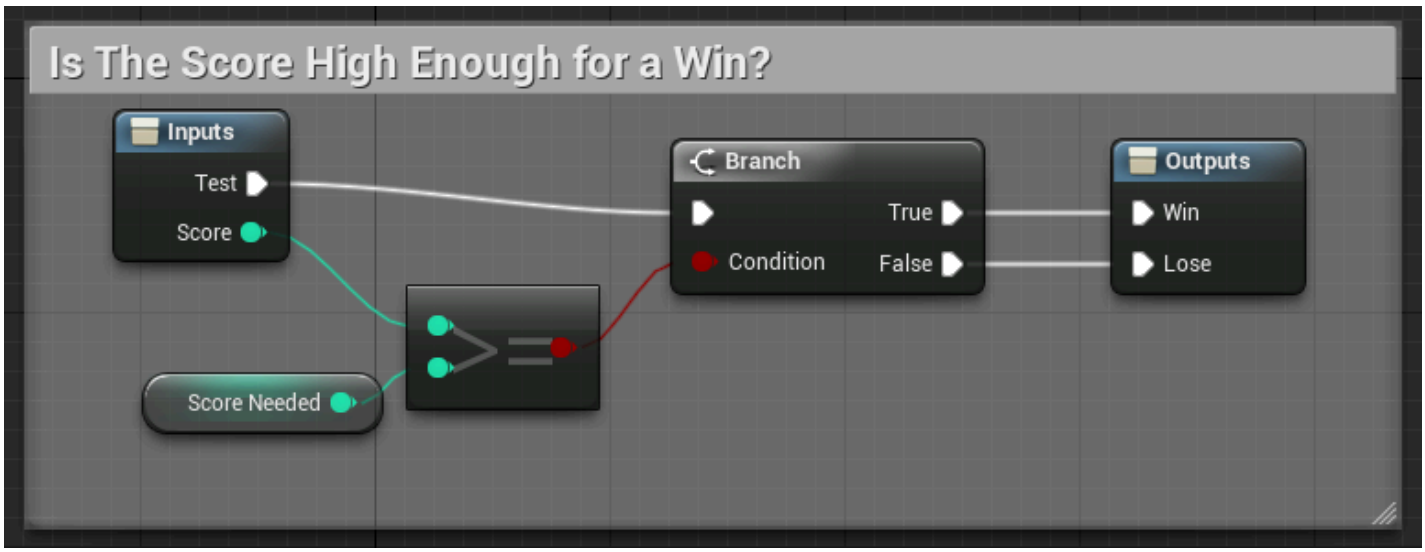


3. You can also set a default value by expanding the entry for the parameter.



To change the location of the pin for this parameter on the edge of the node, use the up and down arrows in the expanded **Details** pane entry.

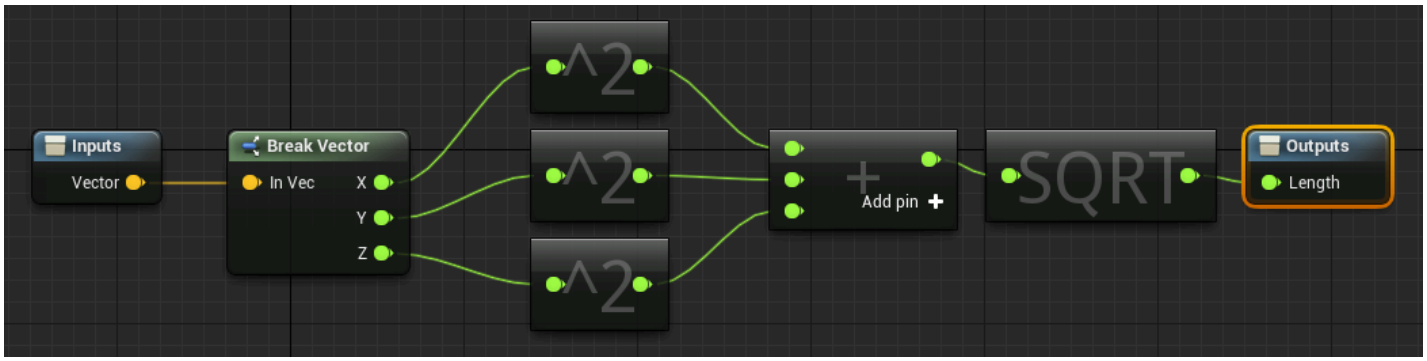
To give your Blueprint Macro some functionality, connect data and execution wires to the pins of your **Inputs** and **Outputs** tunnel nodes and create a network between them.



This example Blueprint Macro checks if the score that is input into the Macro is greater than the necessary score for winning, and then triggers different output execution flows depending on the result of the comparison. Note here that the **Test** and **Score** pins were

flipped using the up and down arrows in the **Details** pane, to avoid wires crossing in the Blueprint Macro graph.

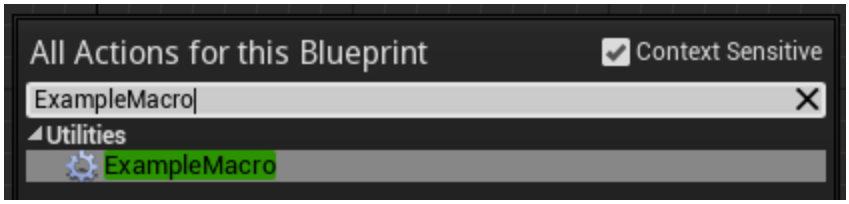
Unlike Functions, Macros can have more than one output execution pin, so you can have execution flow like this where different output execution pins are activated depending on the results of graph logic. Also, you could have a Macro with no execution pins that only manipulates data, as long as the nodes within the Macro are not execution nodes.



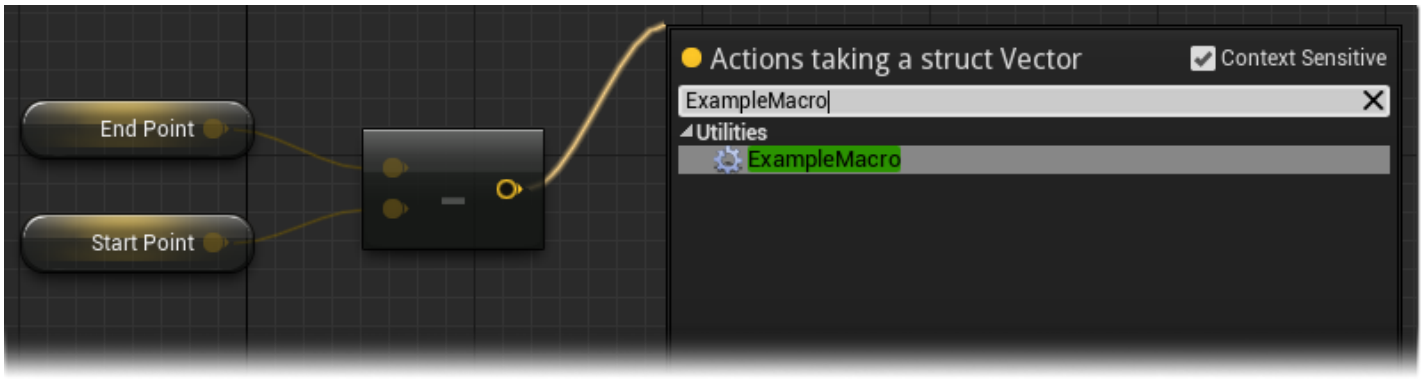
Implementing Macros

There are several ways to add a Macro node to another Blueprint graph. Like Function nodes and Custom Event Call nodes, it is possible to add multiple copies of Macro nodes throughout the graphs in your Blueprint.

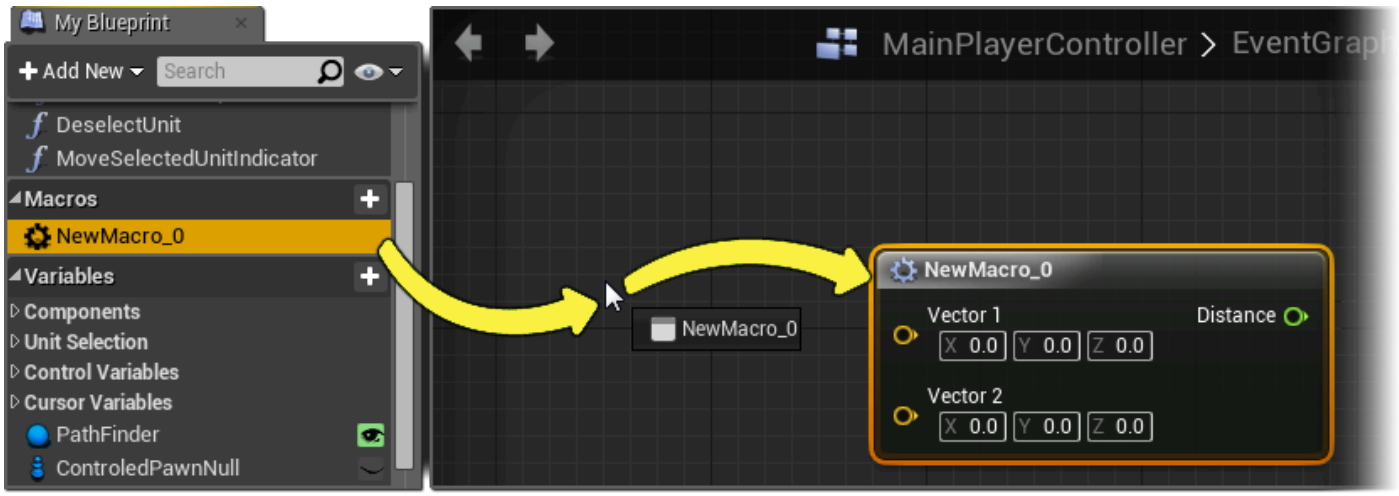
To add a Macro, **Right-click** in the graph and select the Macro in the context menu that appears.



You can also drag and drop off the pin of another node, and the Macro will appear in the context menu if it has a parameter pin of the corresponding variable type and flow direction.



Alternatively, you can drag the Macro from the **My Blueprint** tab and drop it in the graph.



Once the Macro node has been added to the graph, it behaves like any other node and the input and output pins can be wired accordingly. **Double-clicking** on a Macro node will open the Macro's graph.