Using Macro Libraries

Macros inside a Macro Library are used to increase the health and scale of an Actor.

A **Macro Library** is a container that holds a collection of Macros or self-contained graphs that can be placed as nodes in other Blueprints. These can be time-savers as they can store commonly used sequences of nodes complete with inputs and outputs for both execution and data transfer.

> A Blueprint Macro Library cannot contain variables, inherit from other Blueprints, or be placed in levels. Changes to macros in a Blueprint Macro Library will not take effect until client Blueprints are recompiled.

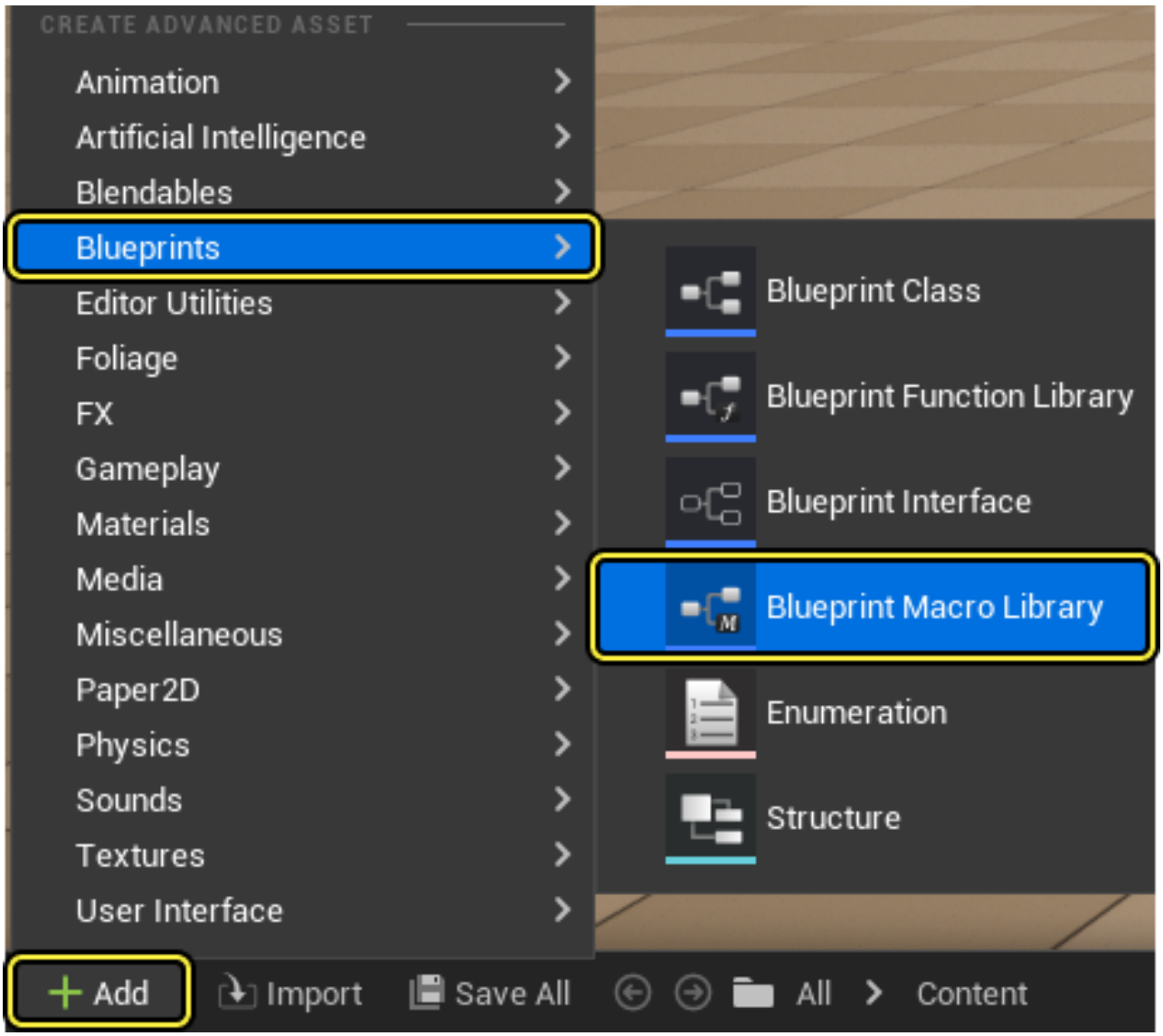
Creating a Macro Library

In this example you will create a Macro Library consisting of two macros, one of which is used to add to a "Health" variable and the other is used to adjust the size of an Actor.

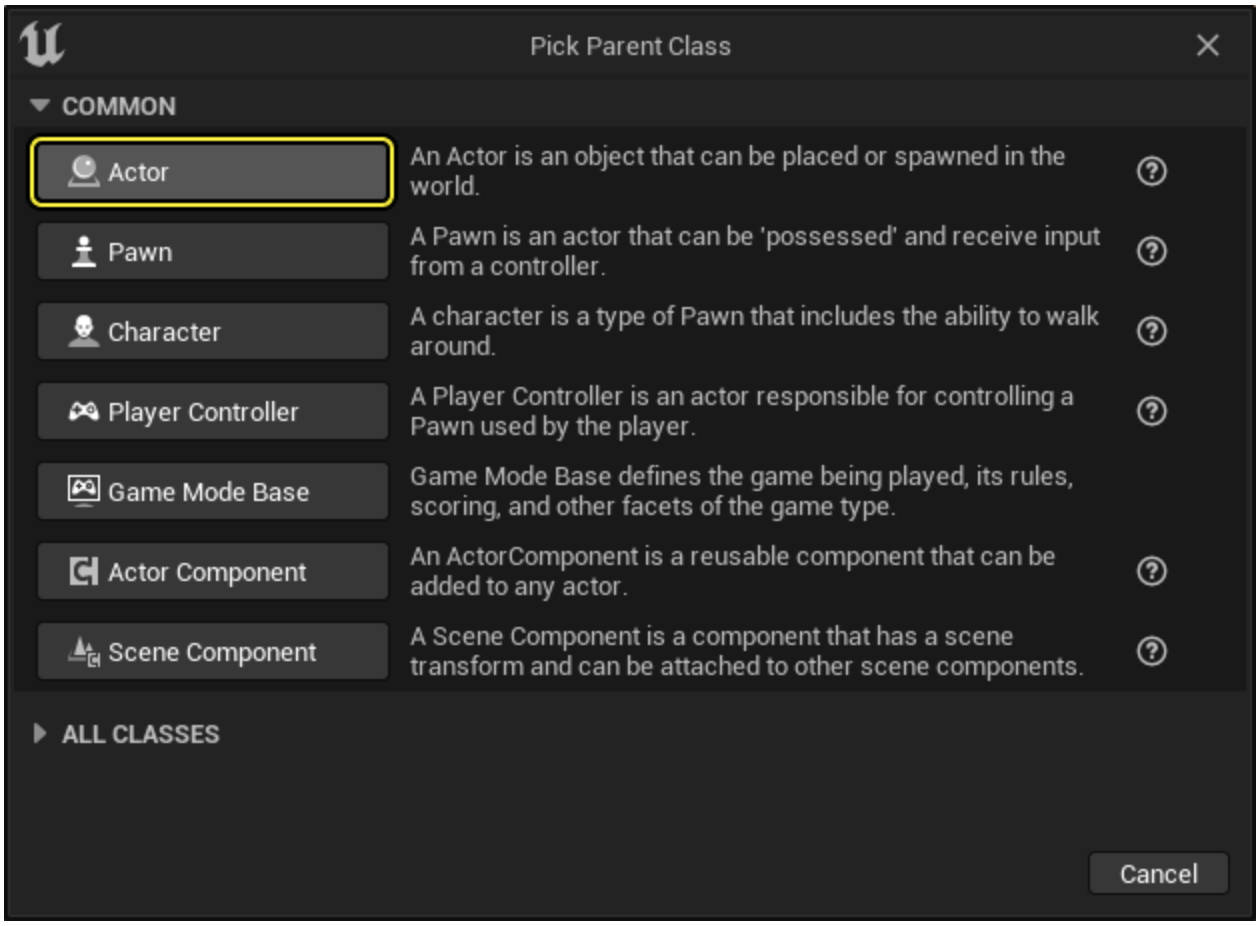
For this example, we are using the Blueprint Third Person Project with **Starter Content** enabled.

1. From the Content Browser, Select **Add** > **Blueprints** > **Blueprint Macro Library**

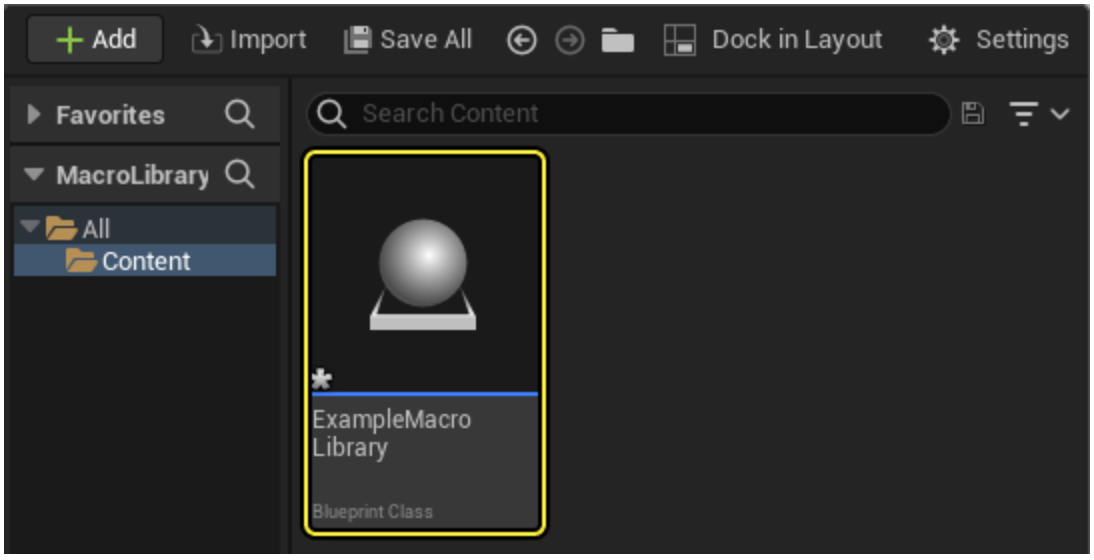
(programming-and-scripting/blueprints-visual-scripting/UserGuide/Types/MacroLibrary).



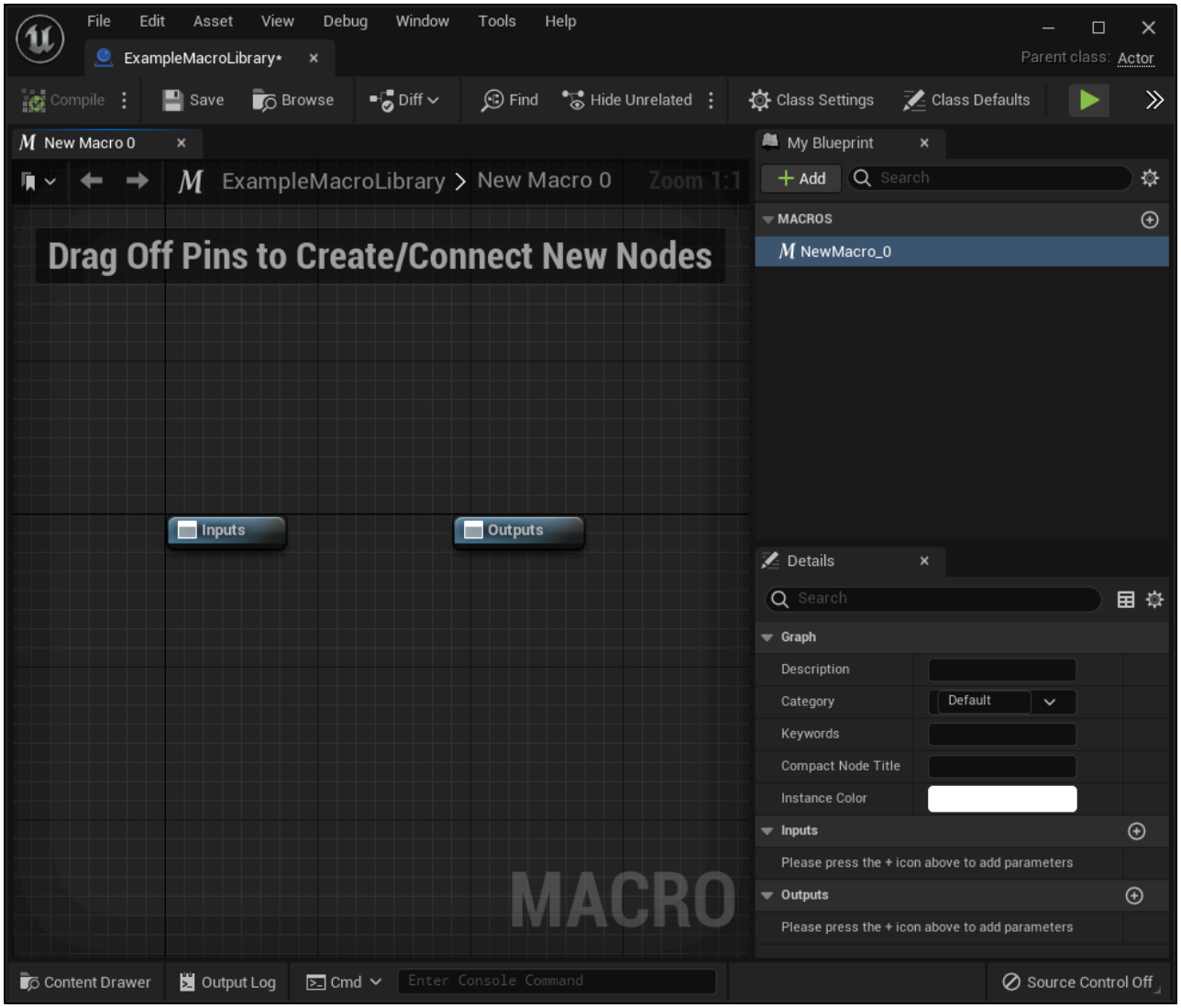
2. In the **Pick Parent Class**, select **Actor** as your Parent Class.



3. Enter a name for your Macro Library, then **Double-click** on it to open it up.

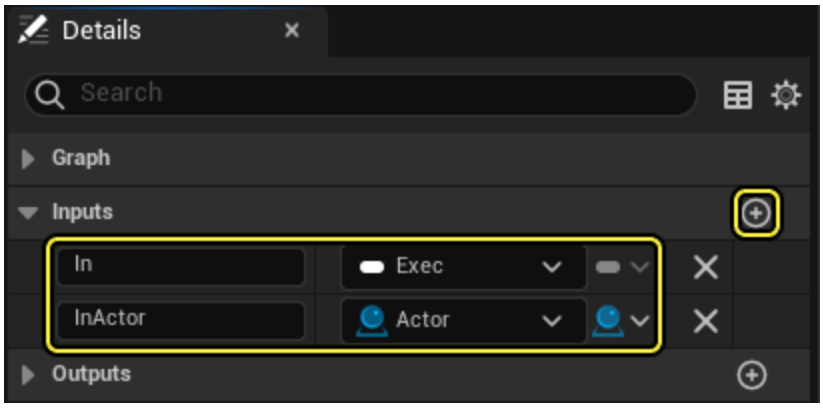


4. This will display the **Blueprint Macro** interface.

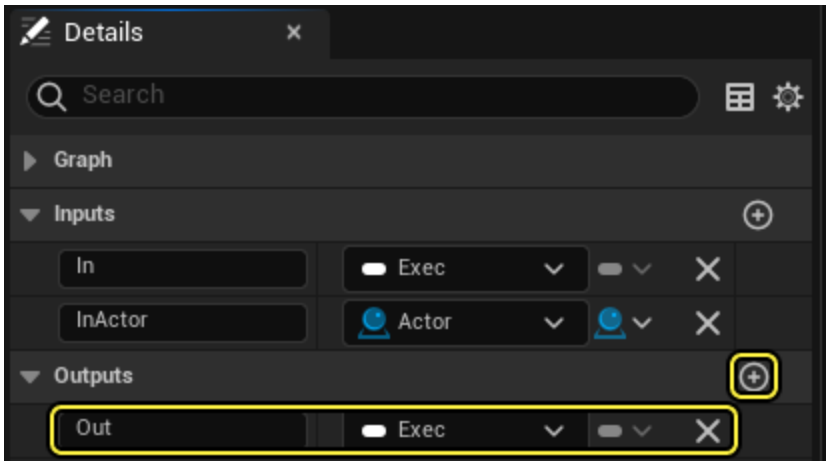


Creating the "Scale Up" Macro

1. Navigate to the **MyBluprint** window and rename the default macro to **ScaleUp** by selecting it and pressing **F2**.
2. In the **Details** panel, navigate to the **Inputs** category and click the **Add** (**+**) button to create two new inputs. Name the first input to **In** and set it to the **Exec** type, then name the other **InActor** and set to the **Actor** type.



3. Navigate to the **Outputs** category and click the **Add** (**+**) button to create a new output named **Out** and set it to the **Exec** type.



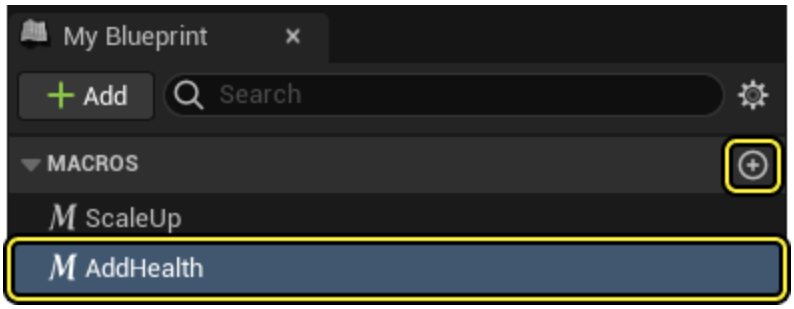
4. In the graph for the **ScaleUp** macro, copy or recreate the Blueprint scripts below.

Copy code

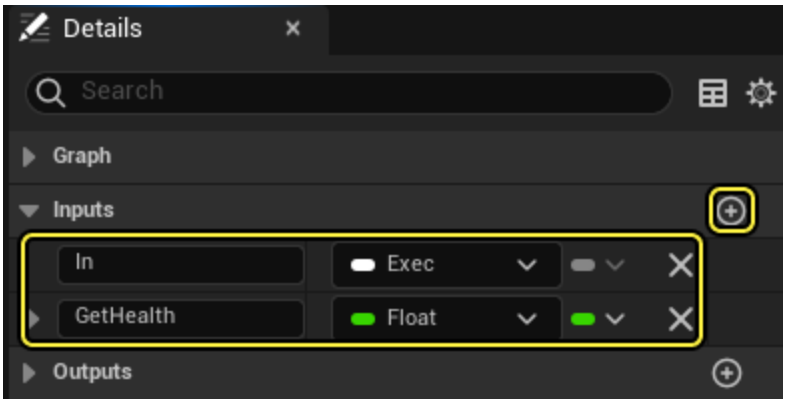
When this macro is called it will get the Actor provided as the **InActor** and get its current scale to multiply by 1.25, then it will set the new scale 3D for the **Target** (the Actor specified as the *InActor*.) You can now use this macro with any Actor and affect their scale through the use of this macro.

Creating the "Add Health" Macro

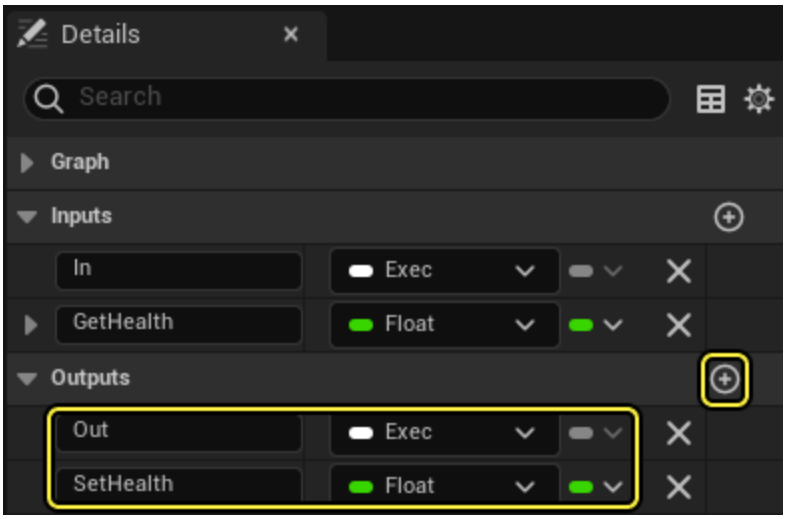
1. Add another macro from the **MyBlueprint** window called **AddHealth**.



2. In the **Details** panel for **AddHealth**, navigate to the **Inputs** category and click the **Add** (**+**) button to create two new inputs. Name the first input to **In** and set it to the **Exec** type, then name the other **GetHealth** and set it to the float type.



3. Navigate to the **Outputs** category and click the **Add** (**+**) button to create two new outputs. Name the first output to **Out** and set it to the **Exec** type, then name the other **SetHealth** and set it to the float type.



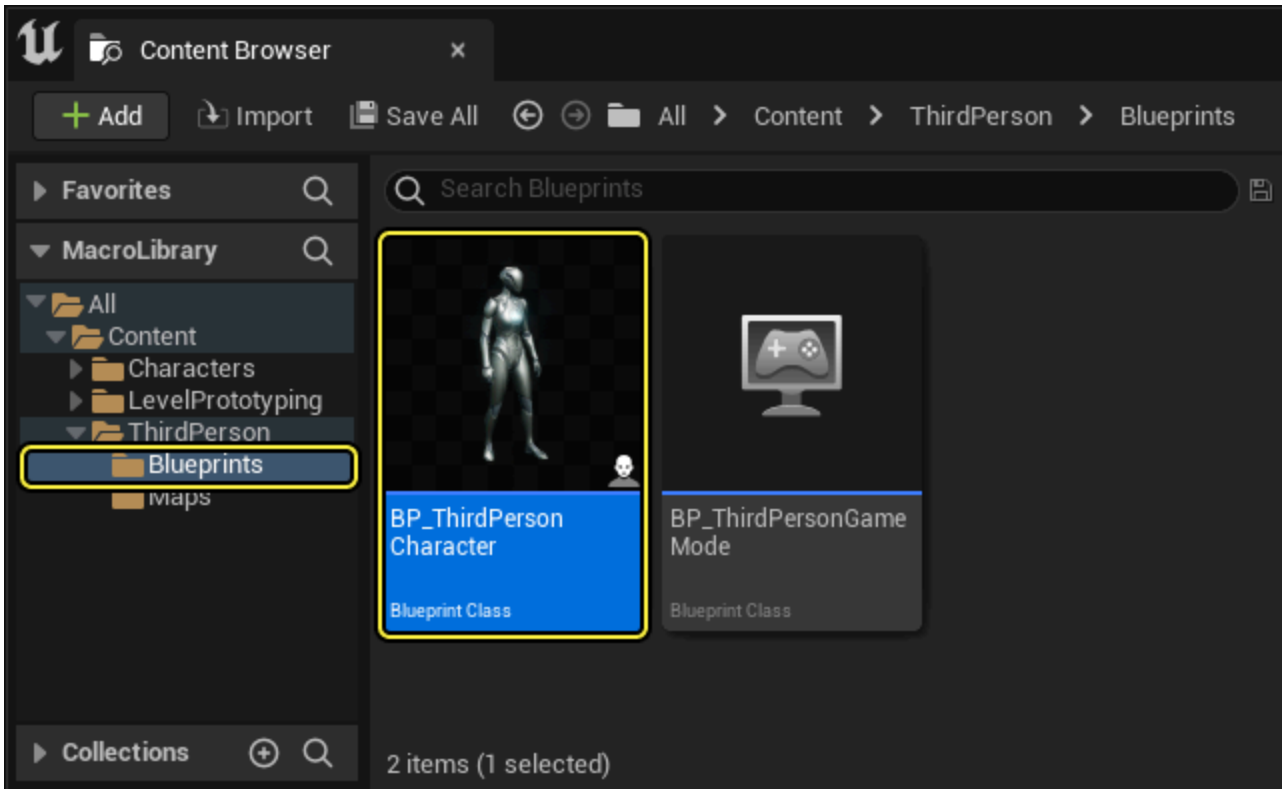
Above we are taking in a float value called GetHealth which we will add to before we pass it through to the SetHealth output node.

4. In the graph for the **AddHealth** macro, recreate the node network indicated below.

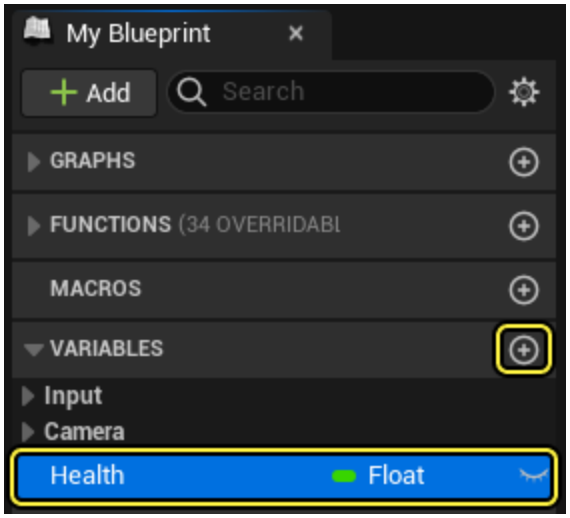
Copy code

Above we are adding **50** to the float value provided as the **GetHealth** value before outputting the resulting value to **SetHealth**.

1. **Save** and close the Blueprint Macro Library.
2. Inside the `Content/ThirdPerson/Blueprints` folder, open the **BP_ThirdPersonCharacter** Blueprint.



3. In the **MyBlueprint** window, navigate to the **Variables** category and click the Add (+) to create a new **Float** variable named **Health**, then click **Compile**.



4. In the Event Graph below, copy or recreate the Blueprint scripts.

In the logic below, we are calling the **AddHealth** macro whenever **Q** is pressed which takes in the **Health** variable (performs the macro script of increasing it) and updates it with the out pin **SetHealth** before printing it to the screen. We then use **E** and check if **Health** is greater than 100 before we call the **ScaleUp** macro on the Actor **Self** which is the **ThirdPersonCharacter**.

Copy code

5. **Compile** and **Play** in the editor.

End Result

In the video above, we press the **E** key to call the **ScaleUp** macro. The Blueprint logic will first check the **Health** value which by default we set to 0, then we press **Q** to call the **AddHealth** macro to increase the *Health* variable by 50. After a few key presses, you can see the character increase values in both health and scale. These macros can be called from any other Blueprints provided the inputs for Health and the Target Actor are supplied.

Creating Macros

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.

Refer to Making Macros for an example on creating a macro.