Blueprint Interface

Blueprints that declare functions to define an interface between Blueprints.

A **Blueprint Interface** is a collection of one or more functions - name only, no implementation - that can be added to other Blueprints. Any Blueprint that has the Interface added is guaranteed to have those functions. The functions of the Interface can be given functionality in each of the Blueprints that added it. This is essentially like the concept of an interface in general programming, which allows multiple different types of Objects to all share and be accessed through a common interface. Put simply, Blueprint Interfaces allow different Blueprints to share with and send data to one another.

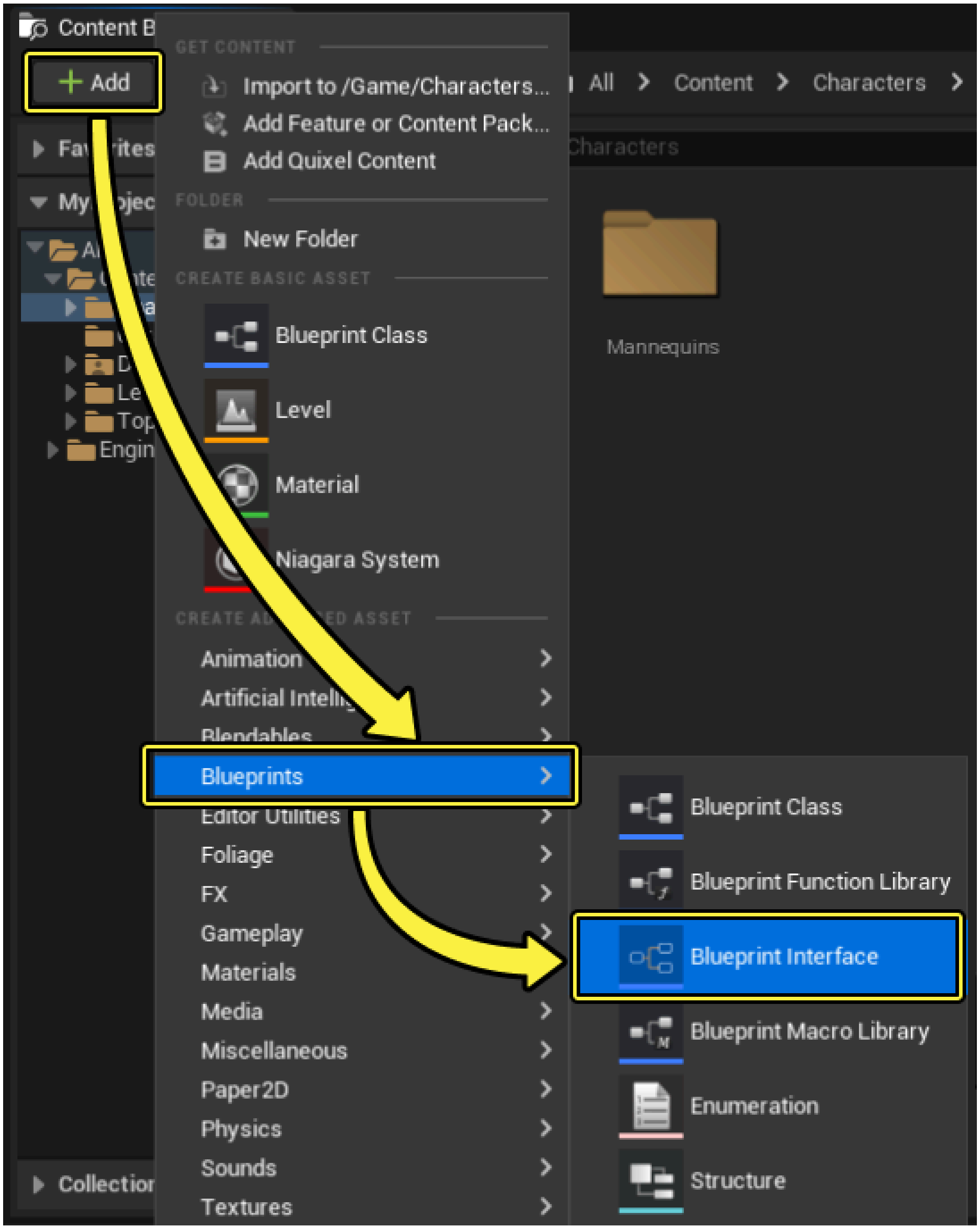
Blueprint Interfaces can be made by content creators through the editor in a similar fashion to other Blueprints, but they come with certain limitations in that they cannot:

- Add new variables
- Edit graphs
- Add Components

The use of Blueprint Interfaces allows for a common method of interacting with multiple disparate types of Objects that all share some specific functionality. This means you can have completely different types of Objects, such as a car and a tree, that share one specific thing like they can both be shot by weapon fire and take damage. By creating a Blueprint Interface that contains an `OnTakeWeaponFire` function, and having both the car and the tree implement that Blueprint Interface, you can treat the car and the tree as the same type and simply call the `OnTakeWeaponFire` function when either of them is shot. Read about how to implement Blueprint Interfaces on the Implement Interface in Blueprint page.

Creating Blueprint Interfaces

Click the **Add** in the **Content Browser**, then select **Blueprints > Blueprint Interface**. Name your new Blueprint Interface.



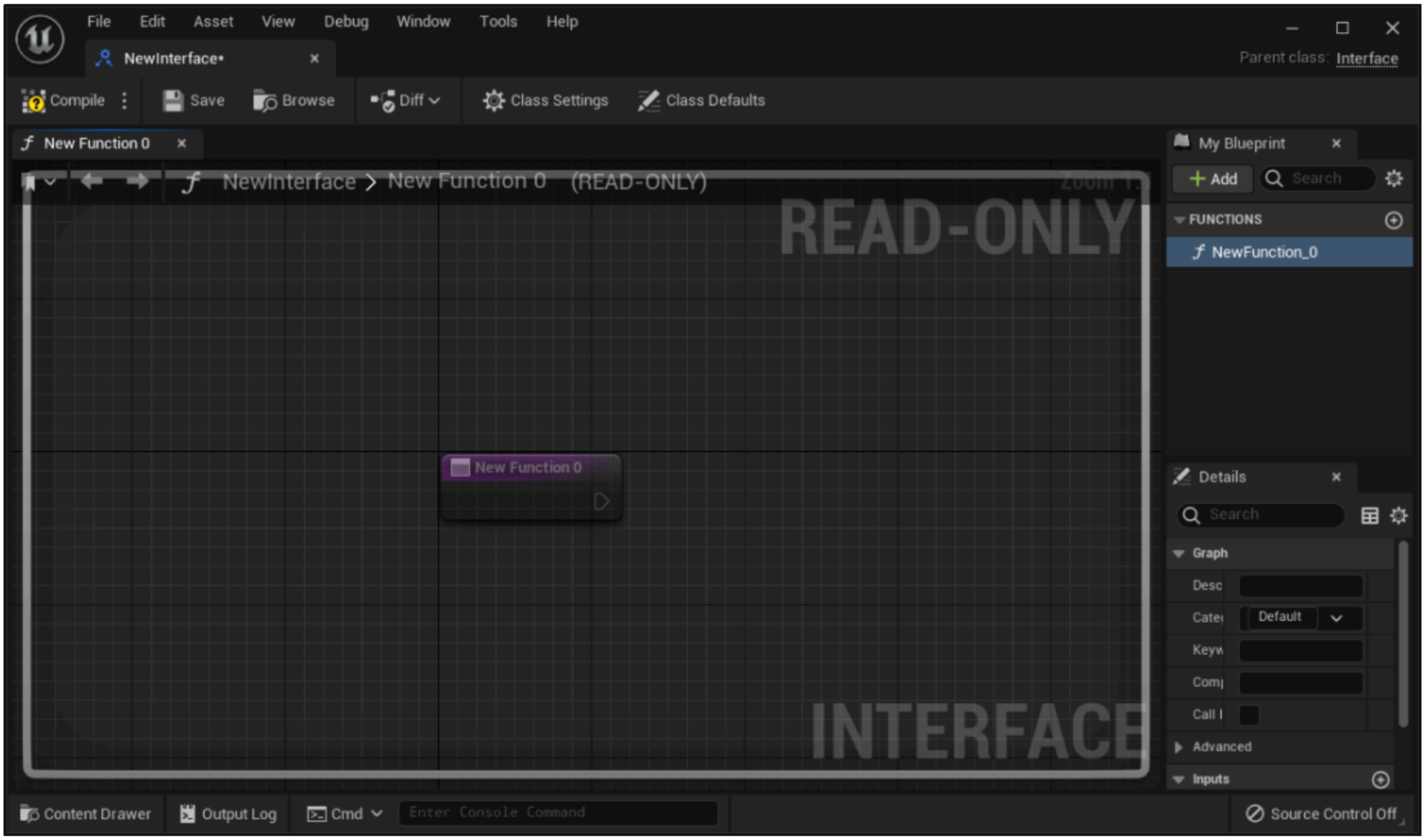
You can also add a Blueprint Interface by right-clicking into **Content Browser**, then **Blueprints > Blueprint Interface**.

Once created, you need to edit the Interface's functions.

Editing Blueprint Interfaces

Blueprint Interfaces are edited by the **Blueprint Editor**. Since you cannot create your own variables, graphs, or components, the process of Interface editing differs greatly from the process of editing standard Blueprint Classes.

When you first open up a new Interface, the editor looks like this:



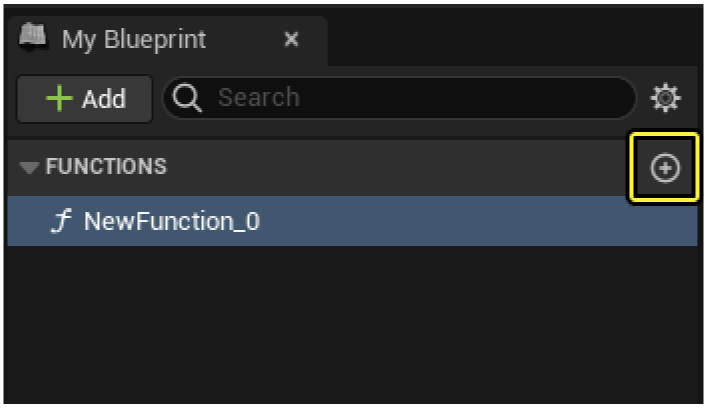
If you have just created your Interface, you will see that the Editor has created a new blank Function for you named **NewFunction_0** and it will be highlighted for you to rename it.

Adding Functions

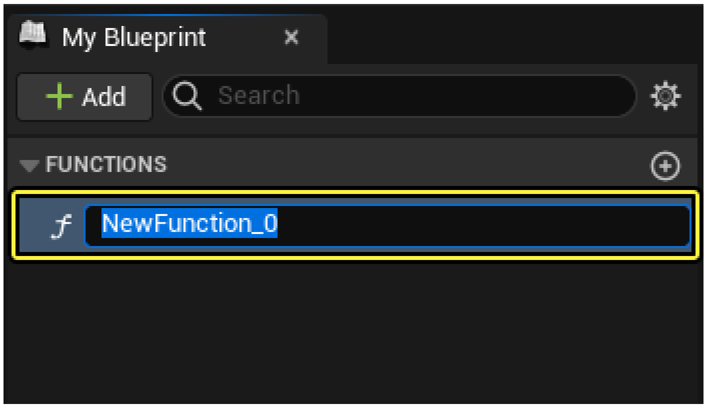
Functions are the primary component of an Interface. Interface functions have no implementation. They exist simply as a definition of inputs and outputs. These can be used to send data through the interface, or can be overridden within any Blueprint that implements that Interface.

To add a new function:

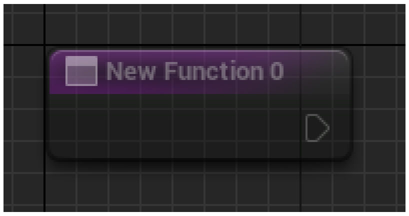
1. In the **My Blueprint** tab create a new function, by clicking on the **+** icon on the functions list header.



2. In the **My Blueprint** pane, enter a name for the new function.



3. The new Graph area will appear with the new function. Note that the function has neither inputs nor outputs.

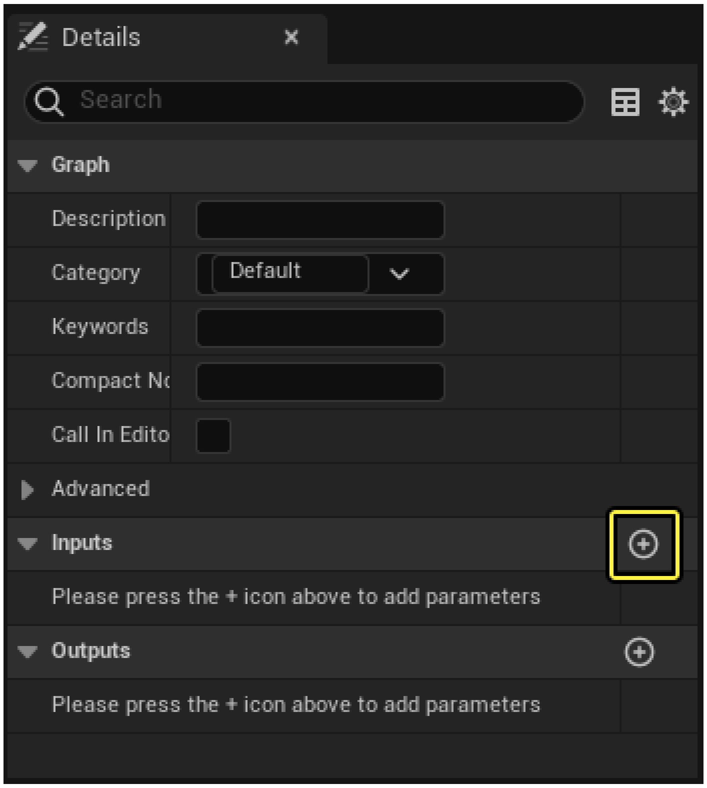


Editing Function Signatures

Since an interface function has no implementation, all you can do as a designer is designate a series of typed inputs and outputs.

To edit a function's signature:

1. In the **Details** tab, scroll to the **Inputs** category and click the **+** icon to create a new Input Parameter.

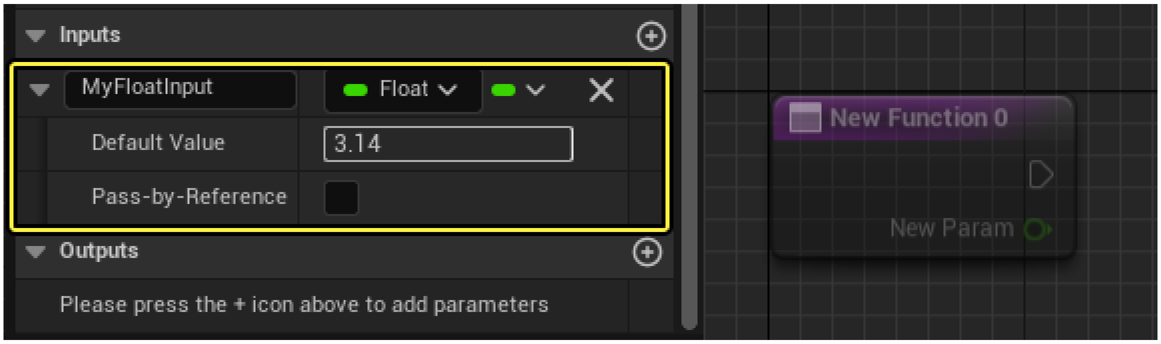


Category: Default

Please press the + icon above to add parameters

Please press the + icon above to add parameters

2. Set the Input Name and Type as desired. You may also expand the input using the ▷ button next to the name, and thereby set a Default Value.



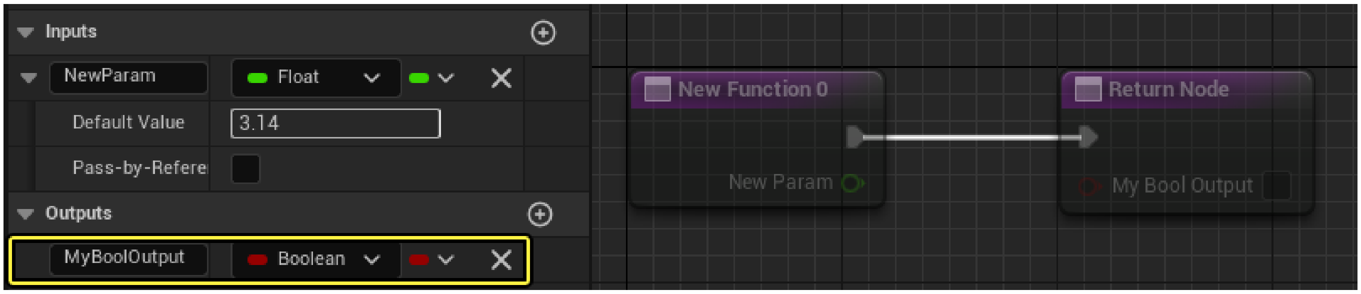
MyFloatInput Float

Default Value 3.14

Pass-by-Reference □

Please press the + icon above to add parameters

3. In the same manner, outputs can also be added. Note how the graph automatically updates to show them.



Other Considerations

- On any input or output parameter, you can click the ✕ button to remove that parameter.
- Input parameters can be given a default value using the small text field immediately under the parameter name field, though you must expand the property entry in the **Details** tab to see it.
- You can change the order of the input and output parameters on the node. Drag the right edge of the parameters in the **Input** and **Output** sections to adjust them to the order you need.
- You can use the Replicates checkbox for any Interfaces containing functions that need to replicate across the server. This is found within the **Details** tab by first clicking the **Class Settings** button.