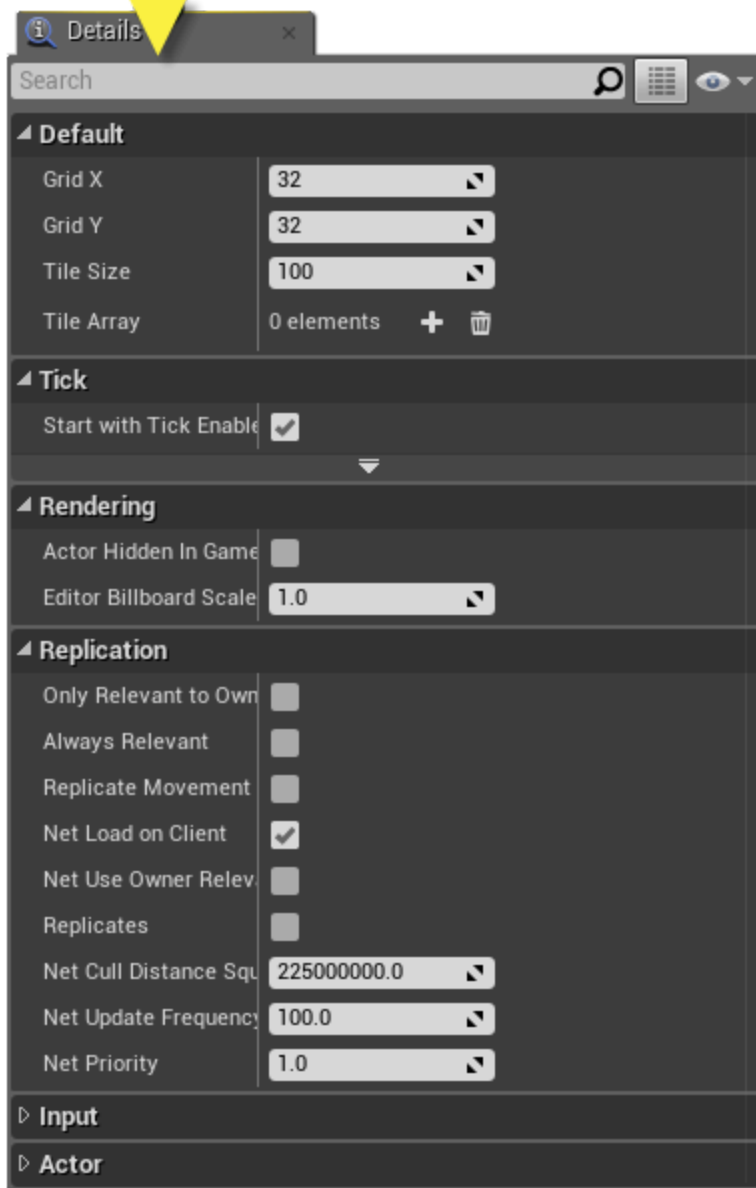# Blueprint Editor Defaults Tab

An overview of the Blueprints Class Defaults tab

The **Class Defaults** tab contains information about the Blueprint's default settings and properties, and about any variables the Blueprint contains. You can modify these settings to suit your needs.

The **Class Defaults** tab contains the following sections:

Item	Description
Default	Lists any variables and their values if you do not specify a different category than **Default** when you create them. Note that you must compile the Blueprint in order to display its variables on the **Class Defaults** tab.
Rendering	Contains information about properties related to how (and whether) an Actor based on the Blueprint is rendered in the game.

| Item	Description
Replication	Contains information about properties that dictate how your Actor behaves in a networked game, such as its network priority and relevancy to other clients.
Input	Contains information about how Actors based on this Blueprint respond to input.
Actor	Contains information about Actors based on this Blueprint, such as whether they can be damaged. Note that for Level Blueprints, only the **Tags** properties are relevant in this section.

> ⓘ If you create custom categories for variables, the **Class Defaults** tab will display those categories too, after the Blueprint is compiled.

Get Class Defaults

The properties defined inside the Class Defaults can also be accessed at runtime through the use of the **Get Class Defaults** Blueprint node. Targeted use case is for data-only Blueprint Class types and other cases where you may not necessarily need/want to create an instance in order to access default property values.

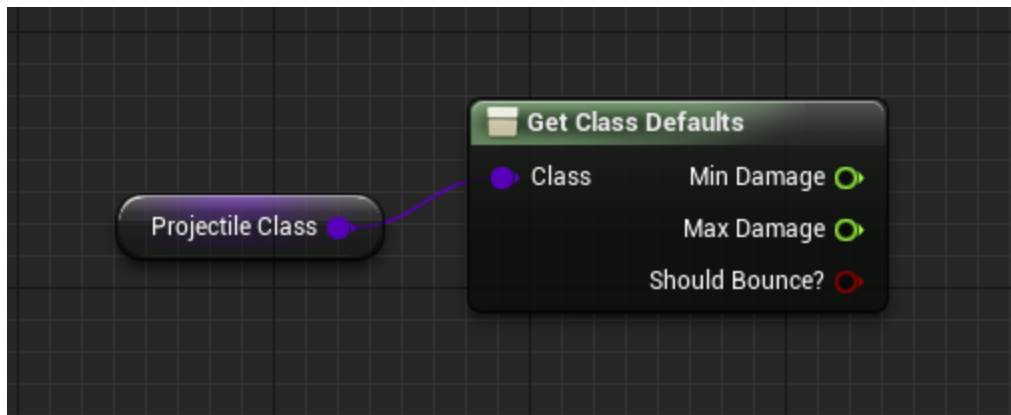**To Access the Get Class Defaults Node**

1. **Right-click** inside your Blueprint Class or Function Graph and enter **GetClassDefaults** from the context menu.
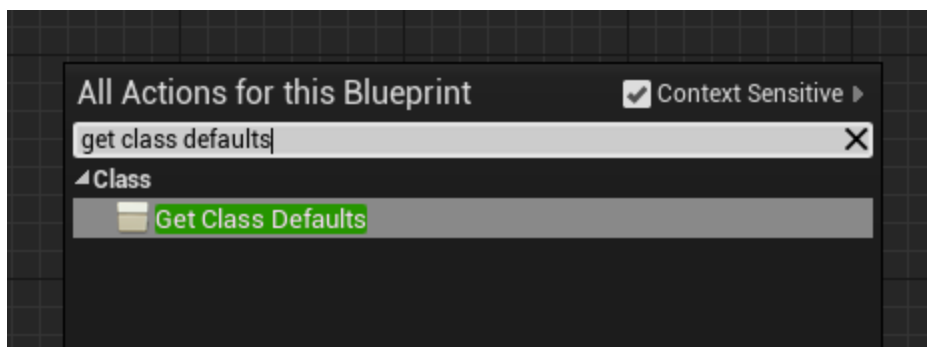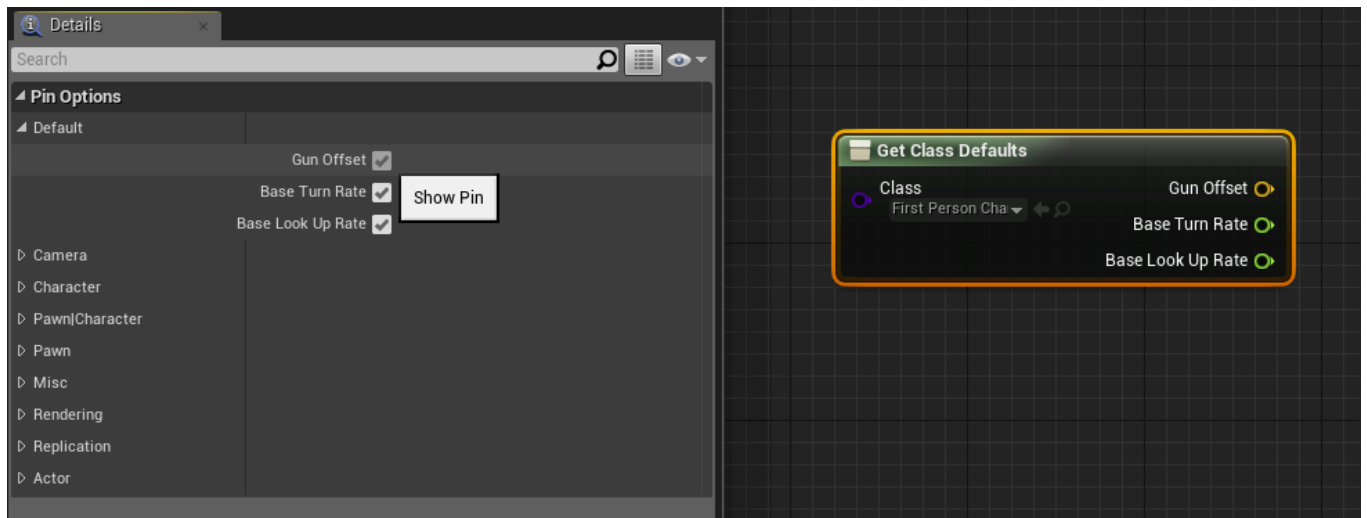

2. Choose the class to access via the class picker drop-down menu.
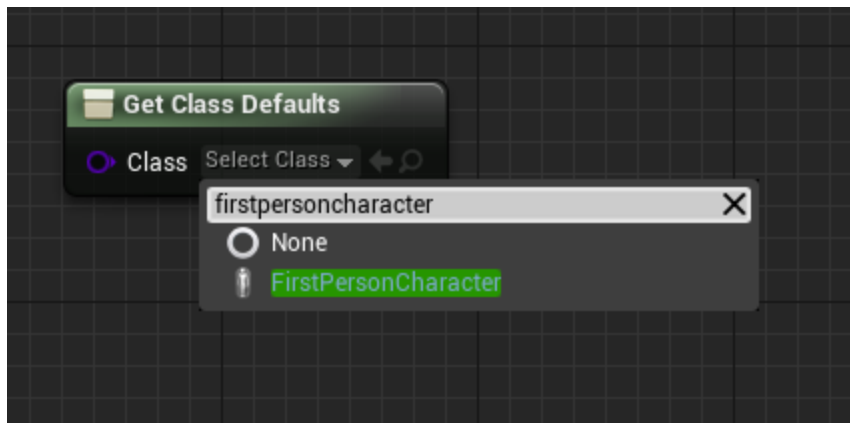
You can also wire in a class type result from a function call or variable instead as shown in the example below.

3. With the node selected, you can show/hide pins via the **Details** panel.

Usage Notes

- Output pins (accessible class properties) are shown by default and can be optionally hidden (similar to Struct Break/Set Nodes).
  - Parent class properties are currently hidden by default to reduce the number of output pins.

- Properties tagged as *Advanced View* are also hidden by default in the **Advanced** node section.
- Object reference property default values are **not** currently exposed.
  - This means you currently cannot, for instance, access the default property values of component templates or other archetypes within a Blueprint Class.
  - This is because we do not currently enforce read-only reference types in a Blueprint, and we'd need to do that in order to ensure that the internal state of the referenced object cannot be changed.\
- Only **BlueprintVisible** or **BlueprintReadOnly** properties are currently exposed for defaults access.
  - This matches the behavior of "normal" variable Get nodes exposed through the context menu.
- Currently the *Class* input drop-down lists **all** UOBject-derived class types (including engine classes).
  - This may be limited in the future to exclude engine-specific class types.
- Adding a **GetClassDefaults** node to an Actor-based Blueprint class (for example) will currently default the input class pin's value to match the current Blueprint Class type when placed.