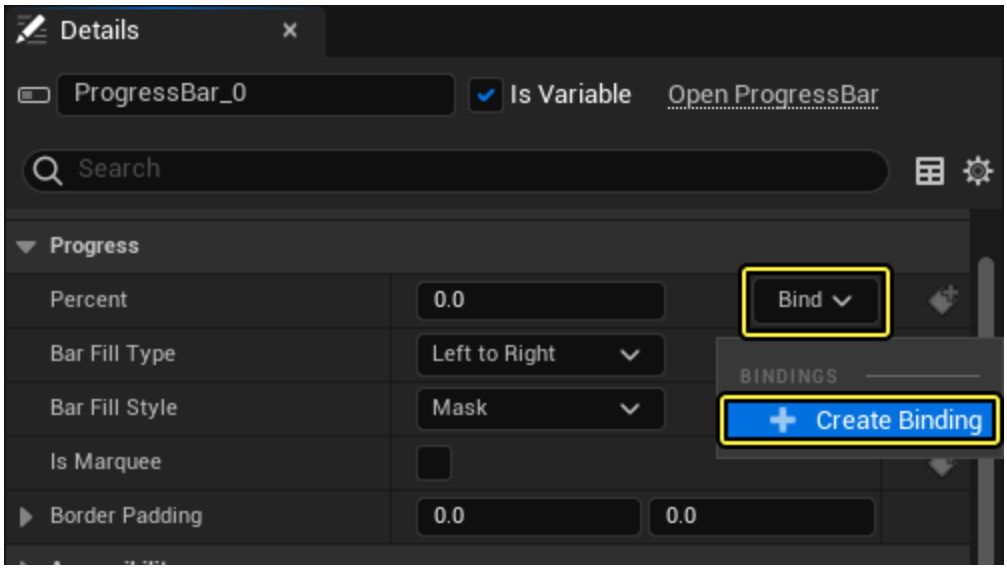
Property Binding

A guide explaines how to bind properties inside UMG.

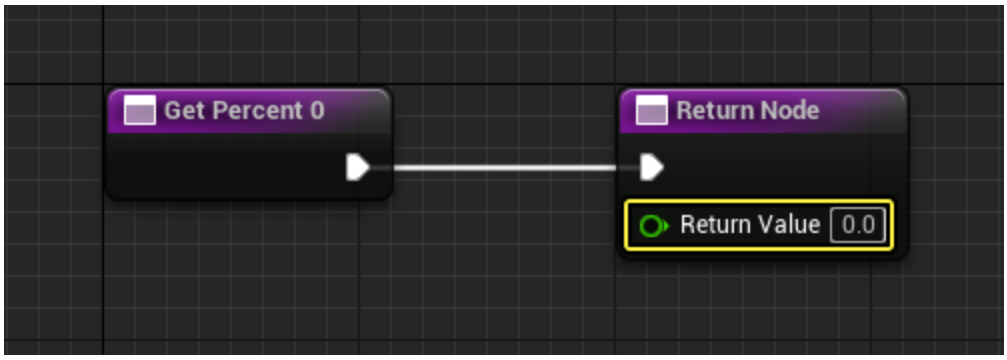
One of the most useful aspects inside UMG is the ability to bind properties of your Widgets to Functions or Properties inside the Blueprint. By binding a property to a Function or Property Variable in your Blueprint, anytime that Function is called or Property is updated, it will be reflected in the Widget.

Function Binding

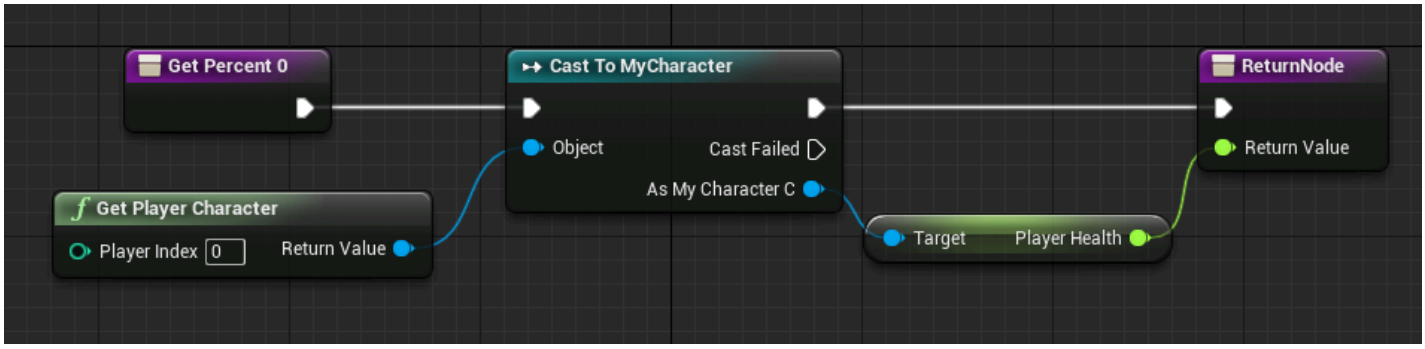
Say for instance you had a **Progress Bar** Widget and you wanted to update it to reflect a player's health at all times. Under **Appearance** for the Progress Bar you will notice a **Percent** option with the option to **Bind** the Percent value to a Function or Property.



After clicking the **Bind** button and selecting the **Creating Binding** option, a new Function will be created and open up.



The **Return Value** is tied to the Percent's Value, so you can plug in a value to feed your Progress Bar data.

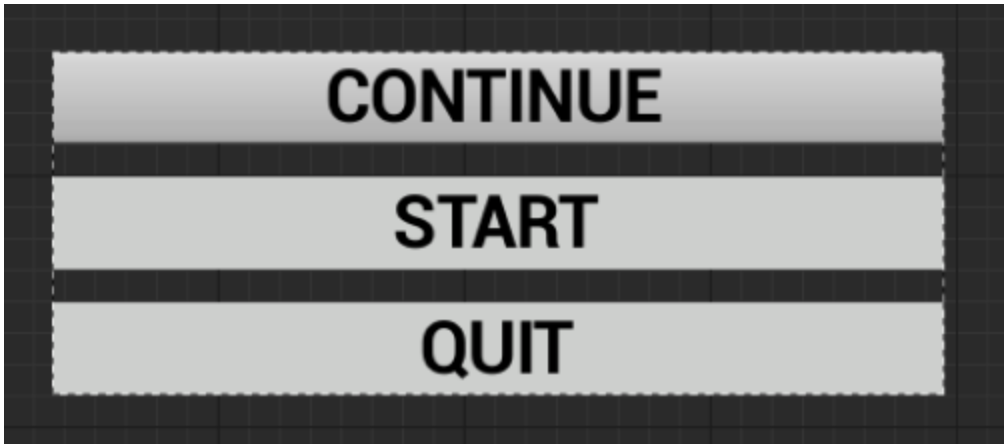


Above, the Function gets a variable called **Player Health** from inside our Character Blueprint. Anytime the Player Health variable is updated, it is automatically passed to and reflected as the Percent Value on the Progress Bar.

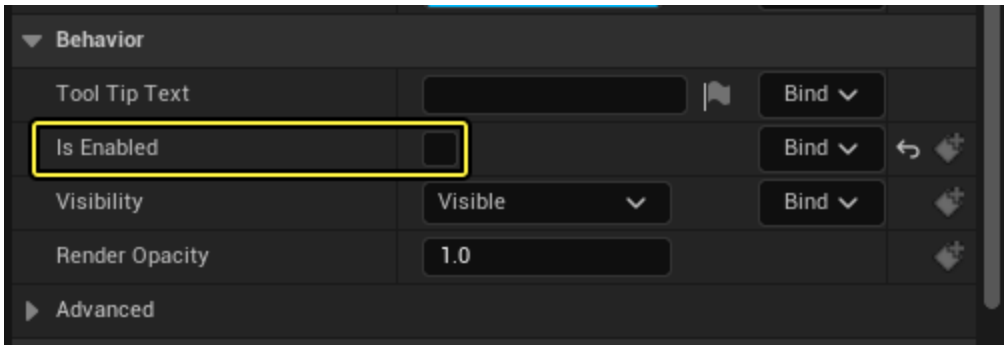
Property Binding

Property Binding consists of specifying a Property Variable that is bound to a Widget's property. When the Property Variable is updated, the setting that is bound to it is automatically updated and reflected in the Widget.

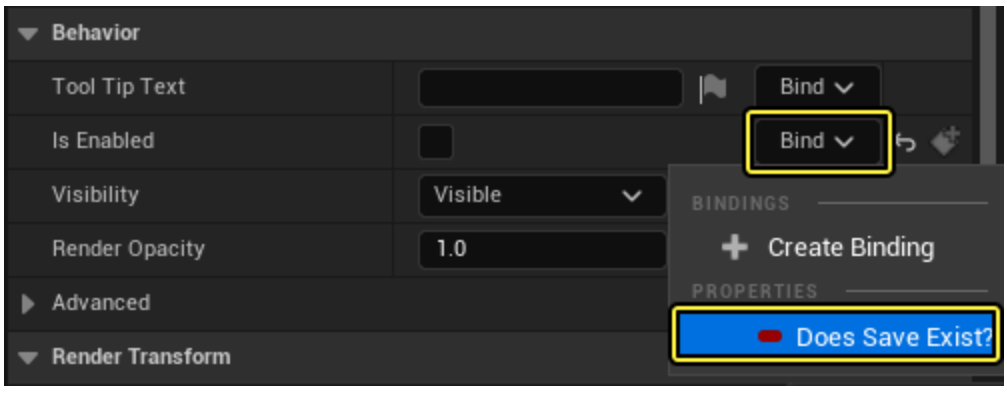
An example of binding a Property Variable for a button is illustrated below.



In the image above, you can see three button for a Main Menu: **Continue**, **Start**, and **Quit**. You can make **Continue** button to only be enabled if the player has a Save Game. Under the **Behavior** section for the **Continue** button, you can un-check **IsEnabled** (so the button is not enabled by default) and then click the **Bind** button.



On the **Graph** tab for this Widget Blueprint, you could then create a **Boolean** variable and once it has been created, can assign it via the Bind button (below we have created a Boolean called **DoesSaveExist?**).



With this variable bound to the **IsEnabled** Behavior, you could then set this whenever your game is started by checking if a save file is present and if so, Casting to this Widget Blueprint to access and set the **DoesSaveExist** variable to *True* which would then enable the button.

If you bind a property of a Widget and then directly call the **Set** function on that Widget, it will break the binding.