Table of Contents

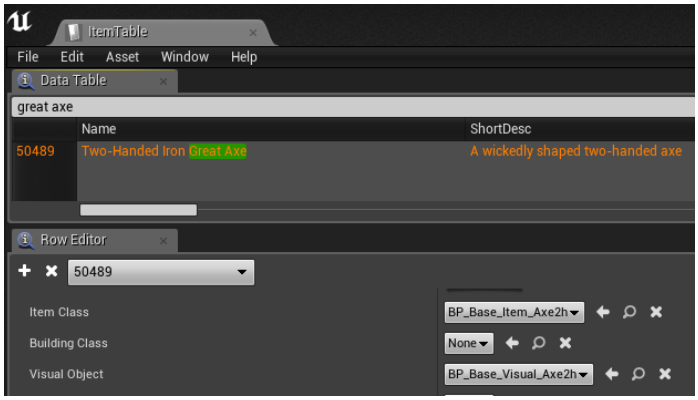
About this tutorial

This tutorial will go over how to easily make changes to weapons in order to customize them in various ways, from adding buffs to them to implementing additional damages. It will not cover all the bases for weapons, but based on this tutorial, it should be possible to find the start of the red thread that will unveil weapon-crafting in the dev-kit.

Types of Weapon Blueprints

There are two weapon blueprint types we will look at in this guide. First up is the Visual Blueprint, and then we will look at the Item Class blueprint.

You can find them in the item-table (see image on the right). This is where you will need to assign them to your weapon once you've created it.



Visual Blueprint

The visual blueprint determines how the weapon will show up visually (that almost makes sense!), meaning, you can control almost any aspect of the weapon through this blueprint - you can also add lights or other features to it if you want to expand it.

Worth noting is that you should always make a blueprint the child of a pre-existing blueprint. In the example to the right, the basic blueprint is "BP_Base_Visual_Axe2h", and so if we wanted to make a special axe that glows in the dark, we should make a child of this blueprint and add our functionality in that one.

If you open up the Axe visual blueprint, you can see that the logic for it is completely empty. That's because most weapons don't have fancy things attached to them, so all the changes you might want to do will need to be functions that override the functions of the parent.

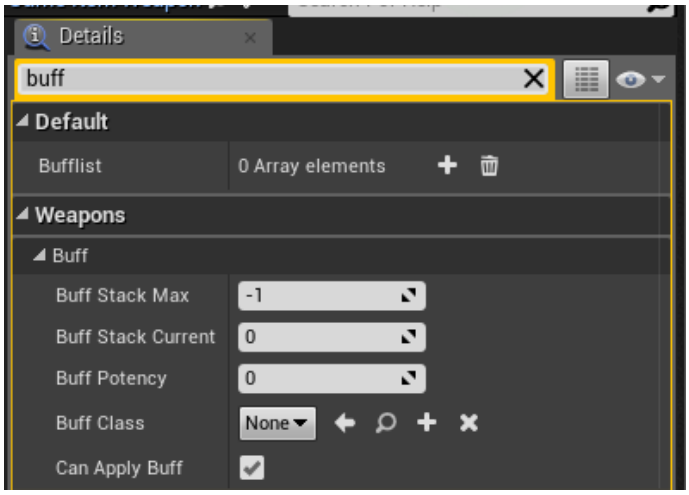
If you open up the Dagger visual blueprint however (BP_dagger_dw), you will see that this blueprint has quite a lot of logic attached to it. Most of this logic is about copying the visual mesh for the dagger, turning it 180 degrees and attaching it to the left hand for the avatar. So - if you want your one-handed sword to be duplicated into the left hand, here's where you can implement the same kind of overrides as in the dagger blueprint.

You can also add components like Lights and Particles and audio-components into the Components-list, like with the "BP_Torch_1h" or the "sword2h_legendary_glow" (this is the visual blueprint for the Jedias Greatsaber weapon).

Item Class Blueprint

The Item Class Blueprint is where most of the really nifty stuff exists though. Most (but not all) of the weapons use the "BPGameItemWeapon" item class. The weapons that don't have other custom functions already built into them, but ultimately they will all be the same, barring these special functions.

Details Pane - how to add buffs



One of the easiest things to add to your weapons are buffs. Searching for "buff" in the Details pane will show you something like the image on the right. So let's go through those entries first.

Bufflist

This list of buffs allows you to run buffs when the weapon is equipped. This allows you to put any kind of buff on the player that uses the weapon - do you want a weapon that poisons the player that uses it? Here's where you put it.

What about a weapon that corrupts the player? Or heals them? Or Cripples, Corrupts, poisons and Bleeds them all at the same time? This is the place.

You're not limited as far as the amount of buffs go but it's generally good practice to not add too many here. This is also not the place to put bonuses to Stats like Strength, Agility etc (you **can**, there just happens to be a simpler way, which is covered in the file "Primer - Armor Bonuses.pdf" included in this .zip file - I know it says "armor" but the same is true for weapons there.).

Buff

Under this heading you can find a lot of buff entries. Mostly, you're not going to want to change these values, as they change during runtime of the game. However, there are two interesting things here:

Buff Class

Buff Class is the buff you want to run on the target when hit by the weapon. This buff is limited to only a single entry, but you can easily make a single buff that starts other buffs. We won't cover how in this guide, but you can check out the "Creating Buffs.pdf" included in this .zip file for a basic guide about buffs.

Can Apply Buff

This is a very badly named entry. It has nothing to do with the buffs mentioned above. In fact - this determines if your weapon can be poisoned or not. If you have a special weapon and someone applies a poison to it, this will overwrite your buff, so if you don't want that to happen, you should untick this box. It does mean you can't poison the weapon, but... you can't have everything. And of course, there are ways around this as well, they just won't be covered in this guide as it's intended to just get you started.

Functions

Next, we'll look at functions for BPGameItemWeapon (or whatever your one may be, for example "BP_Item_MaulBase"). There are 48 functions available for this, and we won't cover all of them. If you want to, you can open up the parent class and look at all the different base functions. Here, we will merely cover the most interesting one, so - without further ado - I present you with:

Get Attack Stamina Cost

This function allows you full control over the stamina cost of the weapon.

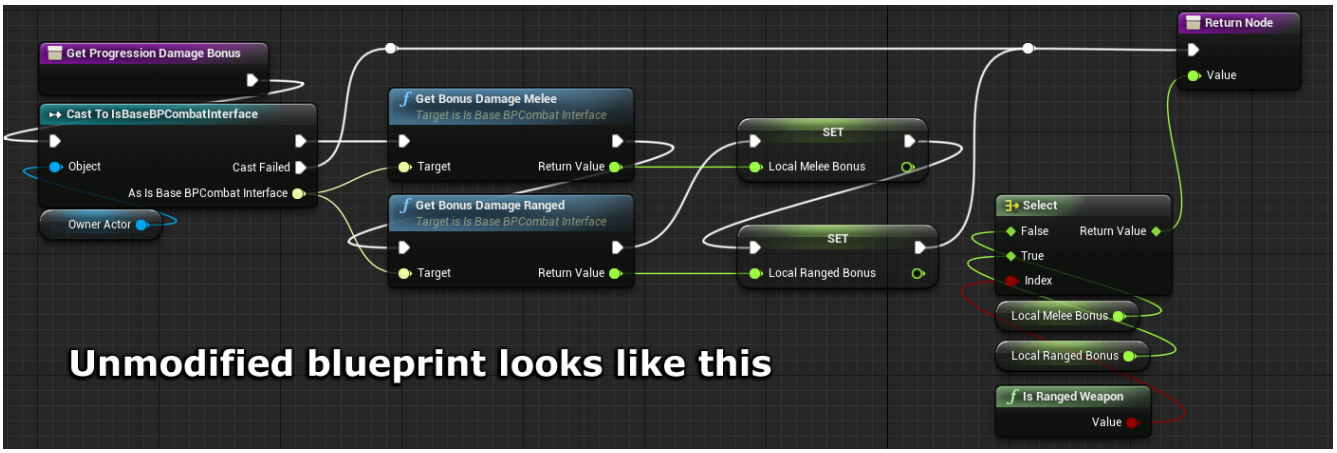
No, just kidding, this is the one:

Get Progression Damage Bonus

In here you can completely override the damage bonus that would normally apply through Strengh, and instead you may want to apply bonus damage if the player is wearing heavy armor, or maybe you want your weapon to get bonus damage based on level, or Agility or Survival.

Keep in mind that the stat displayed in the little information box when you click a weapon will display whatever you put into the Item Table, so you can in fact, use this function to implement a sneaky extra damage on top of that - and you may of course also just get rid of it completely by overriding it to return 0 bonus.

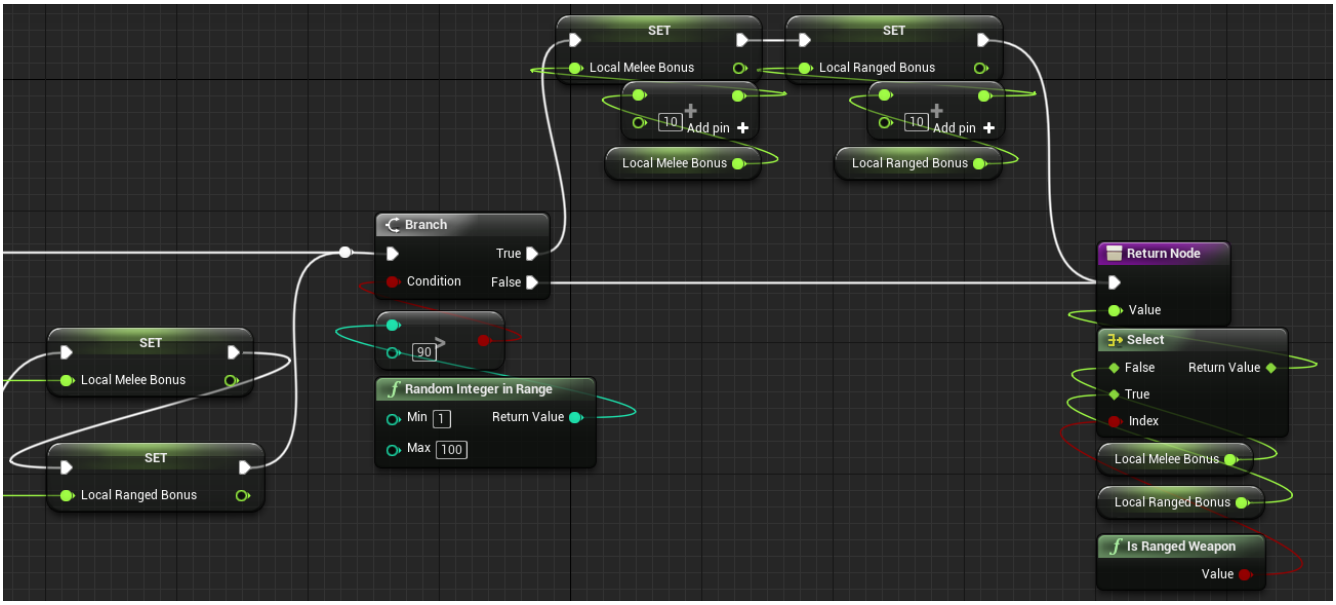
Let's run through a few examples in the appendix - first though, this is what the original logic looks like:



Appendix - Progression Damage Bonus Examples

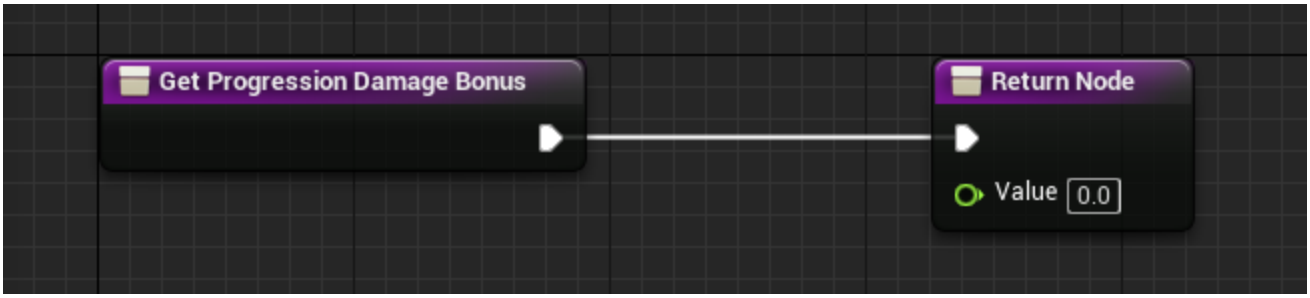
Example 1 - Adding a critical hit

For this we simply need to add a few nodes between the Set Local Ranged Bonus and the Return node. Something like this:



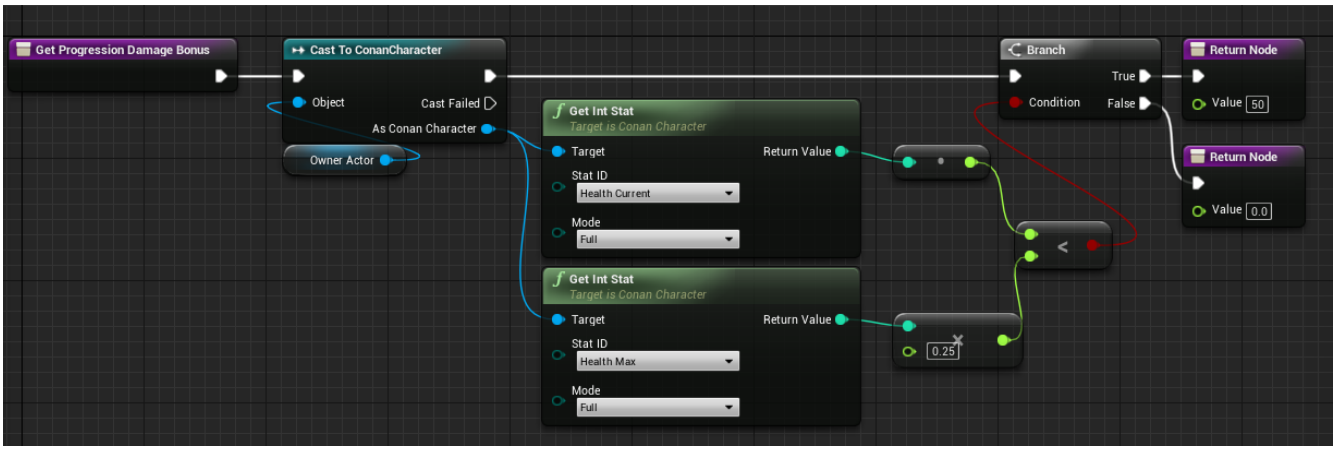
Example 2 - Nullifying the bonus completely

This is very simple - essentially, you just need to remove the logic and make the function look like this:



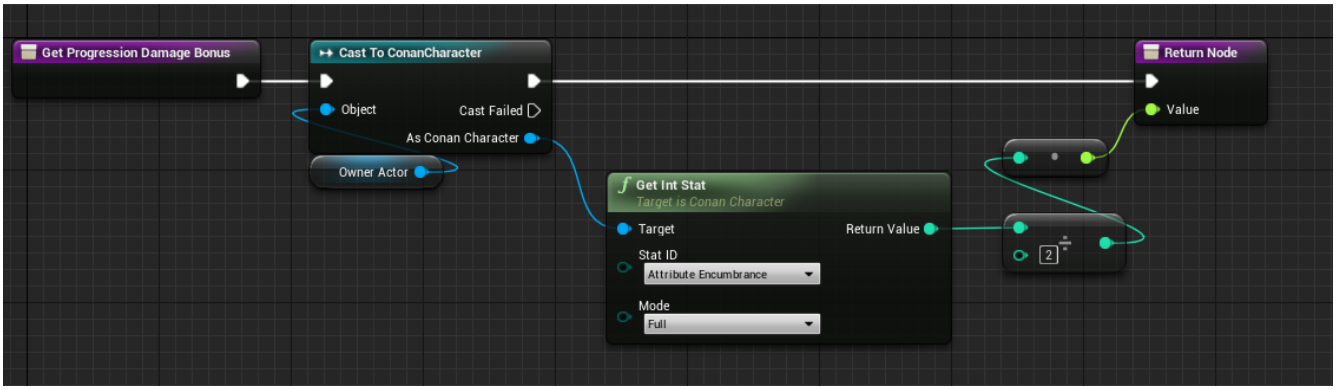
Example 3 - Increase damage if user is below 25% health

If you want to access the player, you can use the "Owner Actor" and cast to ConanCharacter to fetch INT and Float stats. You can do quite a lot with these. Now - for the example below, do note that we've removed the regular bonus damage as well (in order to fit the blueprint on screen) so if you also want the regular bonus damage, you'll need to implement this logic towards the end of the chain.



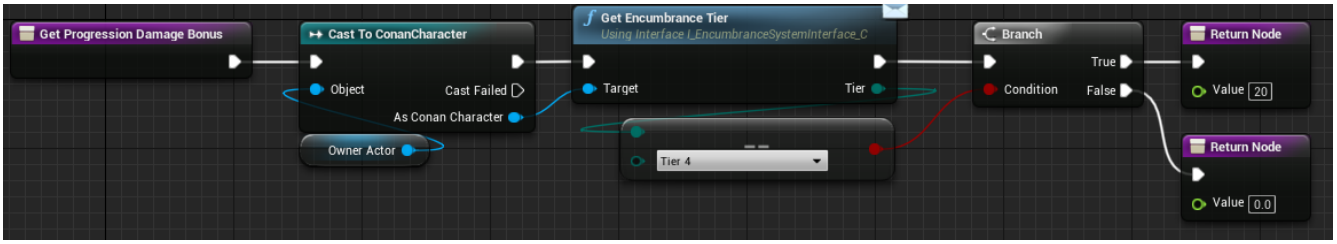
Example 4 - Use Attributes to increase damage

You can, as mentioned before, completely disengage Strength from the bonus damage - perhaps you want to use a weapon that scales using Accuracy, Agility or Encumbrance? You can even scale it based on the players gender if you like.



Example 5 - Only increase damage for non-encumbered players

Because - why not?

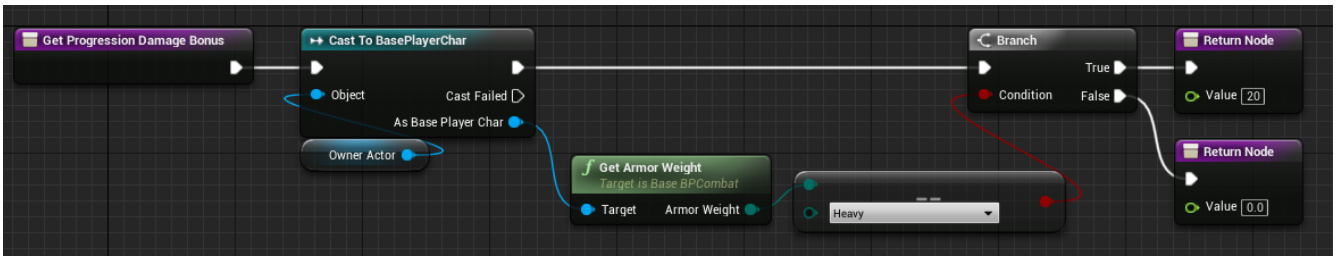


Example 6 - Increase damage if player is wearing heavy armor

Eagle-eyed readers will notice that in the example below, we are doing a cast to BasePlayerChar.

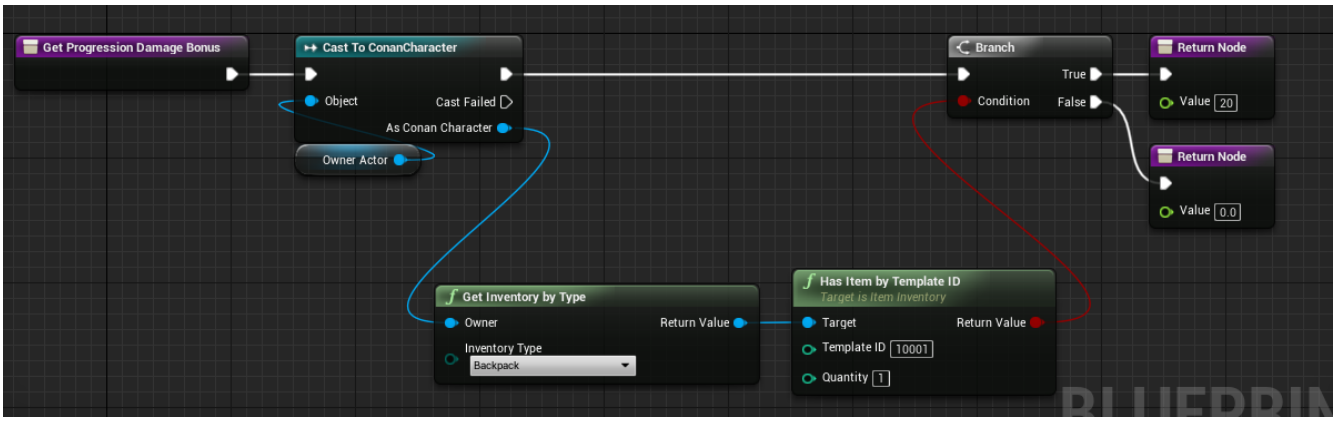
This is an example, so we'll let this slide here - however - you should ideally store the BasePlayerChar as a variable in the blueprint and hook it into the OnEquip event and merely reference it here, otherwise, we'll be casting every time we hit, which can slow down the client and server.

So - keep that in mind.



Example 7 - Increase damage if player carries a specific item

Suffice to say that there are a lot of different modifications you can do inside this function.



Example 8 - Increase damage only for Cimmerians

Ok, this is where we'll stop with the examples, since one could effectively go through every single stat on the player to check them. Do you want red-headed females with Ymir as their god to do extra damage on Sundays if the weather is rainy? I don't see why not!