Table of Contents

About this primer

In this primer, we'll go through the concept of a weighted table, or to give it it's full name; a "weighted chance" table. We will only briefly touch on parts of the game where it is implemented, such as the SpawnTables and the LootTables but this primer is geared more towards understanding the tables themselves and what they can do.

Uses of Weighted Tables in Conan Exiles
The two primary uses for weighted tables in Conan Exiles are the Spawning System and the Loot-Tables system. Any loot-table in the game as well as the WeightedSpawnTable uses Weighted Tables.

What are weighted tables?

Many may look at a weighted table and assume it's merely a percentage-based random-chance table. While this may be the case sometimes, it's not necessarily true. A pure "chance" table would need to have all the numbers add up to 100% whereas a weighted table doesn't work like that.

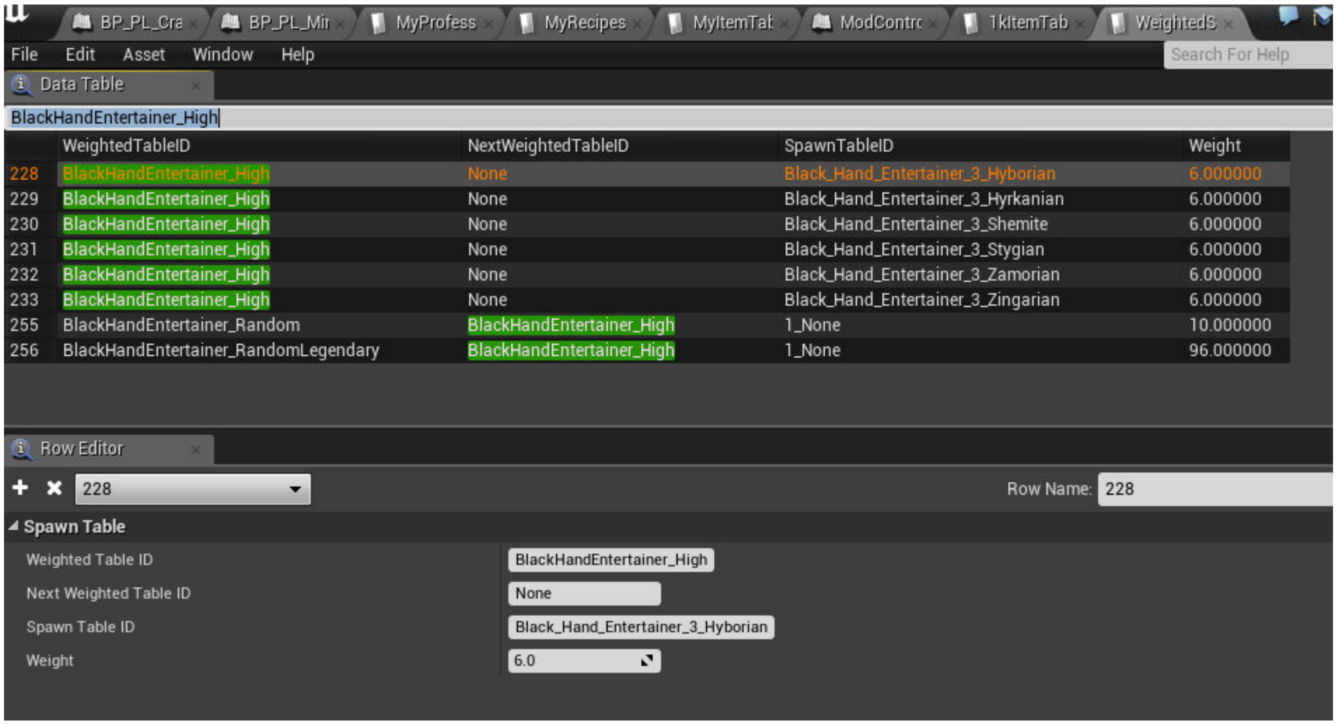
Note that many of the tables in Conan Exiles may actually add up to 100 but that is more out of convenience than necessity.

A weighted table allows you to assign a specific weight (or significance) to an entry. When the table is accessed, the system them pours all those weights into a single pool and randomly selects between them.

This makes it very easy to add new entries to such tables as you don't necessarily need to change the already existing values. If you want to add another entry, simply make a new table, attach it with the mod-controller, and assign a new weight to your entry, and any attempt to call said entry will randomly select yours.

For an example, please see the next page

Example #1 - WeightedSpawnTableRow



	WeightedTableID	NextWeightedTableID	SpawnTableID	Weight
228	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Hyborian	6.000000
229	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Hyrkanian	6.000000
230	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Shemite	6.000000
231	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Stygian	6.000000
232	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Zamorian	6.000000
233	BlackHandEntertainer_High	None	Black_Hand_Entertainer_3_Zingarian	6.000000
255	BlackHandEntertainer_Random	BlackHandEntertainer_High	1_None	10.000000
256	BlackHandEntertainer_RandomLegendary	BlackHandEntertainer_High	1_None	96.000000

228 Row Name: 228

Weighted Table ID	BlackHandEntertainer_High
Next Weighted Table ID	None
Spawn Table ID	Black_Hand_Entertainer_3_Hyborian
Weight	6.0

(Thanks to GUTTA for this image)

In this example, you can see that there are multiple entries of the "BlackHandEntertainer_High" and that they point to different specific SpawnTableID's. Also note that all these have the exact same weight (6 in this case). This means that there is an equal chance of spawnening either of these entries when the "BlackHandEntertainer_High" is called. *(Thanks to GUTTA for this image)*

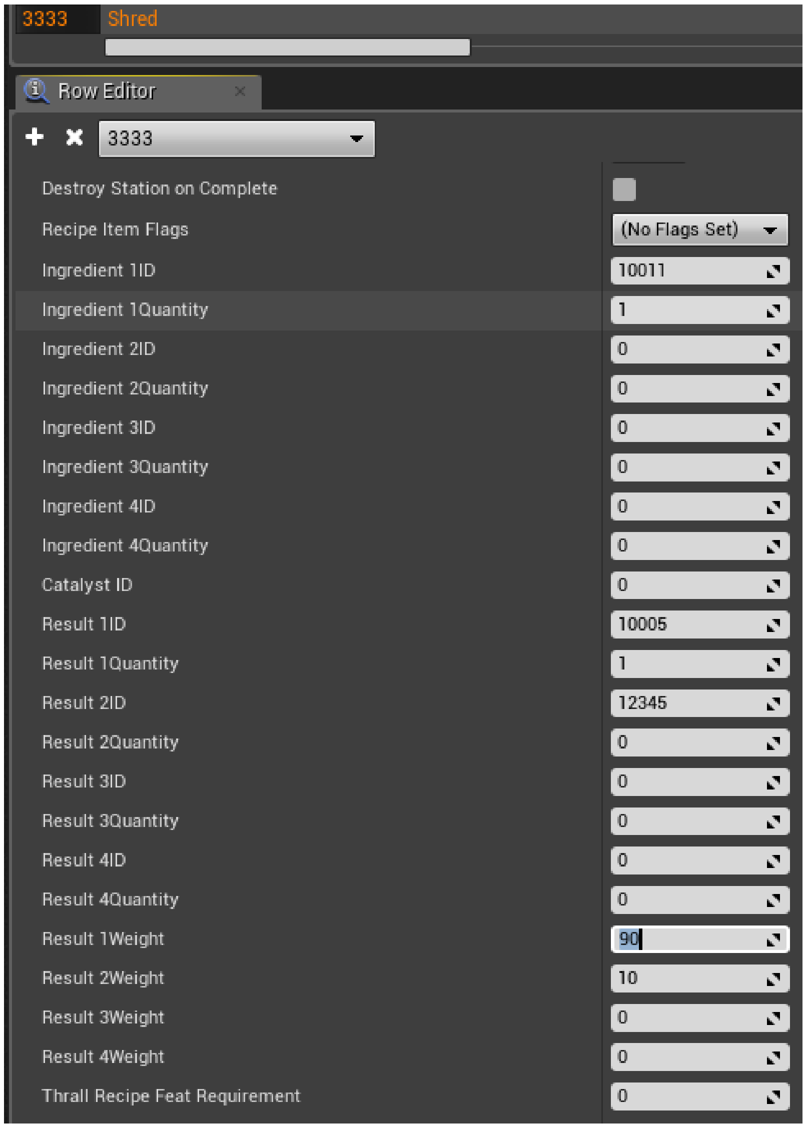
Mathematically, in this case, each entry has a 6 in (6+6+6+6+6+6) = 36 chance of spawning. If we changed the Black_Hand_Entertainer_3_Shemite to be 1000, all other entries would now have a 6 in (6+6+6+6+6+1000) =1030 chance of spawning, whereas the Shemite would have a 1000/1030 chance of spawning (in other words, a 97.08737%...(etc) chance of spawning.

If you instead ADDED a new entry with the same value (6) as all the others, you would now seamlessly add a new entry to the list with the exact same spawn chance as the rest of the entries.

Example #2 - RecipesTable

In the recipes-table there are random outcomes - these are dependant on weights as well, and works as follows:

Let's say that you have made a new recipe called "Shred "that turns "Wood" into "Bark" and "Coarse Powder". You want the ingredient you put in (Wood) to sometimes produce Bark, and sometimes produce Coarse Powder. The table could look something like this:



3333 Shred

Row Editor

3333

Destroy Station on Complete	☐
Recipe Item Flags	(No Flags Set)
Ingredient 1ID	10011
Ingredient 1Quantity	1
Ingredient 2ID	0
Ingredient 2Quantity	0
Ingredient 3ID	0
Ingredient 3Quantity	0
Ingredient 4ID	0
Ingredient 4Quantity	0
Catalyst ID	0
Result 1ID	10005
Result 1Quantity	1
Result 2ID	12345
Result 2Quantity	0
Result 3ID	0
Result 3Quantity	0
Result 4ID	0
Result 4Quantity	0
Result 1Weight	90
Result 2Weight	10
Result 3Weight	0
Result 4Weight	0
Thrall Recipe Feat Requirement	0

As you can see, the Ingredient is Wood (ID 10011) and the Result1ID is Bark. Result2ID is our new resource "Coarse Powder".

For weights, we have put down 90 for ResultItem1, and 10 for ResultItem2.

This means that any craft of this recipe has a 90/100 (90%) of spawning ResultItem1, and a 10/100 (10%) of spawning ResultItem2.

Add all the weights together, and you end up with a number (in this case, 90+10=100). Then take the actual weight of each entry and divide it by the number you have and you will get the actual percentage chance you have of spawning that entry.

As an example, if we had put Result1Weight to be "177" and ResultWeight2 to be "853", it would work out like this:

177+853 = 1030

177/1030 = 0.1718..... = 17%
853/1080 = 0.8281.... = 83%

Nested tables - a quick note

It's worth noting that many tables that use this method in Conan Exiles also use Nested Tables. In the example above, you may have noticed the "NextWeightedTableID" - if you have an entry here pointing to another weighted table, you can make quite extensive hierarchies for tables. You can see this happening in the Spawn Table if you look below the marked green rows in the image, where the "BlackHandEntertainer_Random" and ""BlackHandEntertainer_RandomLegendary" points to other sections of this table as well.