

Table 1: Outlets for UAffinitiesComponent

TODO: to do!
--------------

• • •

Table 2: Outlets for ULevelComponent

GetBaseExpYield	
► Before	<code>const float OriginalYield,</code> <code>float&amp; ReturnedYield</code>
► After	<code>const float OriginalYield,</code> <code>const float ReturnedYield</code>
GetCXP	
► Before	<code>const uint32 OriginalCXP,</code> <code>int32&amp; ReturnedCXP</code>
<i>Note:</i>	ReturnedCXP is <code>int32&amp;</code> instead of <code>uint32&amp;</code> for Blueprint compatability.
► After	<code>const uint32 OriginalCXP</code> <code>const int32 ReturnedCXP</code>
<i>Note:</i>	ReturnedCXP is <code>const int32</code> instead of <code>const uint32</code> for Blueprint compatability.

Continued on next page

Table 2: Outlets for ULevelComponent (Continued)

GetExpYield	
► Before	<pre>const float OriginalYield, float&amp; ReturnedYield, const uint16 DefeatedLevel, const uint16 VictoriousLevel</pre>
<i>Note:</i>	<p>“Defeated” and “Victorious” levels are provided for flexibility (e.g., in case you want to yield exp differently based on level difference, although technically you could always back-calculate the level difference based on the equation and <code>OriginalYield</code>).</p>
► After	<pre>const float OriginalYield, const float ReturnedYield, const uint16 DefeatedLevel, const uint16 VictoriousLevel</pre>
<i>Note:</i>	<p>“Defeated” and “Victorious” levels are provided for symmetry with respect to the <code>Before</code> delegate (since <code>ReturnedValue</code> is already calculated, I can’t think of why you would need them, but you never know!).</p>
GetMaxLevel	
► Before	<pre>const uint16 DefaultMax, int32&amp; AttemptedMax</pre>
<i>Note:</i>	<p><code>DefaultMax</code> is defined in the code. It should normally be 100, but may change for certain subclasses (e.g., a <code>UBossLevelComponent</code> may have a max of 200 instead).</p> <p><code>AttemptedMax</code> is <code>int32&amp;</code> instead of <code>uint16&amp;</code> for Blueprint compatability.</p>
► After	<pre>const uint16 DefaultMax const int32 ReturnedMax</pre>

Continued on next page

Table 2: Outlets for ULevelComponent (Continued)

GetMinLevel	
► Before	<code>const uint16 DefaultMin, int32&amp; AttemptedMin</code>
<i>Note:</i>	<code>DefaultMin</code> is defined in the code. It should normally be 1, but may change for certain subclasses (e.g., a <code>UEggLevelComponent</code> may have a min of 0 instead for whatever reason). <code>AttemptedMin</code> is <code>int32&amp;</code> instead of <code>uint16&amp;</code> for Blueprint compatability.
► After	<code>const uint16 DefaultMin const int32 ReturnedMin</code>
<i>Note:</i>	<code>ReturnedCXP</code> is <code>const int32</code> instead of <code>const uint32</code> for Blueprint compatability.
GetBaseExpYield	
► Before	<code>const float OriginalYield, float&amp; ReturnedYield</code>
► After	<code>const float OriginalYield, const float ReturnedYield</code>
SetBaseExpYield	
► Before	<code>const float OldYield, float&amp; AttemptedYield</code>
► After	<code>const float OldYield const float NewYield</code>
SetCXP	
► Before	<code>const uint32 OldCXP, int32&amp; AttemptedCXP</code>
<i>Note:</i>	<code>AttemptedCXP</code> is <code>int32&amp;</code> instead of <code>uint32&amp;</code> for Blueprint compatability.
► After	<code>const uint32 OldCXP const uint32 NewCXP</code>
<i>Note:</i>	<code>UStatsComponent</code> subscribes to this in order to change stats on level change.

Table 3: **Outlets** for **UStatsComponent**

<b>ModifyStat</b>	
► Before	<pre>const EStatEnum TargetStat, const EStatValueType ValueType, const EModificationMode Mode, const float OriginalValue, float&amp; AttemptedValue</pre>
► After	<pre>const EStatEnum TargetStat, const EStatValueType ValueType, const EModificationMode Mode, const float OriginalValue, const float NewValue</pre>
<i>Note:</i>	All “ModifyStat” functions from <b>UStatsComponent</b> (such as <b>ModifyStatsUniformly</b> or <b>RandomizeStats</b> ) go through <b>ModifyStatInternal</b> , which calls this <b>Outlet</b> .

  

<b>RandomizeStats</b>	
► Before	<pre>const EStatEnum TargetStat, const FStatRandParams OriginalParams, FStatRandParams&amp; ParamsToBeUsed</pre>
► After	<pre>const EStatEnum TargetStat, const FStatRandParams OriginalParams, const FStatRandParams UsedParams</pre>
<i>Note:</i>	The <b>EStatEnum</b> is not the acutal <b>FStat</b> . To get the <b>FStat</b> (such as <b>FHealth</b> ), use <b>UStatsComponent::GetStat(EStatEnum)</b>

Continued on next page

Table 3: Outlets for UStatsComponent (Continued)

RecalculateStats	
► Before	<pre>const EStatEnum TargetStat, const bool bResetCurrent, const float OriginalCurrent, const float OriginalPermanent</pre>
► After	<pre>const EStatEnum TargetStat, const bool bResetCurrent, const float OriginalCurrent, const float OriginalPermanent</pre>