

`UDependentEffectComponents` add common, cross-class functionality to other effects. Any `UDependentEffectComponent`:

- has an `Owner` that it's dependent upon.
- `should` be added just after its `Owner` is added. See how to attach below.
- `should` be destroyed just before its `Owner` is destroyed. This should happen via `destroy` and `unregister`. See below.
- follows its `Owner`'s `Silence` and `Unsilence`.
- returns its `Owner`'s `Priority`, `Stacks`, `MaxStats`, `Priority`, and `ShouldApplyEffect`.
- is not visible to UI.

0.0.1 Concrete Example

For example, both `UBerserkerGene` (a `UMutation`) and `UFullBloom` (a `UPositiveAura`) modify stats. Despite being different inheritance hierarchically, they do pretty similar things. They utilize `UPermStatMod`. For this example, let's use

```
UEffectComponent → UBerserkerGene
                  ↳ +15%PhA | -10% PhD | -10%SpD
UDependentEffectComponent → UPermStatMod
```

To attach `UPermStatMod` to `UBerserkerGene`, add the following to `UBerserkerGene::OnComponentCreated`:

```
1 Super::OnComponentCreated();
2 ADD_COMPONENT(UPermStatMod, PermStatMod, GetOwner())
3 PermStatMod->SetOwner(this);
4 # Do some customization here, such as setting Dependent variables
```

You can also initialize things on the `UPermStatMod` side of things. For example, if you wanted to make sure there were stats attached to the new owner:

```
1 // Careful! Squirrels everywhere...
2 if (NewOwner != nullptr)
3 {
4
5     // Get StatsComponent
6     SEARCH_FOR_COMPONENT_OR_DESTROY(UCombatStatsComponent, StatsComponent,
7         NewOwner->GetOwner(), true)
8
9     // No stats component?
10    if (StatsComponent == nullptr)
11    {
12        UE_LOG(LogTemp, Warning, TEXT("No UCombatStatsComponent found for PermStatMod!
13            This is required *before* the Owner is set.))
14        return;
15    }
16 }
```

15
16
17

```
// This also calls ApplyEffect  
Super::SetOwner(NewOwner);
```

Finally, when you destroy `UBerserkerGene`, make sure you also destroy `UPermStatMod`:

1
2
3
4
5

```
void UBerserkerGene::OnComponentDestroyed(bool bDestroyingHierarchy)  
{  
    PermStatMod->DestroyComponent();  
    Super::OnComponentDestroyed(bDestroyingHierarchy);  
}
```

Table 1: All `UDependentComponents` currently implemented and tested.

Dependent	Short Description	Implemented via	Priority	Note
PermStatMod	Modifies according to its <code>TArray</code> of <code>FStatMods</code> .	<code>AfterRecalculateStats</code>	<i>Dependent</i>	See, e.g., <code>UBerserkerGene</code> .