UDependentEffectComponents add common, cross-class functionality to other effects. For example, both UBerserkerGene (a UMutation) and UFullBloom (a UPositiveAura) modify stats. Despite being different inheritance hierarchically, they do pretty similar things. UDependentEffectComponent:

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

To attach a UDependentEffectComponent to another UEffectComponent, add the following to UEffectComponent::OnComponentCreated:

```
Super::OnComponentCreated();
ADD_COMPONENT(UDependentEffectComponent, Dependent, GetOwner())
Dependent->SetOwner(this);
# Do some customization here, such as setting Dependent variables
```

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

```
// Careful! Squirrels everywhere...
if (NewOwner != nullptr)
{

   // Get StatsComponent
   SEARCH_FOR_COMPONENT_OR_DESTROY(UCombatStatsComponent, StatsComponent,
      NewOwner->GetOwner(), true)

   // No stats component?
   if (StatsComponent == nullptr)
   {
      UE_LOG(LogTemp, Warning, TEXT("No UCombatStatsComponent found for PermStatMod!
         This is required *before* the Owner is set."))
      return;
   }
}

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

Table 1: All `UDependentEffectComponent`s currently implemented and tested.

DependentEffect	Short Description	Implemented via	Priority	Note
PermStatMod	Modifies according to its `TArray` of `FStatMod`s.	`AfterRecalculateStats`	*Dependent*	See, e.g., `UBerserkerGene`.