

Blueprint Nativization Support

 **Blueprint nativization** converts a virtualized blueprint code of the node to its C++ representation, that is compiled ahead of time instead of being interpreted at a runtime. For complex scenarios it can run a lot faster as C++ compilers are very intricate on optimizations.

Unfortunately, Unreal Engine doesn't support nativization of the functions that perform *custom* blueprint virtual machine manipulations. Apparatus is quite a low-level tool with a bunch of functions exactly like that, so when you mark your Mechanic as nativized and then package a build you may get something like that in the console

```
UATHelper: Packaging (Windows (64-bit)):  
.../Intermediate/Plugins/NativizedAssets/Windows/Game/Source/NativizedAssets  
/Private/...cpp(657): error C3861: 'GetChunkChainTraitAt': identifier not  
found  
UATHelper: Packaging (Windows (64-bit)):  
.../Intermediate/Plugins/NativizedAssets/Windows/Game/Source/NativizedAssets  
/Private/...cpp(783): error C2039: 'SetSubjectTrait': is not a member of  
'FCustomThunkTemplates'
```

Luckily there is a workaround that is quite easy to use.

You can actually and pretty painlessly change a single header file inside an Unreal Folder. It's located at a path like so: C:\Program Files\Epic Games\UE_4.26\Engine\Source\Runtime\Engine\Public\GeneratedCodeHelpers.h

Open this file in your preferred text editor and add the following code at the end of the FCustomThunkTemplates struct. As of version 4.26 the struct is declared at line 89. You should add the following snippet after the public: access specifier of the struct.

```
#pragma region Apparatus Nativization Support  
  
template <typename TSubjectHandle>  
static void  
SetSubjectTrait(TSubjectHandle& SubjectHandle,  
                UScriptStruct* TraitType,  
                void* InTraitData)  
{  
    SubjectHandle.SetTrait(TraitType, InTraitData);  
}  
  
template <typename TSubjectHandle, typename T>  
static void  
SetSubjectTrait(TSubjectHandle& SubjectHandle,  
                UScriptStruct* TraitType,  
                const T& InTraitData)
```

```
{
    SubjectHandle.SetTrait(TraitType, (const void* const)&InTraitData);
}

template <typename TSubjectHandle>
static void
ObtainSubjectTrait(TSubjectHandle& SubjectHandle,
                  UScriptStruct* TraitType,
                  void* const OutTraitData)
{
    SubjectHandle.ObtainTrait(TraitType, OutTraitData);
}

template <typename TSubjectHandle, typename T>
static void
ObtainSubjectTrait(TSubjectHandle& SubjectHandle,
                  UScriptStruct* TraitType,
                  T* OutTraitData)
{
    SubjectHandle.ObtainTrait(TraitType, (void* const)&OutTraitData);
}

template <typename TSubjectHandle>
static void
GetSubjectTrait(TSubjectHandle& SubjectHandle,
               UScriptStruct* TraitType,
               void* const OutTraitData)
{
    SubjectHandle.GetTrait(TraitType, OutTraitData);
}

template <typename TSubjectHandle, typename T>
static void
GetSubjectTrait(TSubjectHandle& SubjectHandle,
               UScriptStruct* TraitType,
               T& OutTraitData)
{
    SubjectHandle.GetTrait(TraitType, (void* const)&OutTraitData);
}

template <typename T>
static void
GetChunkChainTraitAt(const int32& ChunkChainId,
                    const int32 TraitIndex,
                    T& OutTraitData)
{
    ensure(UMachine::ObtainChunkChain(ChunkChainId).GetTraitAt(TraitIndex,
        (void* const)&OutTraitData)
        >= EApparatusStatus::Success);
}
```

```
template <typename T>
static void
GetChunkChainTraitAt(const int32& ChunkChainId,
                    const int32 TraitIndex,
                    T* const OutTraitData)
{
ensure(UMachine::ObtainChunkChain(ChunkChainId).GetTraitAt(TraitIndex,
( void* const)OutTraitData)
      >= EApparatusStatus::Success);
}

#pragma endregion Apparatus Nativization Support
```

From:

<http://turbanov.ru/wiki/> - **Turbopedia**

Permanent link:

<http://turbanov.ru/wiki/en/toolworks/docs/apparatus/blueprint-nativization?rev=1622573721>

Last update: **2021/06/01 18:55**

