NPlayer Template 문서

**개요**

: Unreal Engine 4(UE4)의 In Camera VFX Template을 Vive Studios에서 사용 할 수 있도록 만든 Template Project.

In Camera VFX Template Project는 C++ Project를 지원 하지 않으나 NPlayer Template Project는 C++로 개발 할 수 있다. 그래서 Vive Studios에서 개발한 기술들을 UE4의 In Camera VFX에 적용이 가능하다.

목차

[NPlayer Template 문서 1](#_Toc66196307)

[**UE4 In Camera VFX에 필요한 Plugins** 1](#_Toc66196308)

[**NPlayer Template Project 설정** 2](#_Toc66196309)

[**NPlayerStagePawn 사용** 8](#_Toc66196310)

[**NPlayer Template 실행** 9](#_Toc66196311)

[**UE4 In Camera 최적화 옵션** 12](#_Toc66196312)

[**UE4 In Camera VFX에서 Multi-User Editing 설정** 14](#_Toc66196313)

[**Multi GPU 활용 하기** 17](#_Toc66196314)

[**참고** 17](#_Toc66196315)

## **UE4 In Camera VFX에 필요한 Plugins**

UE4 In Camera VFX에 필요한 Plugins은 다음과 같다.

**◦ 필수 항목:**

**nDisplay** : UE4에서 다중 디스플레이에 렌더링하기 위한 기능.

**VirtualProductionUtilities** : Virtual Production에 유용한 유틸리티 기능을 제공하는 플러그인

**MediaFrameworkUtilities** : SDI 캡처 카드의 라이브 비디오, 타임 코드 및 젠록과 관련된 유틸리티 기능을 제공하는 플러그인.

**AjaMedia** : Aja SDI 캡처 카드에 대한 지원 제공.

**BlackmagicMedia** : Blackmagic SDI 캡처 카드에 대한 지원 제공.

**MultiUserClient** : UE4에서 여러 편집자가 공유 세션에 참가 하여 같이 편집 하는 기능.

**LiveLink** : UE4에서 모션 캡쳐 및 카메라 Tracking과 같은 라이브 데이터를 수집하기 위한 기능.

**LiveLinkOverNDisplay** : Primary 노드는 Live Link 데이터를 수신하고, 효율적이으로 동기화 되도록 수신된 데이터를 재분배 하기 위한 기능.

**◦ 옵션 항목:**

**SequencerScripting**

**OSC**

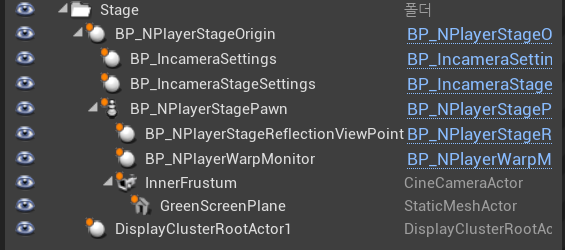
**RemoteControl**

## **NPlayer Template Project 설정**

: NPlayer Template는 기존 In Camera VFX와 다른점은 NPlayer Template 전용 Pawn을 구현하여 입력 처리를 Custom화 할 수 있다.

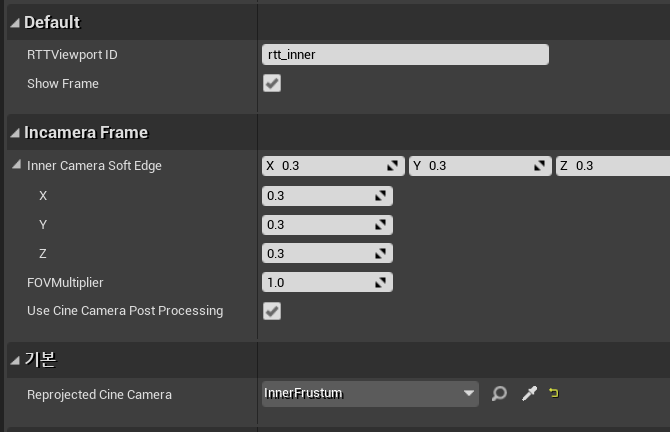
전용 Pawn은 C++로 구현되어 있어 C++ API 기능을 Coding을 통해 Pawn에 구현하여 통합 할 수 있다..

**◦ NPlayer Template Stage Hierarchy 구조**



**◦ BP\_NPlayerStageOrigin Actor** : Stage Hierarchy의 Root 역할을 하며 이 Root를 이동하며 그 하위에 존재하는 Actor들이 월드상에서 동일한 형태를 유지하며 이동 시킬 수 있다.

**◦ BP\_IncameraSettings Actor** : UE4 nDisplay에서 제공하는 Actor로 내부에서 사용할 Frustum Camera를 설정 한다. 이 Frustum Camera는 Detail 패널의 Default 항목에 있는 Reprojected Cine Camera 속성에 지정한다. 이 Actor는 Frustum Camera의 블렌딩 영역과 Camera 프레임 섹션에 있는 FOV 승수를 결정 할 수 있다.



RTTViewport ID : nDisplay Config 파일에서 Frustum Camera Viewport 설정 항목에서 사용할 ID에 적용.

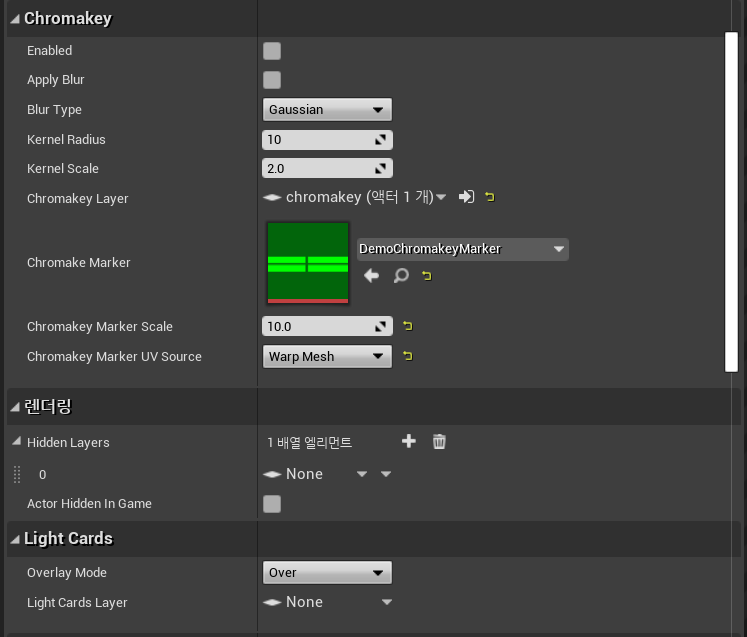
Inner Camera Soft Edge: Frustum의 외곽을 Bluring 할 영역 크기 지정

**(X,Y,Z 전부 0이면 Soft Edge 비활성화 됨)**

FOVMultiplier : FOV 승수 지정.

**◦ BP\_IncameraStageSettings Actor** : UE4 nDisplay에서 제공하는 Actor로 기본 크로마키 마커를 프로젝트에 제공하기 위해 Detail 패널의 크로마키 항목에 크로마키 어셋 텍스처를 변경 할 수 있다.

렌더링 항목에서 LED 벽에 표시 되지 않도록 Hidden Layers 배열에 오브젝트를 추가 할 수 있고, 라이트 카드 항목에 라이트 카드 레이어를 지정 할 수 있다. 이 라이트 카드는 LED Stage에 조명을 제공하기 위해 외부 절두체에서만 볼 수 있다.



Enabled : 크로마키 활성화 여부.

Chromakey Layer : 크로마키 영역으로 캡처할 오브젝트가 나열된 레이어 등록.

Chromakey Marker : 크로마키 영역에 표시 될 크로마키 어셋 텍스처.

Chromakey Marker Scale : 크로마키 영역에 표시 될 마크 크기 지정.

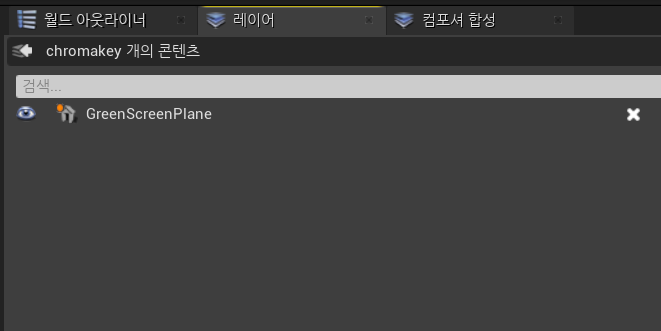
Chromakey Marker UV Source : 크로마키 영역에 표시 될 마크 UV 방식 지정.

옵션으로 지정할 만한 설정:

Hidden Layers: Hidden 레이어 설정.

Light Cards Layer: 라이트 카드 레이어 설정.

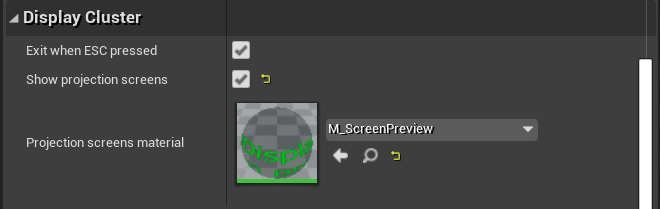
**◦ Chromark Layer 지정 샘플:**



NPlayer Template에서는 Green색으로 지정된 Plane인 GreenScreenPlane Actor를 추가 한다.

**◦ DisplayClusterRootActor** : 이 Actor의 위치는 외부 Frustum이 투영되는 원근을 정의 한다.

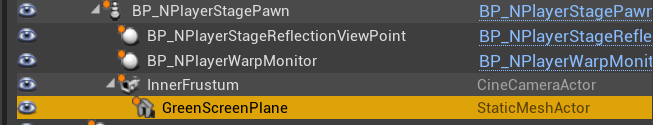
만일 BP\_NPlayerStageReflectionViewPoint Actor를 배치 했다면, 외부 Frustum이 투영되는 원근 정의는 BP\_NPlayerStageReflectionViewPoint Actor의 위치를 기반으로 정의 된다.



Show projection screens 항목을 Check 해주고, 옵션으로 Projection screen material를 설정 해 줄 수도 있다.

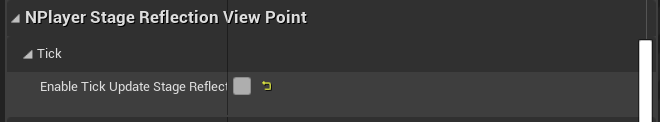
**◦ InnerFrustum (Cine Camera Actor)** : 실제 촬영하는 카메라의 Tracking 정보를 입력 받아 내부 Frustum 투영 효과를 만드는 Camera Actor.(실제 In Camera VFX 기능을 한다)

이 InnerFrustum 아래에 크로마키 할 영역을 만들어 주는 Green Screen Plane인 Static Mesh Actor를 놓아 둔다.



**◦ BP\_NPlayerStageReflectionViewPoint Actor** : LED 화면에 투영하기 위한 카메라 위치를 설정하는 Actor.

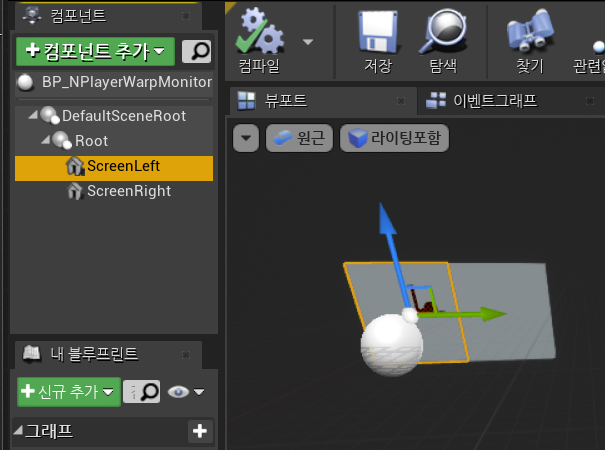
NPlayer Template에서는 NPlayerStagePawn에서 LED 화면 투영 카메라 위치를 업데이트 할 수 있도록 하기 위한 기능이 추가 되었다.



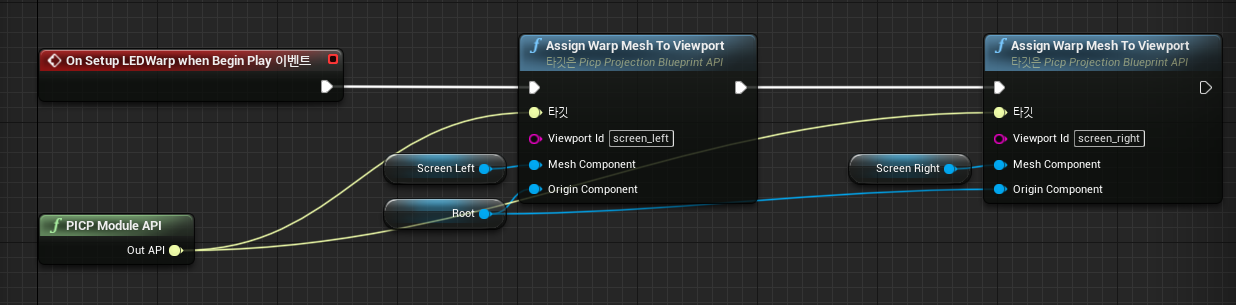
Enable Tick Update Stage Reflection View Point를 활성화 하면 Tick 함수에서 카메라 위치를 업데이트 하고, 비활성화 하면 NPlayerStagePawn의 Tick 함수에서 카메라 위치를 업데이트 한다.

**◦ BP\_NPlayerWarpMonitor / BP\_NPlayerWarpMonitorLevel Actor** : LED 화면에 투영할 카메라 Viewport 크기 및 영역을 설정 할 때 사용하는 Static Mesh를 nDisplay Picp Mesh Module에 등록하는 기능을 담당하는 Actor.

BP\_NPlayerWarpMonitor는 Static Mesh Component를 Actor 안에 직접 추가하여 BeginPlay 이벤트에서 nDisplay Picp Mesh Module에 등록하는 방식으로 작동 한다.

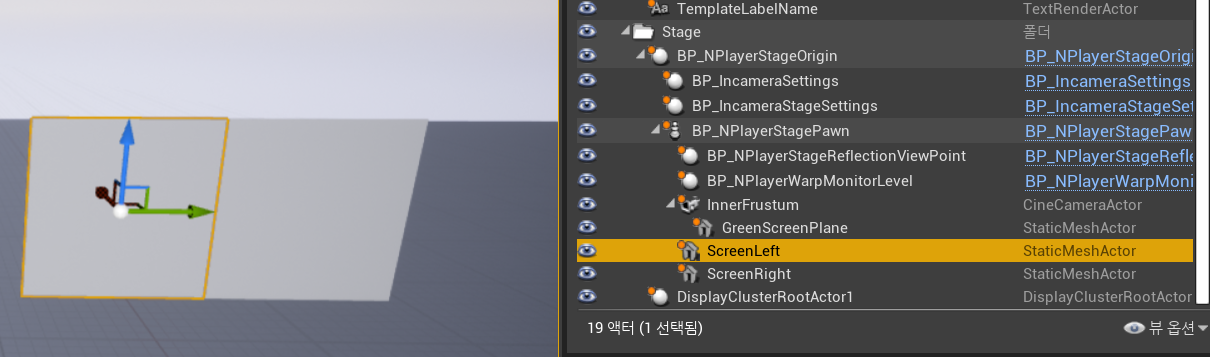


<Actor에 StaticMeshComponent를 추가한 모습>

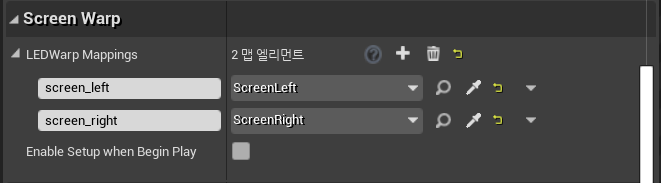


<OnSetupLEDWarpWhenBeginPlay 이벤트에서 Picp Mesh Module에 등록>

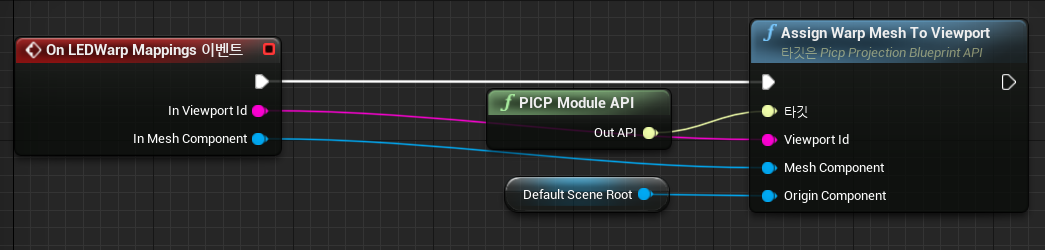
BP\_NPlayerWarpMonitorLevel은 World상에 배치된 Static Mesh Actor들을 참조하여 실행시 nDisplay Picp Mesh Module에 등록하는 방식을 작동 한다.



<World에 배치된 Static Mesh Actor>



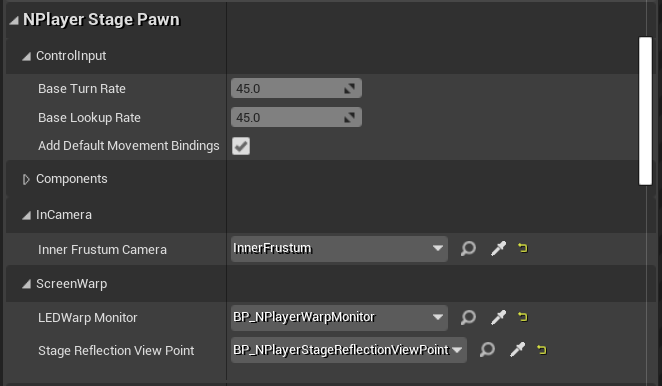
<LEDWarp Mappings 목록에 World에 배치된 Static Mesh Actor들을 등록>



<OnLEDWarpMappings 이벤트에서 Picp Mesh Module에 등록>

BP\_NPlayerWarpMonitorLevel Actor를 World에 배치 할 때 주의 해야 할 점은 Actor 위치가 Warp Mesh(Group)의 중심을 바라보는 위치에 배치 해 줘야 한다. Warp Mesh(Group)의 중심을 바라 보는 위치가 아니라면, 실행시 Projection 되는 화면 영상이 틀어져 보이게 된다.

**◦ BP\_NPlayerStagePawn Actor** : NPlayer Template 전용 Pawn Actor.



Inner Frustum Camera : World에 배치된 InnerFrustum Cine Camera Actor를 등록 한다.

LEDWarp Monitor : World에 배치된 BP\_NPlayerWarpMonitor Actor를 등록 한다.

Stage Reflection View Point : World에 배치된 BP\_NPlayerStageReflectionViewPoint Actor를 등록 한다.

## **NPlayerStagePawn 사용**

**◦ NPlayerStagePawn / InnerFrustum 조작**

: UE4에서 기본 정의된 Key Bind로 이동이 가능하고, 마우스를 통해 Yaw, Pitch 회전이 가능 하다.

UE4에서 기본 정의된 Key Bind 대신 새로 Key Bind를 정의해서 사용 할 수도 있다.

기본 정의된 Key Bind:

**W/S: 전후 이동**

**A/D: 좌우 이동**

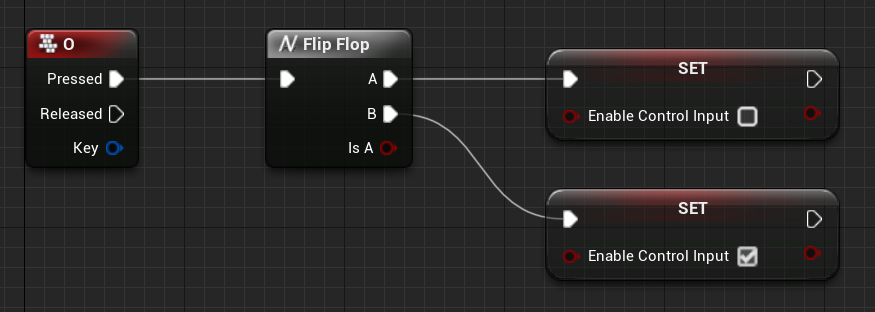
**E/C: 위아래 이동**

**마우스: Yaw, Pitch 회전**

NPlayerStagePawn의 ControlInput 항목에서 **Add Default Movement Bindings**를 체크 해제 하면 기본 Key Bind를 사용하지 않고, 새로 Key Bind를 정의 할 수 있다.

**◦ NPlayerStagePawn 입력 가능여부 설정**

실행 중 NPlayerStagePawn 이동과 회전 입력이 되지 않도록 할 수 있는데, O Key를 누르면 된다. 다시 이동과 회전 입력이 되고자 한다면, 다시 O Key를 누르면 된다.

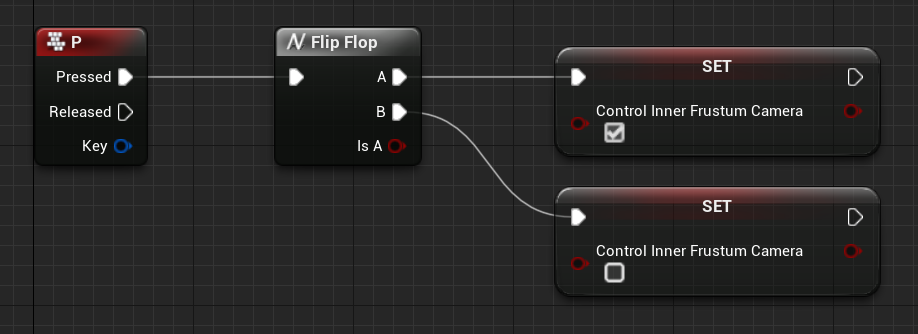


<NPlayerStagePawn 이동과 회전 입력 가능여부 설정 Blueprint Code>

**◦ NPlayerStagePawn과 InnterFrustum 입력 전환**

: 기본 정의된 Key Bind로 Pawn 뿐만 아니라 Inner Frustum 카메라 이동과 회전도 제어 가능하다.

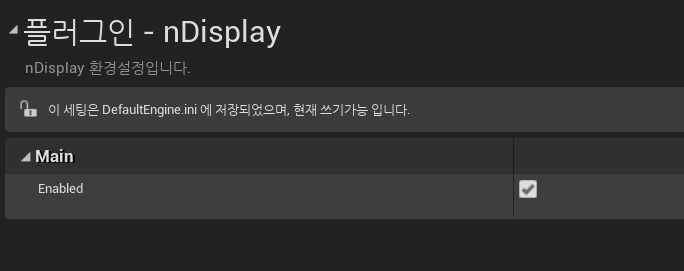
Inner Frustum 카메라 이동과 회전 입력을 사용 할 수 있도록 하기 위해서는 P Key를 누르면 된다. 다시 Pawn의 이동과 회전 입력을 되고자 한다면, 다시 P Key를 누르면 된다.



<Inner Frustum Camera 이동과 회전 입력 가능여부 설정 Blueprint Code>

## **NPlayer Template 실행**

: 우선 nDisplay 기능이 정상 동작 하기 위해서는 nDisplay 기능을 활성화 해야 된다. nDisplay 기능 활성화는 Project Settings의 nDisplay에서 Enable 항목을 Check 해 주면 된다.



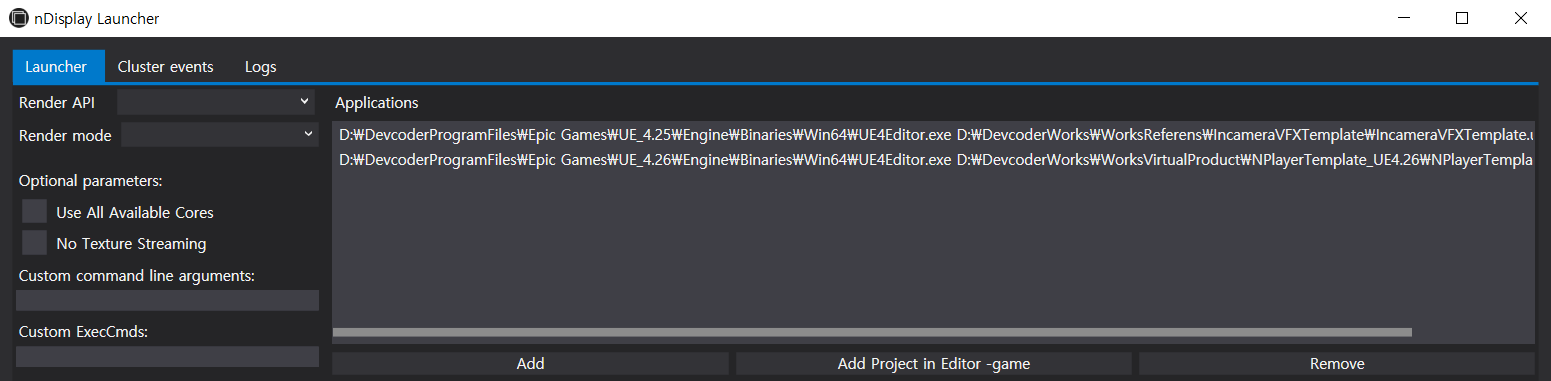
<nDisplay 기능 활성화>

1. 각 PC(nDisplay Cluster PC)에 동일한 어셋과 Project DLL 파일, Unreal Engine 4 버전을 설치 한다. **설치시 유의 할 사항은 Unreal Engine 4 설치 경로와 Project 설치 경로가 모든 PC에서 동일 해야 된다.**

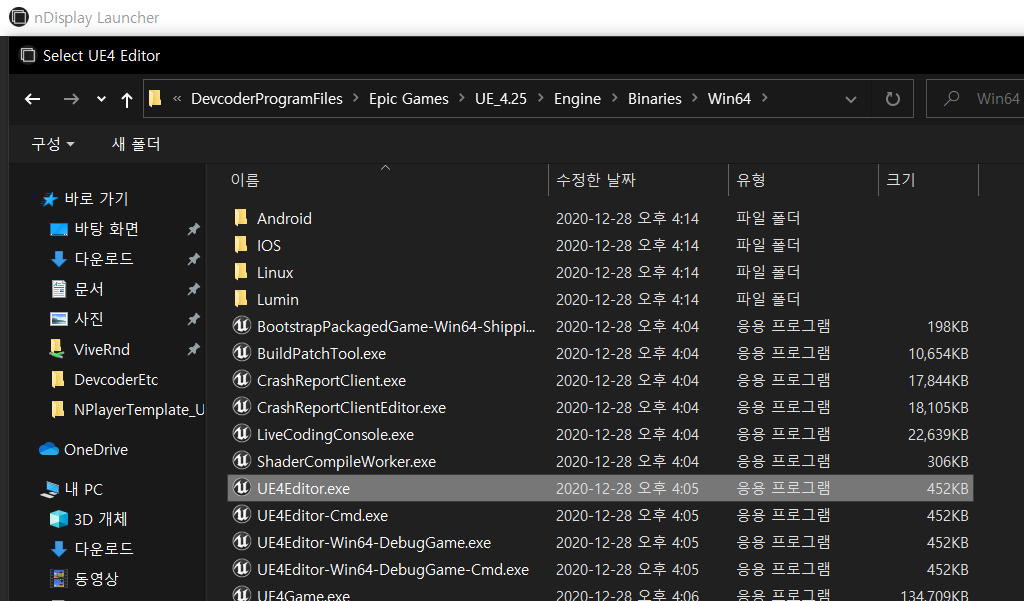
2. Unreal Engine 4의 Binaries 폴더 아래 있는 DotNET 폴더에서 nDisplayListener.exe를 각 PC에서 실행 해 둔다.

3. Main Cluster 역할을 하는 PC에서는 Unreal Engine 4의 Binaries 폴더 아래 있는 DotNET 폴더에서 nDisplayLauncher.exe를 실행 한다.

4. nDisplayLauncher에서 실행 할 Project를 등록 해 줘야 한다. Applications 항목에 보면 Add Project In Editor –game 버튼을 눌려 UE4Editor.exe 파일을 선택한다. UE4Editor.exe 파일은 Project에서 사용하는 엔진 버전의 것을 선택 해야 한다.

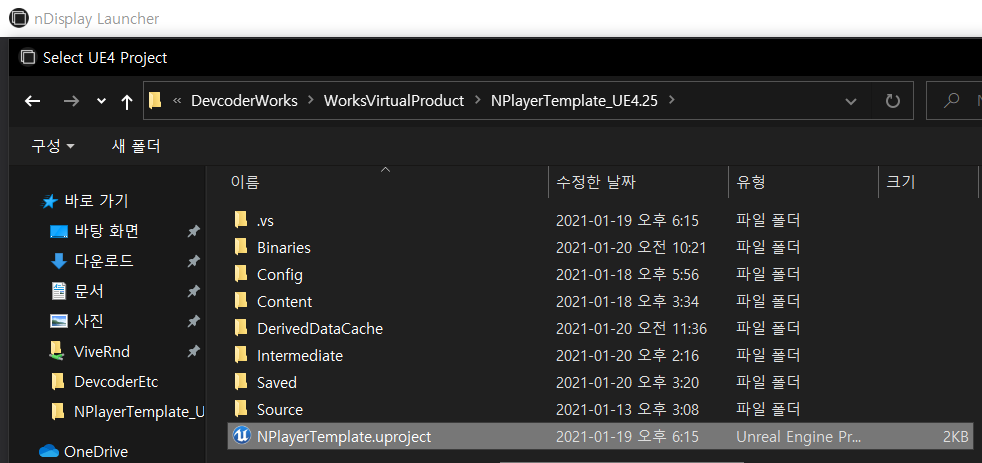


<Add Project in Editor -game>



<UE4Editor.exe 선택>

5. NPlayerTemplate.uproject 파일을 선택 한다.



<NPlayerTemplate.uproject 선택>

6. nDisplay Config File을 선택하기 위해 NPlayerTemplate의 Content폴더 내 nDisplayConfigs 폴더에서 NPlayerTemplate\_Multiviewports.cfg을 선택 한다.

7. nDisplay Launcher에서 추가한 project를 선택하고, 추가한 nDisplay Config File을 선택한 후 Run 버튼을 누르면 Main PC와 각 Cluster PC에서 nDisplay Config File에 설정한 정보에 따라 NPlayerTemplate Project가 실행 된다.

## **UE4 In Camera 최적화 옵션**

**NVIDIA NVLink를 사용하여 Multi GPU 지원을 통한 렌더링 성능 개선 방법**

배경용 LED에 렌더링 할 것과 내부 프러스텀에 렌더링 할 것을 분리하여, GPU 하나는 배경용 LED로 출력될 것을 렌더링하고, 또 다른 GPU 하는 내부 프러스텀으로 출력될 것을 렌더링하는 방식으로 성능을 개선 할 수 있다.

이 기능을 사용하기 위해서는 NVIDIA NVLink 카드와 최소 GPU 2개가 필요하다.

하드웨어 사양이 갖춰졌다면, nDisplay Config 파일의 Viewport 항목에서 다음 2가지 옵션을 설정 하면 된다.

**◦ gpu\_node=”번호”**

**◦ allow\_gpu\_transfer=”true/false”**

예:

[viewport] id=”screen\_left” … projection=”proj\_wrap” gpu\_node=”0” allow\_gpu\_transfer=”false”

[viewport] id=”rtt\_inner” … projection=”camera” gpu\_node=”1” allow\_gpu\_transfer=”true”

예시와 같이 설정 하면 된다.

gpu\_node 옵션 값에 2개 GPU를 사용하므로 0/1값을 설정 할 수 있다. 보통 배경 렌더링을 0으로 내부 프러스텀 렌더링은 1로 설정 한다.

allow\_gpu\_transfer 옵션은 이렇게 2개로 분리 하여 렌더링된 결과를 최종적으로 외부 LED로 출력을 해야 하는데, 어느쪽 GPU를 통해 보낼지를 선택 하는 옵션이다.

보통 0번 GPU가 출력을 담당하는 GPU이므로, 출력을 담당하는 GPU가 설정된 viewport에서는 allow\_gpu\_transfer 값을 false로 설정하고, 그 외 GPU가 설정된 viewport에서는 값을 true로 설정 한다.

**추가적으로 Multi GPU를 활용하기 위해서는 nDisplay Launcer에서 시작시 custom command line에 인수로 MaxGPUCount=2를 추가해줘야 한다.**

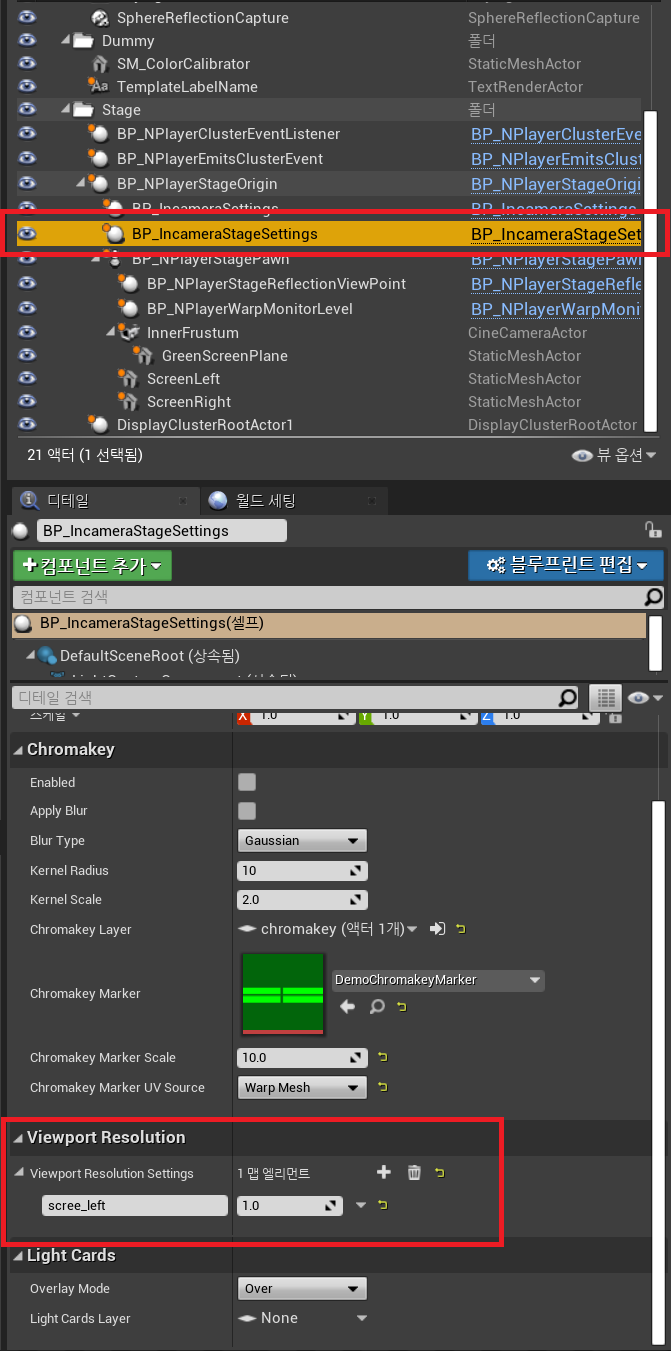
**단일 GPU에서 렌더링 성능 개선 방법**

하나의 GPU에서 배경용 LED와 내부 프러스텀으로 출력될 내용 전부 렌더링 하면 많은 부하가 발생 된다. 그래서 하나의 GPU를 사용하더라도 렌더링 성능을 개선 하기 위해 특정 Viewport의 해상도를 낮출 수 있는 기능을 사용한다.(렌더링 할 대상 크기(화면 사이즈)는 그대로 두고, 렌더링 되는 이미지 사이즈만 변경함).

방법은 BP\_IncameraStageSettings Actor에서 Viewport Resolution 항목에 Viewport Resolution Settings에 해상도를 낮출 Viewport 항목을 추가하고, nDisplay Config 파일에 설정한 Viewport ID 명을 입력하고, 변경할 해상도 비율값을 설정 한다.

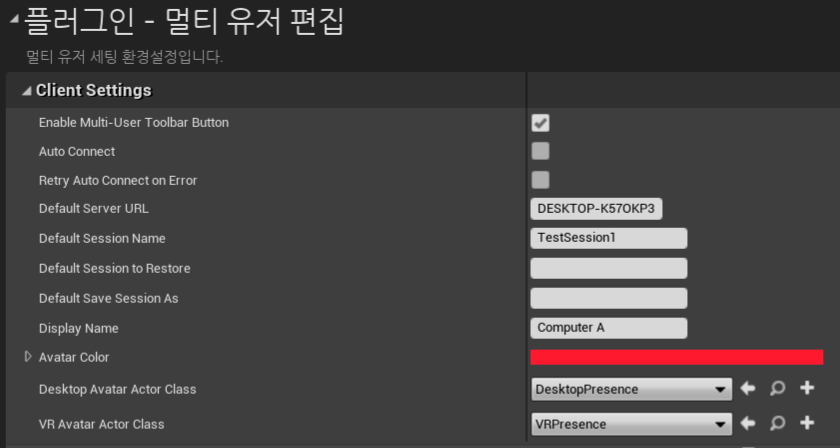
해상도 비율값은 1.0 ~ 0.0으로 1.0이면 원래 해상도이고, 1.0보다 작은 값이면 렌더링 되는 해상도는 작아지게 된다.

예시로 다음 그림을 참고.



## **UE4 In Camera VFX에서 Multi-User Editing 설정**

**◦ Multi-User Editing 설정**



Default Server URL: Multi-User Server가 실행 중인 PC의 이름을 설정.

Default Session Name: 하나의 Multi-User Server에 여러 개의 Session을 만들 수 있는데, 그 중 하나의 Session을 설정하여 자동으로 접속 가능한 Session 이름을 설정.

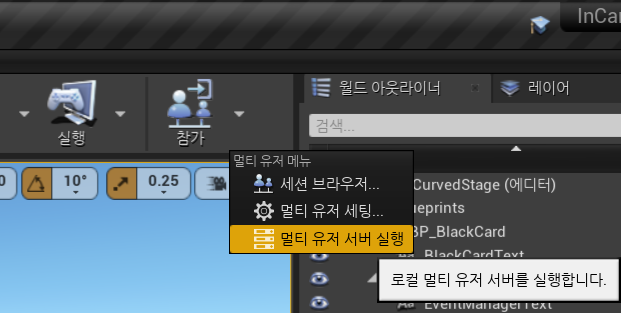
Display Name: Multi-User Editing시 화면에 보여지는 사용자 이름 설정.

Avatar Color: Multi-User Editing시 화면에 표시 되는 유저 아바타 모델의 색상을 지정.

Auto Connect: UE4가 실행되면 Default Server URL과 Default Session Name을 참고하여 바로 Multi-User Server에 접속 되도록 하는 기능을 활성화 하는 설정.

**◦ Main Cluster PC에서 Multi-User Editing Server 실행**

Toolbar의 참가 항목에서 Multi-User Server 실행 항목을 눌러 서버를 띄운다.

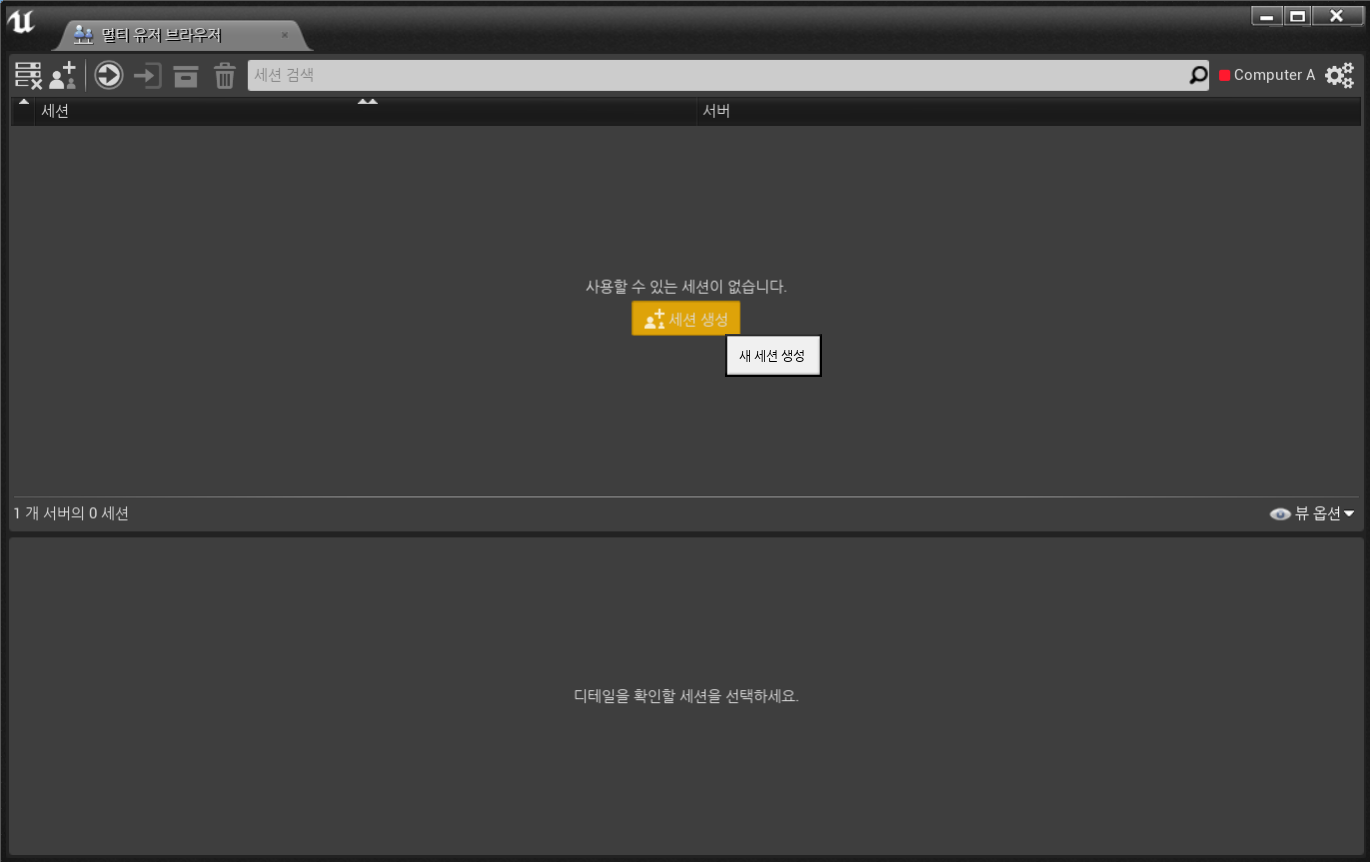


**◦ Main Cluster PC에서 Session Brower를 통해 Session 생성**

Toolbar의 참가 항목에서 Session Browser 항목을 눌러 Session Brower 창을 띄우며, 자동으로 실행된 서버에 접속 된다.



<Multi-User Session Brower 실행>

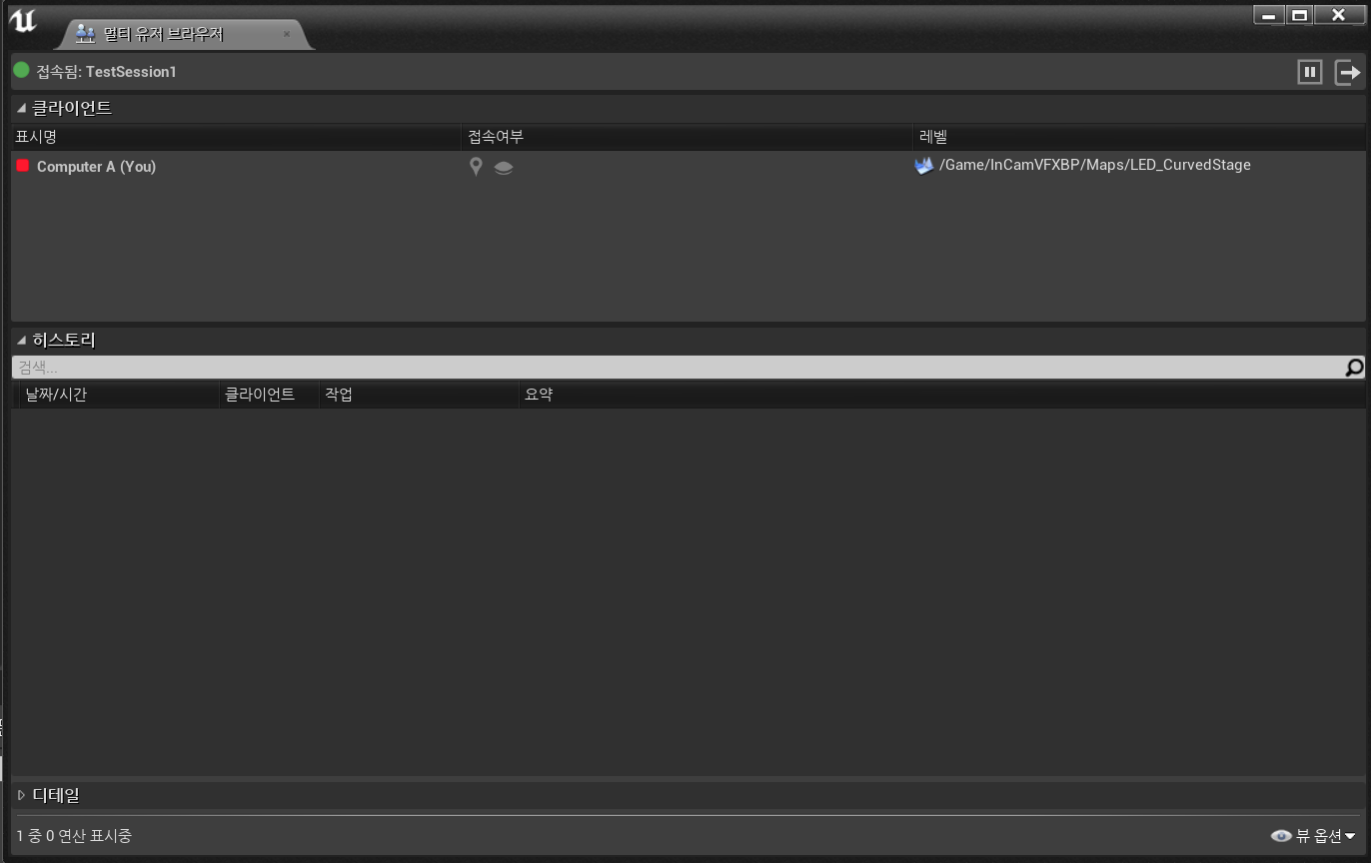


<Multi-User Session Browser>

Session 생성을 통해 새로운 Session을 생성 한다. 참고로 Cluster PC의 Multi-User Editing 설정에서 Default Server URL과 Default Session Name이 설정되어져 있다면, Default Session Name에 설정된 Session 이름과 같은 이름으로 Session을 생성해야 함. 그렇게 Session을 생성하면 Cluster PC에서 NPlayerTemplate Project 실행시 Multi-User Editing의 서버와 세션에 자동으로 접속 하게 된다.

**◦ Main Cluster PC의 UE4 Editor는 Session에 참가**

Session을 생성하면 자동으로 생성된 Session에 참가 하게 된다.

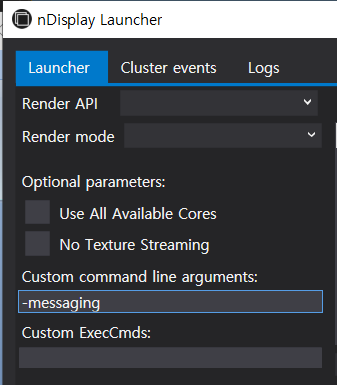


<Session에 참가된 모습>

**◦ Cluster PC들이 Multi-User Editing에 참가**

Cluster PC들에서 실행된 NPlayerTemplate Project들이 Multi-User Editing에 참가 할려면, Multi-User Editing 설정 항목중에서 Default Server URL과 Default Serssion Name이 설정 되어져 있어야 하며, Auto Connect 항목이 Check 되어져 있어야 한다.

그런 후 nDisplayLauncher에서 **–messaging** 명령어를 추가하여 실행 시키면 자동으로 Multi-User Editing에 참여 하게 된다.



## **Multi GPU 활용 하기**

## **참고**

1. [In Camera VFX Overview](https://docs.unrealengine.com/en-US/WorkingWithMedia/InCameraVFX/InCameraVFXOverview/index.html)

2. [In Camera VFX Quick Start](https://docs.unrealengine.com/en-US/WorkingWithMedia/InCameraVFX/InCameraVFXQuickStart/index.html)

3. [nDisplay Configuration File Reference](https://docs.unrealengine.com/en-US/WorkingWithMedia/nDisplay/Configuration/index.html)

4. [Multi-User Editing Overview](https://docs.unrealengine.com/en-US/ProductionPipelines/MultiUserEditing/Overview/index.html)

5. [Multi-User Editing Quick Start](https://docs.unrealengine.com/en-US/ProductionPipelines/MultiUserEditing/QuickStart/index.html)

6. [Advanced Multi-User Networking](https://docs.unrealengine.com/en-US/ProductionPipelines/MultiUserEditing/Networking/index.html)

7. [Multi-User Editing Reference](https://docs.unrealengine.com/en-US/ProductionPipelines/MultiUserEditing/Reference/index.html)