

Universal cross-platform Pro Voice Chat

With this plugin, you can integrate easily in Blueprint or C++ :

- Team voice chat like in Counter Strike
- Proximity voice chat like in PUBG
- Global voice chat in a lobby for example
- It is cross platform Windows / Android / iOS

There is no need to add any component to your character, there is no need to heavily modify your existing project.

Patch Updates:

1.4 : Fix attenuation 3D on player host on listen server

1.3 : Fix bug PC if no microphone connected

1.2 : Fix windows microphone first speak random sound, adjust opus codec settings

1.1 : Fix listen server

1.0 : Release

Content

I.	Description	2
1.	Features.....	2
2.	Contact	2
II.	Setting up your project.....	3
1.	Project Config file.....	3
2.	Install plugin.....	3
III.	Using voice chat in Blueprint.....	4
1.	How the plugin works.....	4
2.	Creating voice chat actor "PlayerVoiceChatActor" on server	4
3.	Using voice chat actor "PlayerVoiceChatActor" on client	7
4.	Complete API PlayerVoiceActor	11
IV.	Example Project	13
V.	UE4 Project Network Settings	13
VI.	FAQ, Troubleshoots	14

I. Description

This plugin is made for **cross-platform multiplayer voice chat**. It supports **Windows, Android, iOS, Linux and Windows dedicated server**. There is several **voice chat mode** : **Global, team/radio, proximity**.

It is heavily **network optimized** with usage of **Opus encoding**, and server **send network data only to relevant clients** (for example, only clients in the same team for team voice chat).

It is intended to be used with **Blueprint but is also 100% compatible with C++**.

It can be used on Internet or LAN on dedicated or listen server.

1.Features

- **Cross-platform** Windows, Android, iOS voice chat
- **Different types of voice chat** : Global, team/radio, proximity
- **Network optimized** : server use compressed data with opus codec, server send data to only relevant clients given settings of voice chat (Global, team/radio, proximity)
- **Mute a player**
- **3D Sound or 2D Sound**, you can use your own attenuation settings
- **Supports choose of microphone sample rate** and channels number to let you choose quality and network usage.
- **Set your microphone volume**
- **Get raw PCM data of microphone input** on Windows, Android, iOS (useful if you want to make your own voice chat)
- **Compatible with any UE4 Online Subsystem** as long as they uses UE4 standard network API (RPC, replicated variables).
- **Tested on hardware** : Dedicated server Linux, Dedicated server Windows, Client Windows, Client Android 8 / 9 / 10, Client iPhone iOS 12.0 / 13.0 / 14.0

2.Contact

For any help related to this plugin, please send an email at contact@meoplay.com

Or join discord server : [DISCORD](#)

II. Setting up your project

1. Project Config file

If you use this plugin for a Windows or both Windows/Mobile game then you need to activate microphone input. If you use this plugin for a mobile game only, you can skip this part.

- 1) Go to your UE4 project folder, open Config/DefaultEngine.ini
- 2) Add the following lines :
[Voice]
bEnabled=true
- 3) Save

2. Install plugin

- 1) Open your UE4 project
- 2) Enable plugin “Universal cross-platform voice chat”
- 3) Restart

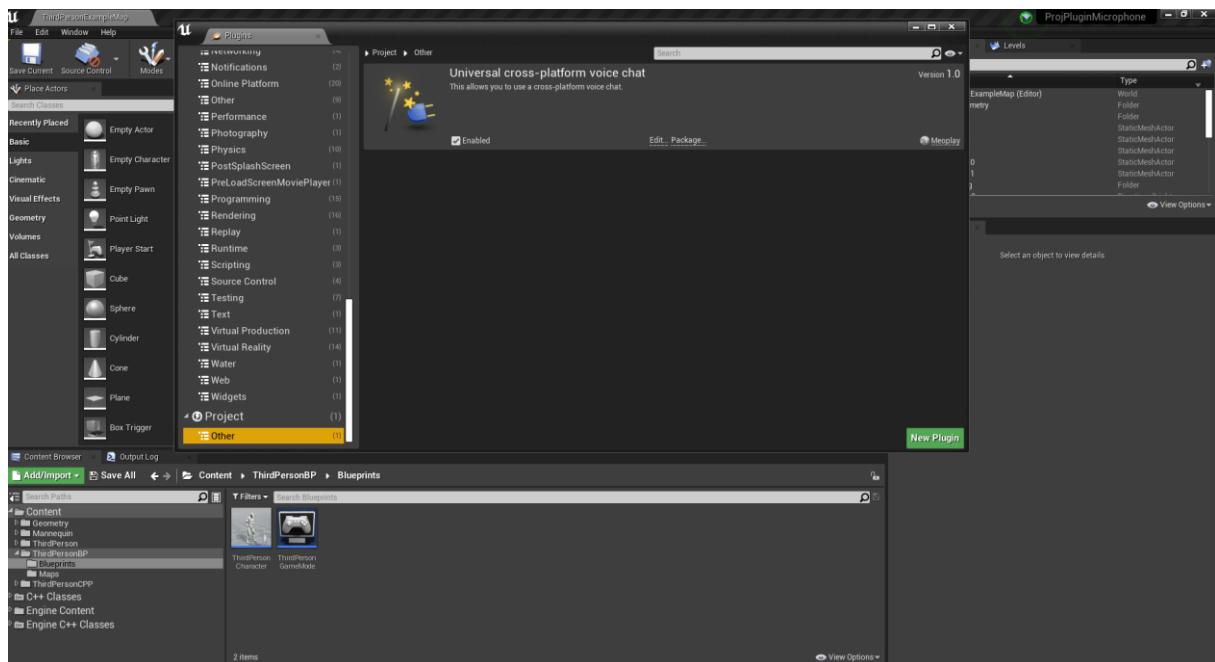


Figure 1 Enable plugin

III. Using voice chat in Blueprint

1. How the plugin works

Server spawns an actor called “PlayerVoiceChatActor”, it is replicated to the client, then the client can start using it (start speaking, stop speaking, mute, ...)

There is one class called “PlayerVoiceChatActor” that controls one player voice chat. There is one component called “MicrophoneSpeakComponent” that handles microphone input and audio output.

You are only required to use “PlayerVoiceChatActor”, you need to **spawn one “PlayerVoiceChatActor” on server for each player** that connects to the game.

“PlayerVoiceChatActor” is then **replicated from server to client**. Each client easily retrieve his own “PlayerVoiceChatActor” with a static variable comparing owner with local player controller.

“PlayerVoiceChatActor” is **automatically destroyed** by server when player leaves the game (binded event to “player controller destroyed”)

On Mobile Android and iOS, you should ask permission before starting voice chat.

2. Creating voice chat actor “PlayerVoiceChatActor” on server

“PlayerVoiceChatActor” needs to be spawn on server and needs to be owned by a player controller.

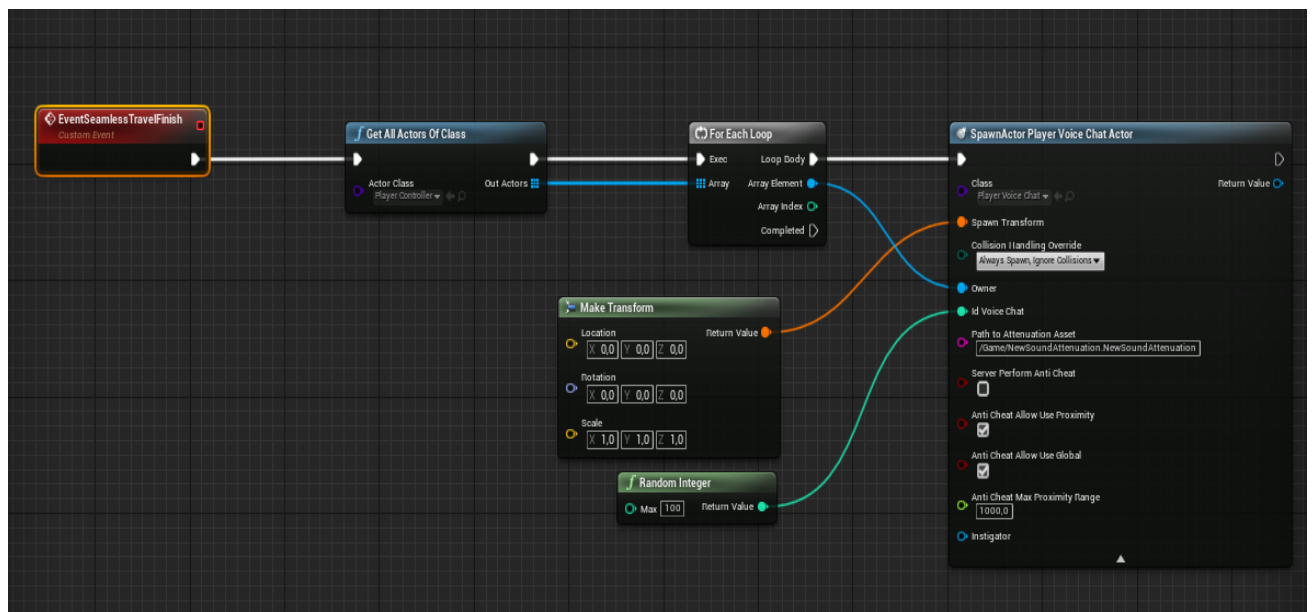
It is recommended to **spawn “PlayerVoiceChatActor” in GameMode when OnPostLogin** is called.

You can do this like this:



Note relating attenuation asset in packaged game : In cooked builds your assets won't get cooked if you load them from code. They need to be referenced through some properties or in a level.

If you use seamless travel, in game mode or another blueprint on server side only you need to spawn one Voice Chat Actor for each controller when seamless travel is done.



3.Using voice chat actor

“PlayerVoiceChatActor” on client

The functions used in next steps are global functions, you don't need to have a reference to actor or component, just search for function in category “Voice chat universal” in Blueprint.

1) Initializing Audio Settings (Optional)

You can manually set the sample rate and number of channel used by the microphone. If you don't set it, it will be automatically set when you start speaking (default : 48000Hz and 1 channel).

Supported sample rate are 8000, 16000, 24000, 48000Hz. Number of channels supported are 1, 2.

Using a sample rate of 48000 Hz consume twice as much network data to 24000Hz.

Using 2 channel instead of 1 consume twice as much network data

For a good audio quality, use 48000Hz and 1 channel. 24000Hz, 1 channel is acceptable too.

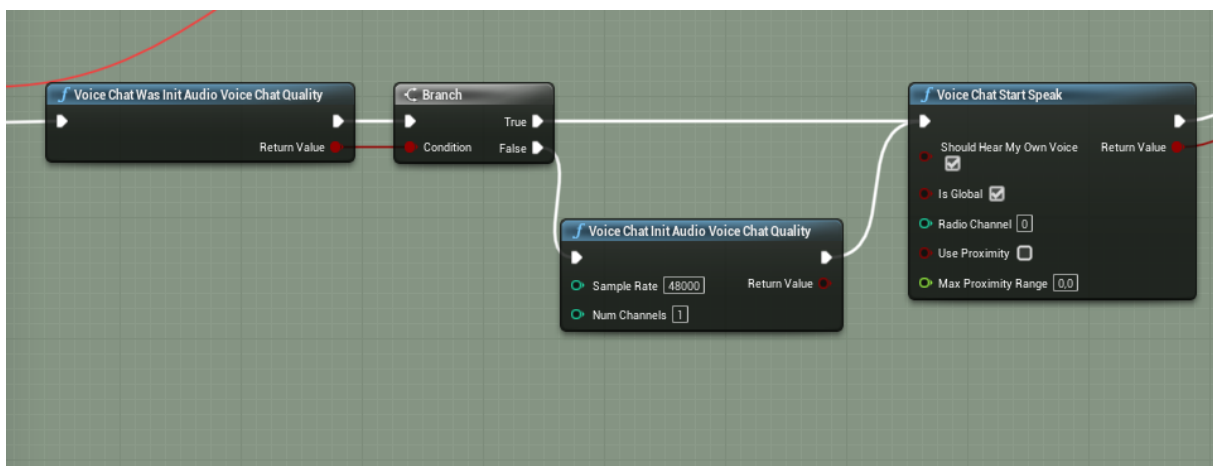


Figure 3 Manually initializing before speaking

2) Speak voice chat

To start speak on voice chat, use node “Voice Chat Start Speak”.

To stop speak on voice chat, use node “Voice Chat stop Speak”.

If you want to hear your own voice when speaking, set “ShouldHearMyOwnVoice” to true in “Voice Chat Start Speak”.

For making sure your microphone and the plugin works, you should make sure to enable voice playback with parameter “ShouldHearMyOwnVoice” when using node “Voice Chat Start Speak”

a) Speaking global voice chat, no proximity (Example : Lobby)

For example use the following in your character blueprint (or anywhere else it doesn't matter), use the blueprint node "Voice Chat Start Speak" with following settings :

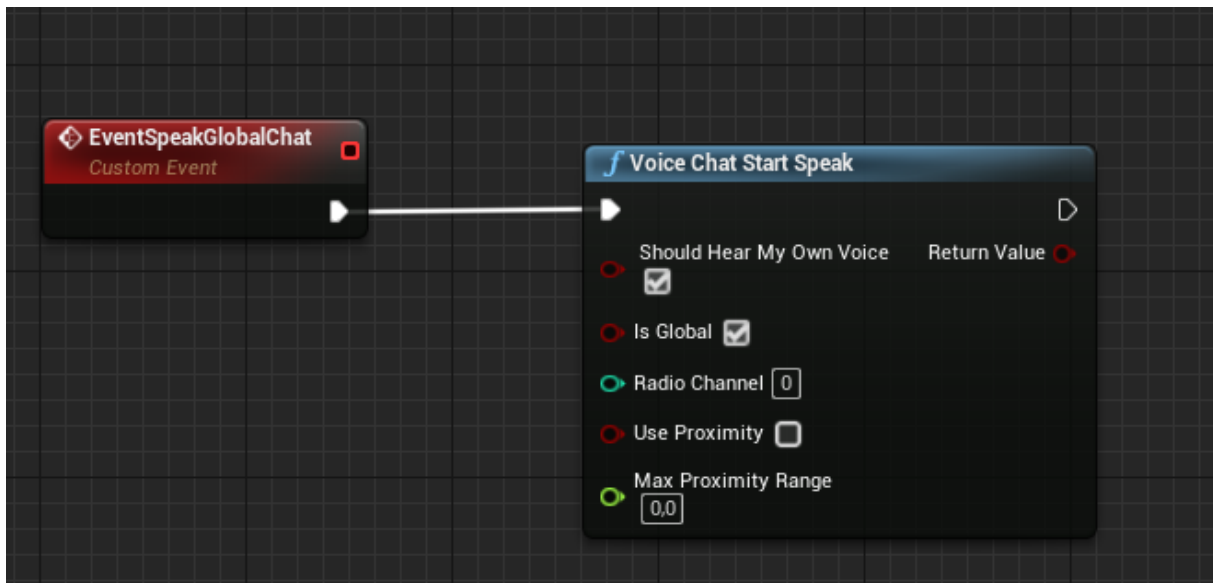


Figure 4 Speak to everyone wherever their character are in the map

b) Speaking team voice chat, no proximity (Example : Counter Strike)

Use the blueprint node "Voice Chat Start Speak" with following settings, **usually you would put one radio channel per team** :

IsGlobal : false

Radio channel : Put your team channel, here it's 1

Use proximity : false

Max Proximity Range : players within this distance will be sent voice chat, you can 1000 for example

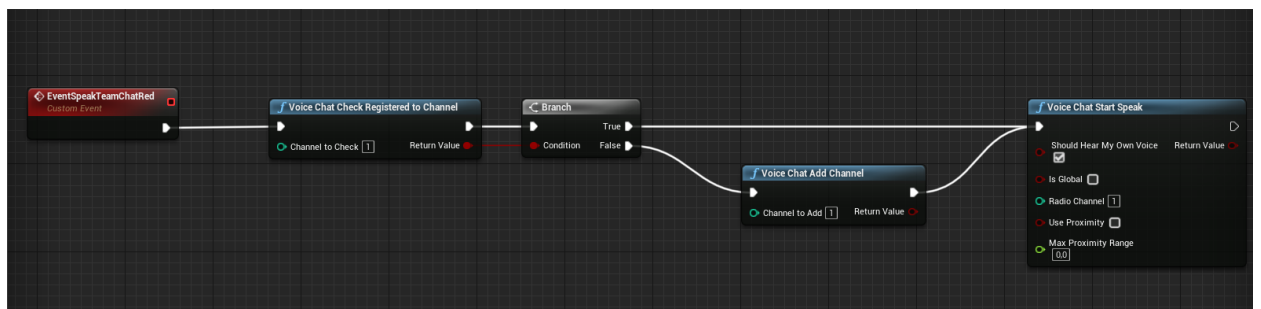


Figure 5 Speaking to players that are registered in channel 1, wherever their character are in the map

Example of Counter Strike like voice chat with red team (Channel 1) and green team (Channel 2):

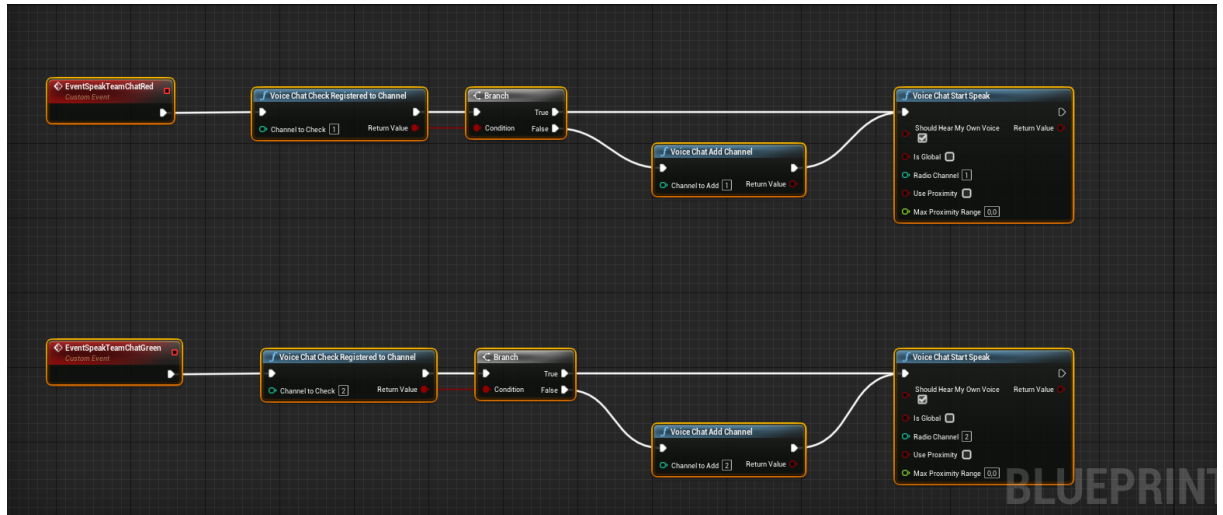


Figure 6 Example of two team voice chat

c) Speaking global voice chat within inside proximity (Example : PUBG)

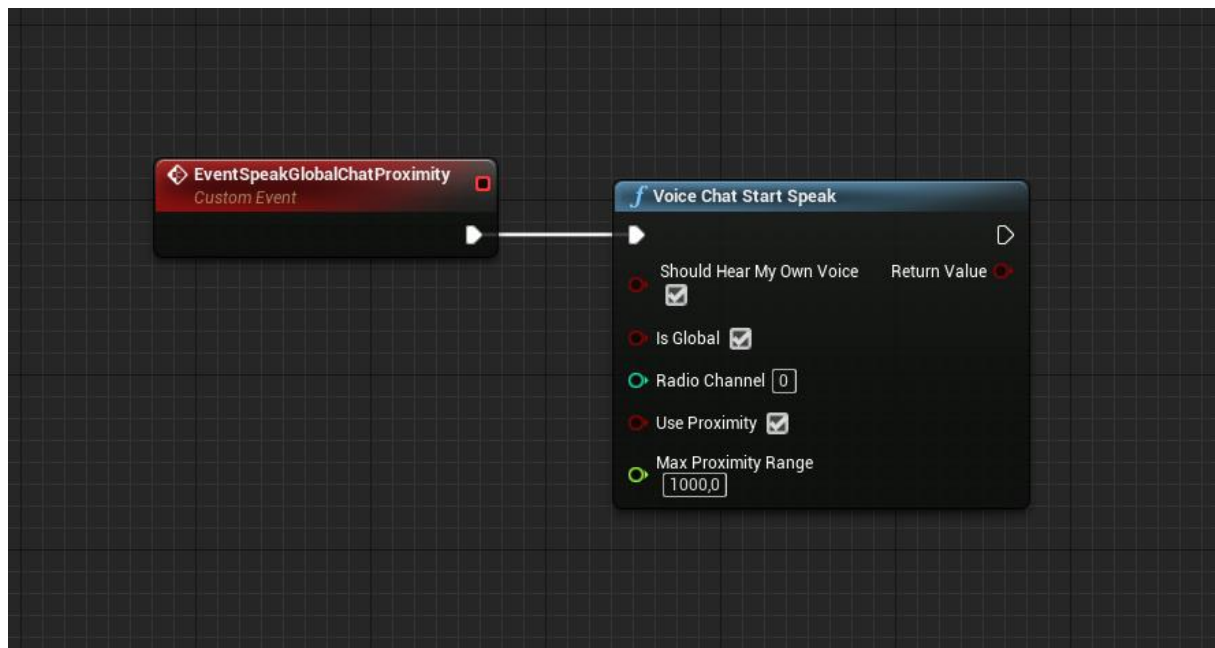
Use the blueprint node “Voice Chat Start Speak” with following settings :

IsGlobal : true

Radio channel : Not used, you can put 0

Use proximity : true

Max Proximity Range : players within this distance will be sent voice chat, you can 1000 for example



d) Speaking team voice chat within inside proximity (Example : PUBG)

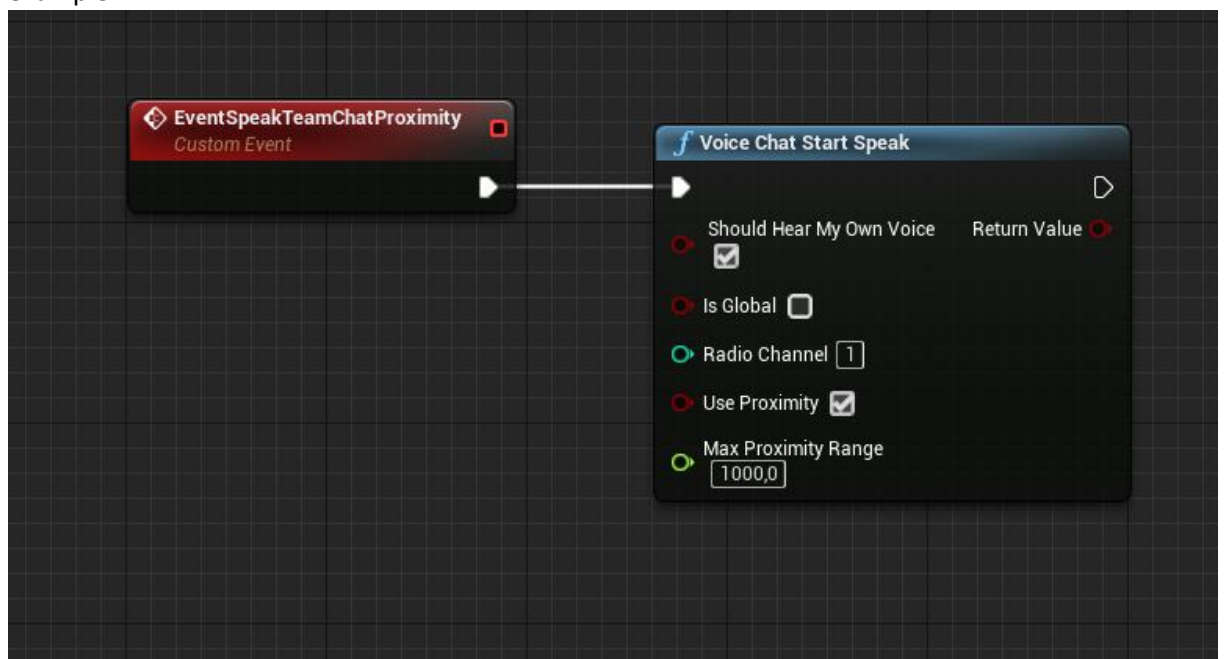
Use the blueprint node “Voice Chat Start Speak” with following settings :

IsGlobal : false

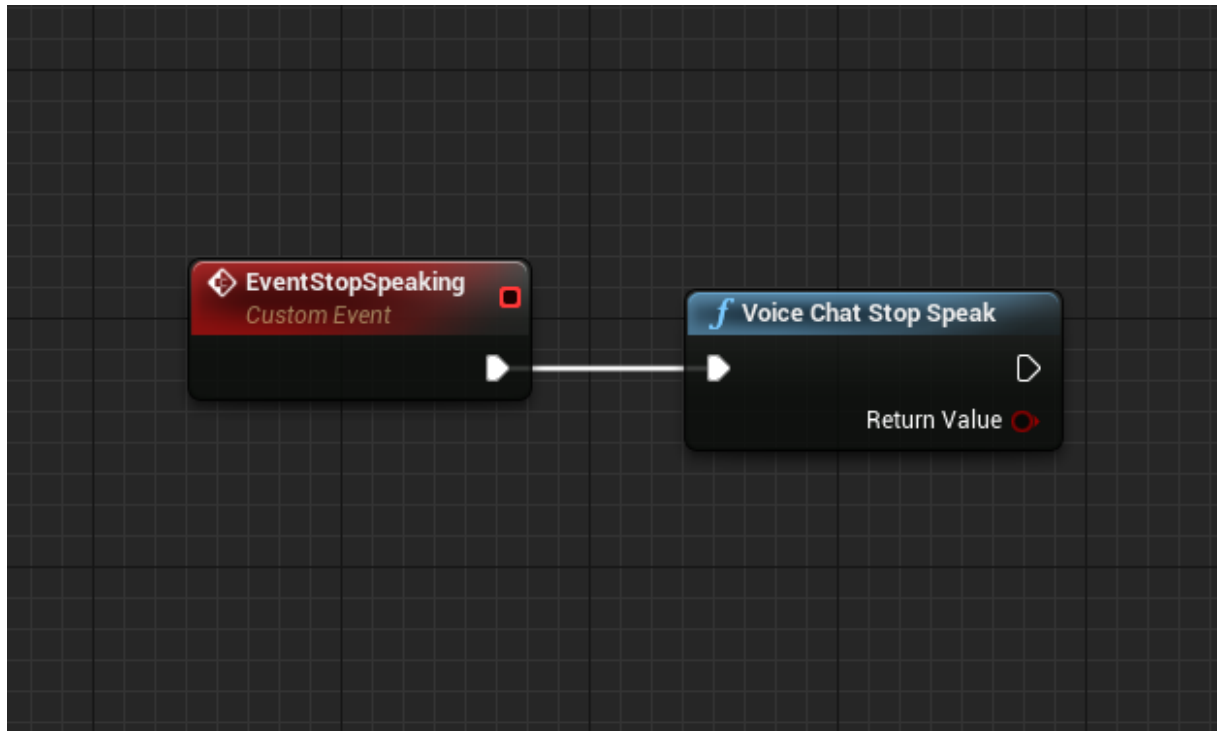
Radio channel : Put your team channel, here it is 1

Use proximity : true

Max Proximity Range : players within this distance will be sent voice chat, you can 1000 for example



e) Stop speaking



4. Complete API PlayerVoiceActor

This are the functions from “PlayerVoiceChatActor” you can use client side with this plugin with Blueprint or C++.

This is a list to manipulate *local player only*, and should be called on *client program only*.

These are *global function that you can call anywhere* which retrieve the right PlayerVoiceChatActor that *your local client own*.

1) Client

bool VoiceChatInitAudioVoiceChatQuality(int32 _sampleRate = 48000, int32 _numChannels = 1);

- Used to initialize audio encoder / decoder. If you don't initialize before starting to speak, it will automatically be initialized with default sample rate of 48000 and 1 channel.

bool VoiceChatWasInitAudioVoiceChatQuality();

- Check if the audio was already initialized

bool VoiceChatStartSpeak(bool _shouldHearMyOwnVoice = true, bool isGlobal = true, int radioChannel = 0, bool useRange = false, float maxRange = 0);

- Use this to start speaking

bool VoiceChatStopSpeak();

- Use this to stop speaking

bool UUniversalVoiceChat::VoiceChatSetMicrophoneVolume(float volume);

- Use this to set your microphone volume.

void VoiceChatLocalMuteSomeone(const UObject* WorldContextObject, APlayerState *playerToMute, bool shouldMute);

- Use this to mute someone

bool VoiceChatLocalsIsMutedSomeone(const UObject* WorldContextObject, APlayerState *playerToCheckMute);

- Use this to check if someone is currently muted

bool VoiceChatAddChannel(int32 channelToAdd);

- Use this to add a team / radio channel

bool VoiceChatRemoveChannel(int32 channelToRemove);

- Use this to remove a team / radio channel

bool VoiceChatHasMicrophonePermission();

- Use this on mobile to check if audio permission was granted

void VoiceChatAskMicrophonePermission();

- Use this on mobile to ask for audio permission

2) Server

This are the functions from “PlayerVoiceChatActor” you can use server side with this plugin with Blueprint or C++. This is meant for server validation, client and server should allow settings (global, team ...) or server discard will discard voice chat if settings are not allowed.

void ServerSetAllowUseGlobal(bool _allowUseGlobal);

- Use this to allow a client to use global chat, **Note : do nothing if anticheat is not enabled**

void ServerAddChannel(int32 channelToAdd);

- Use this to add a team channel on a client, **Note : do nothing if anticheat is not enabled**

void ServerRemoveChannel(int32 channelToRemove);

- Use this to remove a team channel on a client, **Note : do nothing if anticheat is not enabled**

void ServerSetAttenuation(bool enableAttenuation, FString _pathToAttenuationAsset);

- Use this to change 3D sound attenuation settings, replicated to every clients.

void ServerSetAllowUseProximity(bool _allowUseRange);

- Use this to enable proximity chat on a client, **Note : do nothing if anticheat is not enabled**

void ServerSetMaxProximityRange(float _maxProximityRange);

- Use this to set proximity max range on a client, **Note : do nothing if anticheat is not enabled**

IV. Example Project

The example project use 3D sound with a range of 1600. You can disable this in example game mode blueprint by setting attenuation path asset empty.

You can try a test project connecting to my master server in the link below:

<https://drive.google.com/file/d/172X-wUUT7CtysoyiHdB64TBo1w0Dkl6k/view?usp=sharing>



Figure 7 Example Project

V. UE4 Project Network Settings

It is highly recommended to setup your project to fit your network bandwidth needs, otherwise you can expect stutter, packet drops.

An example of project settings configuration I use :

File DefaultEngine.Ini :

```
[/Script/Engine.Player]
ConfiguredInternetSpeed=500000
ConfiguredLanSpeed=500000
[/Script/OnlineSubsystemUtils.IpNetDriver]
MaxClientRate=100000
MaxInternetClientRate=100000
```

File DefaultGame.Ini :

```
[/Script/Engine.GameNetworkManager]
TotalNetBandwidth=4000000
MaxDynamicBandwidth=100000
MinDynamicBandwidth=40000
```

VI. FAQ, Troubleshoots

- I can't listen others players.
 - **All platforms:** It is not recommended to use "**VoiceChatStartSpeak**" in begin play event, you need to wait for "PlayerVoiceChatActor" to be replicated. If you need to use "**VoiceChatStartSpeak**" in begin play event, then you should make a timer and poll return value of "**VoiceChatGetMyLocalPlayerVoiceChat**" until it return a non null value.
 - **Android, iOS, Oculus Quest:** make sure to ask permission with "**VoiceChatAskMicrophonePermission**" before using "**VoiceChatStartSpeak**"
- It is lagging.
 - Try to adjust your network settings, refer to chapter V of this documentation.
 - You can also use a sample rate of 24000 Hz 1 channel instead of standard 48000Hz 1 channel in function **VoiceChatInitAudioVoiceChatQuality(int32 _sampleRate = 48000, int32 _numChannels = 1);**