RemoteAssistant SDK

1. Description

The SDK supports UnityEditor,Android,iOS,Windows,UWP(Hololens).

Support multiple video types:device camera/unity camera /custom Texture/PC screen, support VR,AR, MR mixed video.

Support group audio and video calls,server chat video record.

The interface is simple, the video/audio capture and codec are separated, you can insert the sdk to your own network easily

https://github.com/stonerey/RemoteAssitantSDK

Support Email:1053050442@qq.com

2.Project Setting

2.1 Player Settings -> Other Settings -> Configuration

->Scripting Runtime Version ->. Net 4.x . Mono

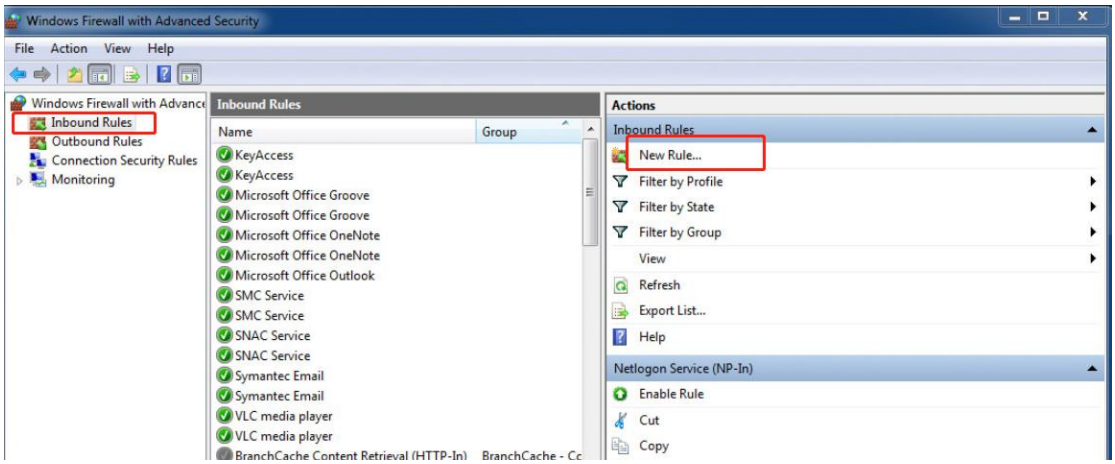
2.2Create a project with Unity 2019.4 and above.

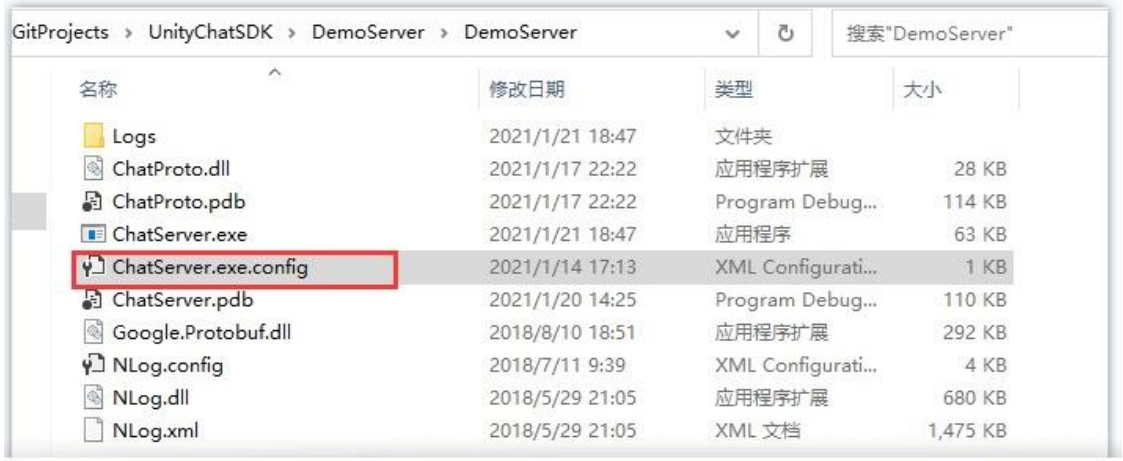
2.3Edit-> ProjectSettings -> Audio ->DSP -> **Good Latency**

3.Server Config

In the Project contains a test server DemoServer.zip, please unzip the zip and run the server, you can set the port， the number of connections and whether to record in the config file, in addition,

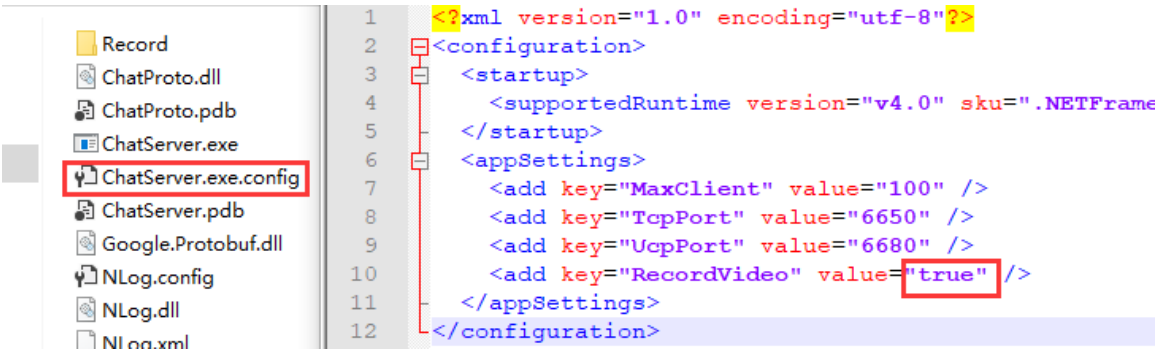
you need to set the firewall advanced settings , add those port to inbound rule.

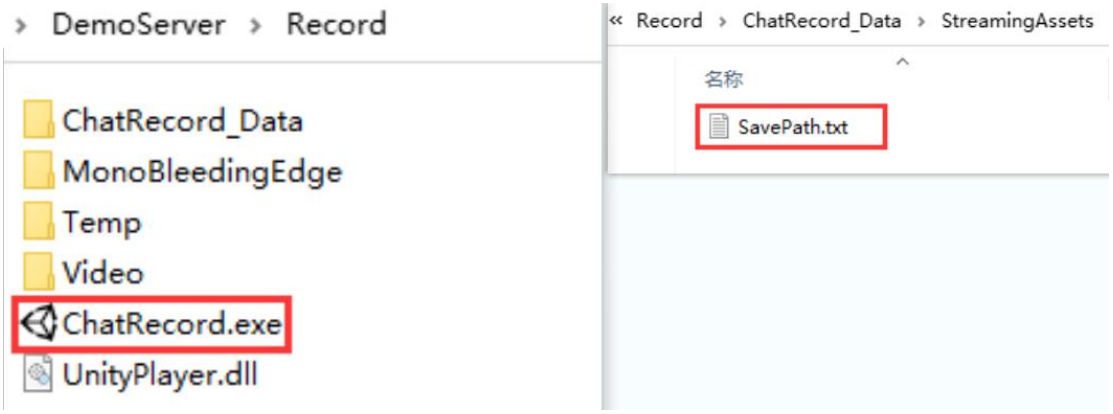





```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <appSettings>
    <add key="MaxClient" value="100" />
    <add key="TcpPort" value="6650" />
    <add key="UcpPort" value="6680" />
    <add key="RecordVideo" value="false" />
  </appSettings>
</configuration>
```

If you need to record videos please to set config “recordVideo” key to “true” and run ChatRecord.exe. If you need to set a specific video directory, please set SavePath.txt.





4.Integrate the SDK into your own project

You can intergrate the sdk in your own network. The SDK provides video capture, codec

5. Main Interface

```
/// <summary>
/// Initialize audio
/// </summary>
void InitMic(int index = 0);
/// <summary>
/// Initialize video
/// </summary>
void InitVideo(int index = 0);
/// <summary>
/// Set the type of video capture
/// </summary>
/// <param name="type">type of video capture</param>
/// <param name="captureCamera">If the capture type is unitycamera,
you need to identify a unitycamera.</param>
```

```
bool SetVideoCaptureType(VideoType type, Camera captureCamera =
null);
/// <summary>
/// Set the resolution of the video
/// </summary>
/// <param name="resolution">resolution of the video</param>
void SetResolution(VideoResolution resolution);
/// <summary>
/// Get the size of the current video resolution
/// </summary>
/// <returns></returns>
Vector2 GetResolutionSize();
/// <summary>
/// Set the compression quality of the video
/// </summary>
/// <param name="quality">quality of the video</param>
void SetVideoQuality(VideoQuality quality);
/// <summary>
/// Start capturing audio and video
/// </summary>
/// <returns>Capture results</returns>
CaptureResult StartCapture();
/// <summary>
/// Stop capturing audio and video
/// </summary>
void StopCpture();
/// <summary>
/// Get the current audio package
/// </summary>
/// <returns></returns>
AudioPacket GetAudio();
/// <summary>
/// Get the current video package
/// </summary>
/// <returns></returns>
VideoPacket GetVideo();
/// <summary>
/// Decode the audio of the peer
/// </summary>
/// <param name="id">peer id</param>
/// <param name="packet">peer audioPacket</param>
void DecodeAudioData(AudioPacket packet);
float[] DecodeAudioFloatData(AudioPacket packet);
/// <summary>
```

```
/// Decode the video of the peer
/// </summary>
/// <param name="packet">peer videoPacket</param>
Texture2D DecodeVideoData(VideoPacket packet);
/// <summary>
/// Send your customTexture
/// </summary>
/// <param name="tex">the Texture2D to be sent</param>
void UpdateCustomTexture(Texture2D tex);
/// <summary>
/// Add extra float data to the current video frame, optional
/// </summary>
/// <param name="data"></param>
void AddVideoFloatData(List<float> data = null);
/// <summary>
/// Set up audio capture available
/// </summary>
/// <param name="enable"></param>
void SetAudioEnable(bool enable);
/// <summary>
/// Set up video capture available
/// </summary>
/// <param name="enable"></param>
void SetVideoEnable(bool enable);
/// <summary>
/// Switch device camera
/// </summary>
void SwitchCam();
/// <summary>
/// Set device front camera
/// </summary>
/// <returns></returns>
bool SetCamFrontFacing();
/// <summary>
/// Get the current volume of the peer
/// </summary>
/// <param name="id">peer id</param>
/// <returns></returns>
float GetPeerAudioVolume(int id);
/// <summary>
/// Get peer videoInfo
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
```

```
VideoInfo GetPeerTexture(int id);
/// <summary>
/// Get the current volume of yourself
/// </summary>
/// <returns></returns>
float GetSelfAudioVolume();
/// <summary>
/// Get self videoInfo
/// </summary>
/// <returns></returns>
VideoInfo GetSelfTexture();
/// <summary>
/// Start record audio
/// </summary>
/// <param name="limit">Maximum recording time, timeout
automatically stop</param>
/// <returns></returns>
bool StartRecordAudio(int limit, OnRecordFinished onFinished);
/// <summary>
/// Stop record audio
/// </summary>
/// <returns></returns>
void StopRecordAudio();
/// <summary>
/// Play record audio
/// </summary>
/// <param name="recordData"></param>
void PlayRecordAudio(byte[] recordData);
/// <summary>
/// Convert texture to Texture2D
/// </summary>
/// <param name="texture"></param>
/// <param name="format"></param>
/// <param name="type"></param>
/// <returns></returns>
Texture2D ConvertTexture(Texture texture, TextureFormat format =
TextureFormat.BGRA32, ConvertType type = ConvertType.None);
/// Capture desk screen video
void StartDeskCapture();
void StoptDeskCapture();
Texture2D GetDeskTexture(int id = 0);
```

6. Registered

In order to prevent piracy, developers need to send an email to obtain the activation code after purchase. Please provide your request key and the purchase order bill in the email . You can get two activation keys to bind two computers.

Email:1053050442@qq.com