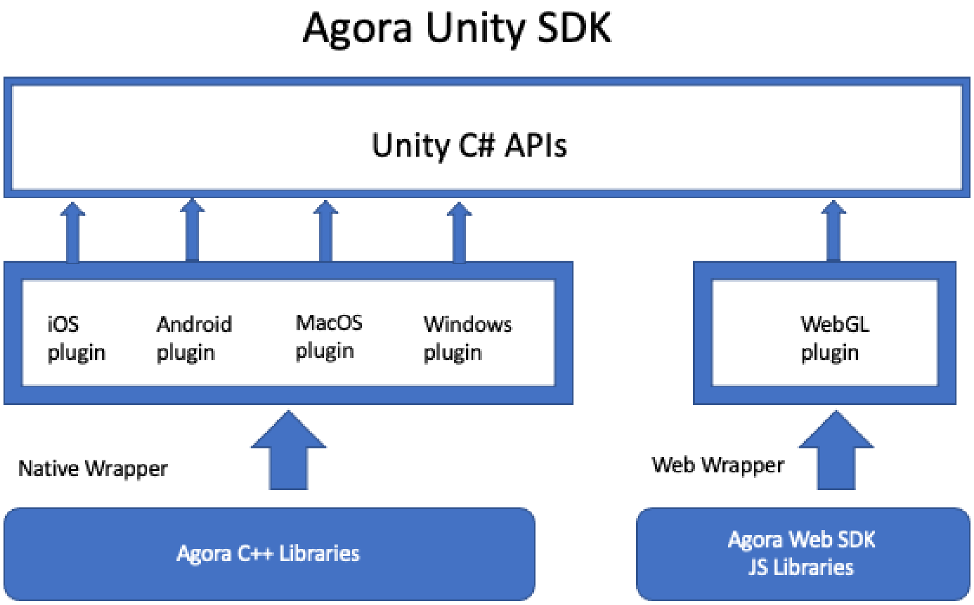
Agora WebGL Plugin README

The Unity WebGL Plugin is a wrapper library that uses Agora Web SDK 4.x as the core RTC engine. The plugin adds the latest Web RTC features to the core functionality provided by the original Agora Unity SDK for native platforms. The following illustrate the structure:



Where can I get the latest version?

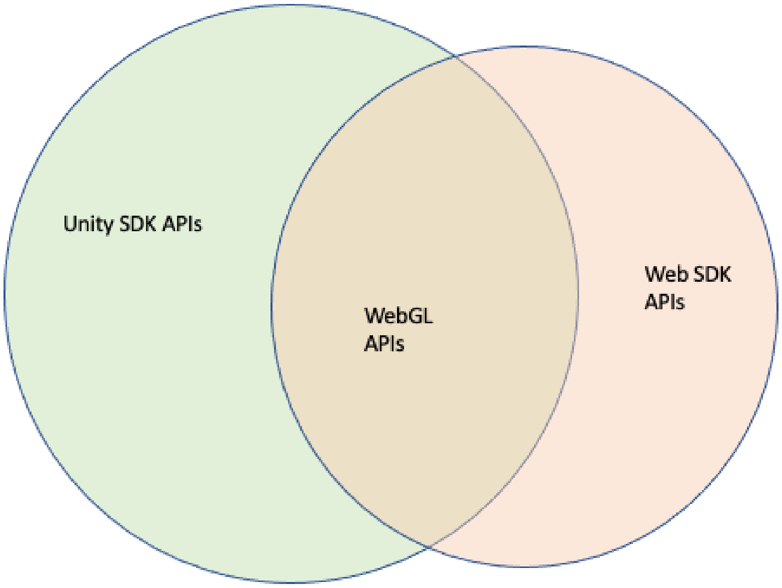
The WebGL plugin is a separate unity package from the main SDK. Download the latest package from the release section of the GitHub repo. Follow the instructions on the Slack announcement.

What is in this plugin?

This plugin is a community beta or preview of the fully featured SDK. It is based on the original Unity SDK version 3.7.0. This beta SDK is open source, and we highly encourage any developer to contribute changes or push new features to the repository in a pull request.

What are the limitations?

Because of the different technologies used on the native SDKs and Web SDK, the WebGL plugin only supports the features available in Agora Web SDK. However, because the WebGL plugin only implements interfaces defined by Unity SDK, it is not possible to map all features in the Web SDK to this plugin. For example, Raw Data Manager and PushAudioFrame API are not included in the WebGL Plugin APIs. The follow diagram roughly illustrate the relationship:



Appendix A lists supported and unsupported features.

Unity does not officially support WebGL on mobile browsers. WebGL is designed for desktop browsers and works best on Google Chrome. Agora APIs that specifically call out for mobile support are not implemented in this release.

Where can I see a demo?

The SDK includes a demo that replaces the original Unity Video SDK demo. You may find a prebuilt version online at https://webgl.agoraguru.net. Since this is a public demo with fixed AppID, you should choose a different channel name for your test, if you don't want to run into other unknown developers who are testing the demo at the same time.

How do I get support?

We collect feedback and bug reports. Please post issues on the GitHub location. However, the best resolution comes from the community. Although Agora staff actively responds to questions, we don't guarantee the same SLA as the official launched products.

Appendix A - Supported APIs

If an API is not listed here, then it is probably not supported.

Class	C# Method
AgoraChannel	bool CreateEngine(string appId);
AgoraChannel	CONNECTION_STATE_TYPE GetConnectionState()
AgoraChannel	int AddPublishStreamUrl(string url, bool transcodingEnabled)
AgoraChannel	int AddPublishStreamUrl(string url, bool transcodingEnabled);
AgoraChannel	int AdjustUserPlaybackSignalVolume(uint userId, int volume)
AgoraChannel	int CreateDataStream(bool reliable, bool ordered)
AgoraChannel	int DisableAudio();
AgoraChannel	int DisableLastmileTest();
AgoraChannel	int DisableVideo();
AgoraChannel	int EnableAudio();
AgoraChannel	int EnableAudioVolumeIndication(int interval, int smooth, bool report_vad);
AgoraChannel	int EnableEncryption(bool enabled, EncryptionConfig encryptionConfig)
AgoraChannel	int EnableLastmileTest();
AgoraChannel	int EnableLocalAudio(bool enabled);
AgoraChannel	int EnableLocalVideo(bool enabled);
AgoraChannel	int EnableVideo();
AgoraChannel	int EnableVideoObserver();
AgoraChannel	int GetConnectionState();
AgoraChannel	int JoinChannel(string channelName, string info = "", uint uid = 0)
AgoraChannel	int JoinChannel(string token, string info, uint uid, ChannelMediaOptions channelMediaOptions)

AgoraChannel	int LeaveChannel();
AgoraChannel	int MuteAllRemoteAudioStreams(bool mute);
AgoraChannel	int MuteAllRemoteVideoStreams(bool mute);
AgoraChannel	int MuteLocalAudioStream(bool mute);
AgoraChannel	int MuteLocalVideoStream(bool mute);
AgoraChannel	int MuteRemoteAudioStream(uint userId, bool mute)
AgoraChannel	int MuteRemoteVideoStream(uint userId, bool mute)
AgoraChannel	int Publish()
AgoraChannel	int RenewToken(string token);
AgoraChannel	int SetAudioProfile(AUDIO_PROFILE_TYPE audioProfile, AUDIO_SCENARIO_TYPE scenario)
AgoraChannel	int SetChannelProfile(CHANNEL_PROFILE profile)
AgoraChannel	int SetClientRole(CLIENT_ROLE_TYPE role, ClientRoleOptions audienceLatencyLevel)
AgoraChannel	int SetClientRole(CLIENT_ROLE_TYPE role)
AgoraChannel	int SetLiveTranscoding(LiveTranscoding transcoding)
AgoraChannel	int SetLogFilter(uint filter);
AgoraChannel	int SetRemoteDefaultVideoStreamType(REMOTE_VIDEO_STREAM_TYPE streamType)
AgoraChannel	int SetRemoteVideoStreamType(uint uid,int streamType);
AgoraChannel	int SetRemoteVideoStreamType(uint userId, REMOTE_VIDEO_STREAM_TYPE streamType)
AgoraChannel	int SetVideoEncoderConfiguration(VideoEncoderConfiguration configuration)
AgoraChannel	int StartAudioMixing(string filePath, bool loopBack, bool replace,int cycle);
AgoraChannel	int StartChannelMediaRelay(ChannelMediaRelayConfiguration channelMediaRelayConfiguration)
AgoraChannel	int StartPreview()

AgoraChannel	int StopAudioMixing();
AgoraChannel	int StopChannelMediaRelay()
AgoraChannel	int StopPreview()
AgoraChannel	int SwitchCamera();
AgoraChannel	int Unpublish()
AgoraChannel	int UpdateChannelMediaRelay(ChannelMediaRelayConfiguration channelMediaRelayConfiguration)
AgoraChannel	string ChannelId()
AudioEffectManagerImpl	int GetEffectsVolume();
AudioEffectManagerImpl	int PauseAllEffects();
AudioEffectManagerImpl	int PauseEffect(int soundId);
AudioEffectManagerImpl	int PlayEffect(int soundId, string filePath,int loopCount, double pitch, double pan,int gain, bool publish);
AudioEffectManagerImpl	int PreloadEffect(int soundId, string filePath);
AudioEffectManagerImpl	int ResumeEffect(int soundId);
AudioEffectManagerImpl	int SetEffectsVolume(int volume);
AudioEffectManagerImpl	int StopAllEffects();
AudioEffectManagerImpl	int StopEffect(int soundId);
AudioPlaybackDeviceManager	bool IsAudioPlaybackDeviceMute();
AudioPlaybackDeviceManager	int GetAudioPlaybackDevice(int index, IntPtr deviceName, IntPtr deviceId);
AudioPlaybackDeviceManager	int GetAudioPlaybackDeviceVolume();
AudioPlaybackDeviceManager	int SetAudioPlaybackDevice(string deviceId);

AudioPlaybackDeviceManager	int SetAudioPlaybackDeviceMute(bool mute);
AudioRecordingDeviceManager	bool IsAudioRecordingDeviceMute();
AudioRecordingDeviceManager	int GetAudioRecordingDevice(int index, IntPtr deviceName,IntPtr deviceId);
AudioRecordingDeviceManager	int GetAudioRecordingDeviceCount();
AudioRecordingDeviceManager	int GetAudioRecordingDeviceVolume();
AudioRecordingDeviceManager	int GetCurrentRecordingDevice(IntPtr deviceId);
AudioRecordingDeviceManager	int GetCurrentRecordingDeviceInfo(IntPtr deviceName, IntPtr deviceId);
AudioRecordingDeviceManager	int SetAudioRecordingDevice(string deviceId);
AudioRecordingDeviceManager	int SetAudioRecordingDeviceMute(bool mute);
IRtcEngine	AddVideoWatermark(RtcImage rtcImage)
IRtcEngine	AddVideoWatermark(string watermarkUrl, WatermarkOptions watermarkOptions)
IRtcEngine	int AdjustAudioMixingPlayoutVolume(int volume);
IRtcEngine	int AdjustAudioMixingPublishVolume(int volume);
IRtcEngine	int AdjustAudioMixingVolume(int volume);
IRtcEngine	int AdjustPlaybackSignalVolume(int volume);
IRtcEngine	int AdjustRecordingSignalVolume(int volume);
IRtcEngine	int AdjustUserPlaybackSignalVolume(uint uid, int volume);
IRtcEngine	int ClearVideoWatermarks();
IRtcEngine	int DisableVideoObserver();
IRtcEngine	int EnableDualStreamMode(bool enabled);
IRtcEngine	int GetAudioMixingCurrentPosition();
IRtcEngine	int GetAudioMixingDuration();
IRtcEngine	int GetAudioMixingPlayoutVolume();

IRtcEngine	int GetAudioMixingPublishVolume();
IRtcEngine	int MuteRemoteAudioStream(uint uid, bool mute);
IRtcEngine	int MuteRemoteVideoStream(uint uid, bool mute);
IRtcEngine	int PauseAudioMixing();
IRtcEngine	int RemovePublishStreamUrl(string url);
IRtcEngine	int SetAudioMixingPitch(int pitch);
IRtcEngine	int SetAudioMixingPosition(int pos)
IRtcEngine	int SetAudioMixingPosition(int pos);
IRtcEngine	int SetCameraCapturerConfiguration(CameraCapturerConfiguration cameraCaptureConfiguration)
IRtcEngine	int SetDefaultMuteAllRemoteAudioStreams(bool mute);
IRtcEngine	int SetDefaultMuteAllRemoteVideoStreams(bool mute);
IRtcEngine	int SetLocalPublishFallbackOption(STREAM_FALLBACK_OPTIONS option)
IRtcEngine	int SetMirrorApplied(bool wheatherApply)
IRtcEngine	int SetMixedAudioFrameParameters(int sampleRate, int samplesPerCall)
IRtcEngine	int SetMultiChannelWant(bool multiChannelWant)
IRtcEngine	int SetMultiChannelWant(bool multiChannelWant)
IRtcEngine	int SetParameters(string parameters)
IRtcEngine	int SetPlaybackDeviceVolume(int volume);
IRtcEngine	int SetRemoteDefaultVideoStreamType(REMOTE_VIDEO_STREAM_TYPE remoteVideoStreamType)
IRtcEngine	int SetRemoteSubscribeFallbackOption(STREAM_FALLBACK_OPTIONS option)
IRtcEngine	int SetRemoteUserPriority(uint uid, PRIORITY_TYPE userPriority)
IRtcEngine	int SetVoiceOnlyMode (bool enable)
IRtcEngine	int StartAudioRecording(string filePath,int quality);
IRtcEngine	int StartChannelMediaRelay(ChannelMediaRelayConfiguration mediaRelayConfiguration)
IRtcEngine	int StopAudioRecording();
IRtcEngine	int StopChannelMediaRelay();

	Int StartScreenCaptureForWeb(bool loopbackEnabled) Int StartNewScreenCaptureForWeb(uint uid, bool loopbackEnabled) int StopScreenCapture();
IRtcEngine	int SwitchChannel()
VideoDeviceManager	bool CreateAVideoDeviceManager();
VideoDeviceManager	bool ReleaseAVideoDeviceManager();
VideoDeviceManager	int GetCurrentVideoDevice(IntPtr deviceId);
VideoDeviceManager	int GetVideoDevice (int index, ref string deviceName, ref string deviceId)
VideoDeviceManager	int GetVideoDeviceCount();
VideoDeviceManager	int SetVideoDevice (string deviceId)
VideoRawDataManager	int RegisterVideoRawDataObserver()
VideoRawDataManager	int SetOnCaptureVideoFrameCallback(OnCaptureVideoFrameHandler action)

Unsupported features:

- AudioRawDataManager
- VideoRawDataManager
- MetaDataObserver
- PacketObserver
- PushAudioFrame
- Spatial Audio

Appendix B - Agora WebGL Programming Guide

See separate Document from this link.

Appendix C - Agora WebGL Build and Run Guide

See separate Document from this link.

This README is kept online for up to date changes.