Photon Voice

v2.31

Generated by Doxygen 1.8.17

Chapter 1

Main Page

Photon Voice 2 has three key classes:

- Photon.Voice.Unity.VoiceConnection (extends `Photon.Realtime.ConnectionHandler`)
- Photon.Voice.Unity.Recorder
- Photon.Voice.Unity.Speaker

If you also use the integration with PUN 2, we added two components for ease-of-use and more convenience:

- Photon.Voice.PUN.PhotonVoiceNetwork
- Photon.Voice.PUN.PhotonVoiceView

Photon Voice 2 also comes with a WebRTC based DSP (Photon.Voice.Unity.WebRtcAudioDsp using Photon.Voice.WebRTCAudioProcessor).

Read more in the official documentation `here`.
You can download Photon Voice 2 `here`.

Chapter 2

Namespace Documentation

2.1 Photon Namespace Reference

2.2 Photon.Voice Namespace Reference

Classes

- class AudioDesc
- class AudioInChangeNotifierNotSupported
- class **AudioInEnumeratorNotSupported**
- class AudioOutDelayControl
- class AudioSyncBuffer
- class AudioUtil

  Collection of Audio Utility functions and classes.
- class BufferReaderPushAdapter

  Simple BufferReaderPushAdapterBase< T > implementation using a single buffer and synchronous LocalVoiceFramed< T >.PushData
- class BufferReaderPushAdapterAsyncPool

  BufferReaderPushAdapter< T > implementation using asynchronous LocalVoiceFramed< T >.PushDataAsync.
- class BufferReaderPushAdapterAsyncPoolCopy

  BufferReaderPushAdapter< T > implementation using asynchronous LocalVoiceFramed< T >.PushDataAsync(T[ ]) and data copy.
- class BufferReaderPushAdapterAsyncPoolFloatToShort

  BufferReaderPushAdapter< T > implementation using asynchronous LocalVoiceFramed< T >.PushDataAsync, converting float samples to short.
- class BufferReaderPushAdapterAsyncPoolShortToFloat

  BufferReaderPushAdapter< T > implementation using asynchronous LocalVoiceFramed< T >.PushDataAsync, converting short samples to float.
- class BufferReaderPushAdapterBase

  Adapter base reading data from IDataReader< T >.Read and pushing it to LocalVoice.
- class DeviceEnumeratorBase
- class **DeviceEnumeratorNotSupported**
- class DeviceFeatures
- struct DeviceInfo
- class FactoryPrimitiveArrayPool

  PrimitiveArrayPool< T > as wrapped in object factory interface.
- class FactoryReusableArray

Array factory returnig the same array instance as long as it requested with the same array length. If length changes, new array instance created.

- struct Flip
- struct FrameBuffer
- class FrameOut
- class Framer

  Utility class to re-frame audio packets.
- interface IAudioDesc

  Audio Source interface.
- interface IAudioInChangeNotifier
- interface IAudioOut
- interface IAudioPusher

  Audio Pusher interface.
- interface IAudioReader

  Audio Reader interface.
- interface IDataReader

  Interface for pulling data, in case this is more appropriate than pushing it.
- interface IDecoder

  Generic decoder interface.
- interface IDecoderDirect

  Interface for an decoder which outputs data via explicit call.
- interface IDeviceEnumerator
- interface IEncoder

  Generic encoder interface.
- interface IEncoderDirect

  Interface for an encoder which consumes input data via explicit call.
- interface IEncoderDirectImage

  Interface for an encoder which consumes images via explicit call.
- interface ILocalVoiceAudio

  Interface for an outgoing audio stream.
- interface ILogger
- struct ImageBufferInfo
- class ImageBufferNative
- class ImageBufferNativeAlloc
- class ImageBufferNativeGCHandleSinglePlane
- class ImageBufferNativePool
- interface IProcessor

  Audio Processor interface.
- interface IResettable
- interface IServiceable

  Interface for classes that want their Service() function to be called regularly in the context of a LocalVoice.
- interface IVoiceTransport
- class LoadBalancingTransport

  Extends LoadBalancingClient with media streaming functionality.
- class LoadBalancingTransport2

  Variant of LoadBalancingTransport. Aims to be non-alloc at the cost of breaking compatibility with older clients.
- class LocalVoice

  Represents outgoing data stream.
- class LocalVoiceAudio

  Outgoing audio stream.
- class LocalVoiceAudioDummy

  Dummy LocalVoiceAudio

- class LocalVoiceAudioFloat

  Specialization of LocalVoiceAudio< T > for float audio
- class LocalVoiceAudioShort

  Specialization of LocalVoiceAudio< T > for short audio
- class LocalVoiceFramed

  Typed re-framing LocalVoice
- class LocalVoiceFramedBase

  Typed re-framing LocalVoice
- interface ObjectFactory

  Uniform interface to ObjectPool< TType, TInfo > and single reusable object.
- class ObjectPool

  Generic Pool to re-use objects of a certain type (TType) that optionally match a certain property or set of properties (TInfo).
- class OpusCodec
- class PhotonAppSettings

  Collection of connection-relevant settings, used internally by PhotonNetwork.ConnectUsingSettings.
- class **PhotonTransportProtocol**
- class Platform
- class PrimitiveArrayPool

  Pool of Arrays with components of type T, with ObjectPool info being the array's size.
- class RawCodec
- class **RemoteVoice**
- class RemoteVoiceInfo

  Information about a remote voice (incoming stream).
- struct RemoteVoiceOptions

  Event Actions and other options for a remote voice (incoming stream).
- class **SpacingProfile**
- class UnsupportedCodecException

  Exception thrown if an unsupported codec is encountered.
- class UnsupportedPlatformException

  Exception thrown if an unsupported platform is encountered.
- class UnsupportedSampleTypeException

  Exception thrown if an unsupported audio sample type is encountered.
- class **Util**
- class **VideoInEnumeratorNotSupported**
- class VoiceClient

  Voice client interact with other clients on network via IVoiceTransport.
- class VoiceEvent
- struct VoiceInfo

  Describes stream properties.
- class WebRTCAudioLib
- class WebRTCAudioProcessor

Enumerations

- enum **CameraFacing**
- enum AudioSampleType

  The type of samples used for audio processing.
- enum **FrameFlags** : byte
- enum Codec

  Enum for Media Codecs supported by PhotonVoice.
- enum **ImageFormat**
- enum **Rotation**

2.2.1 Enumeration Type Documentation

2.2.1.1 AudioSampleType

```
enum AudioSampleType  [strong]
```

The type of samples used for audio processing.

2.2.1.2 Codec

```
enum Codec  [strong]
```

Enum for Media Codecs supported by PhotonVoice.

Transmitted in VoiceInfo. Do not change the values of this Enum!

Enumerator

AudioOpus	OPUS audio

2.3 Photon.Voice.FMOD Namespace Reference

Classes

- class AudioInEnumerator
- class AudioInReader
- class AudioOut
- class AudioOutEvent

2.4 Photon.Voice.Fusion Namespace Reference

Classes

- class FusionVoiceClient
- class VoiceNetworkObject

Typedefs

- using **PhotonAppSettings** = global::Fusion.Photon.Realtime.PhotonAppSettings

2.5 Photon.Voice.IOS Namespace Reference

Classes

- class AudioInChangeNotifier
- class AudioInPusher
- class AudioInReader
- struct AudioSessionParameters
- class AudioSessionParametersPresets
- class MonoPInvokeCallbackAttribute

Enumerations

- enum AudioSessionCategory
- enum AudioSessionMode
- enum AudioSessionCategoryOption

2.5.1 Enumeration Type Documentation

2.5.1.1 AudioSessionCategory

```
enum AudioSessionCategory  [strong]
```

Enumerator

Ambient	Use this category for background sounds such as rain, car engine noise, etc. Mixes with other music. API_AVAILABLE(ios(3.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
SoloAmbient	Use this category for background sounds. Other music will stop playing. API_AVAILABLE(ios(3.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
Playback	Use this category for music tracks. API_AVAILABLE(ios(3.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
Record	Use this category when recording audio. API_AVAILABLE(ios(3.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
PlayAndRecord	Use this category when recording and playing back audio. API_AVAILABLE(ios(3.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
AudioProcessing	Use this category when using a hardware codec or signal processor while not playing or recording audio. API_DEPRECATED("No longer supported", ios(3.0, 10.0)) API_UNAVAILABLE(watchos, tvos) API_UNAVAILABLE(macos);
MultiRoute	Use this category to customize the usage of available audio accessories and built-in audio hardware. For example, this category provides an application with the ability to use an available USB output and headphone output simultaneously for separate, distinct streams of audio data. Use of this category by an application requires a more detailed knowledge of, and interaction with, the capabilities of the available audio routes. May be used for input, output, or both. Note that not all output types and output combinations are eligible for multi-route. Input is limited to the last-in input port. Eligible inputs consist of the following: AVAudioSessionPortUSBAudio, AVAudioSessionPortHeadsetMic, and AVAudioSessionPortBuiltInMic. Eligible outputs consist of the following: AVAudioSessionPortUSBAudio, AVAudioSessionPortLineOut, AVAudioSessionPortHeadphones, AVAudioSessionPortHDMI, and AVAudioSessionPortBuiltInSpeaker. Note that AVAudioSessionPortBuiltInSpeaker is only allowed to be used when there are no other eligible outputs connected. API_AVAILABLE(ios(6.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);

2.5.1.2 AudioSessionCategoryOption

```
enum AudioSessionCategoryOption  [strong]
```

Enumerator

MixWithOthers	This allows an application to set whether or not other active audio apps will be interrupted or mixed with when your app's audio session goes active. The typical cases are: (1) AVAudioSessionCategoryPlayAndRecord or AVAudioSessionCategoryMultiRoute this will default to false, but can be set to true. This would allow other applications to play in the background while an app had both audio input and output enabled (2) AVAudioSessionCategoryPlayback this will default to false, but can be set to true. This would allow other applications to play in the background, but an app will still be able to play regardless of the setting of the ringer switch (3) Other categories this defaults to false and cannot be changed (that is, the mix with others setting of these categories cannot be overridden. An application must be prepared for setting this property to fail as behaviour may change in future releases. If an application changes their category, they should reassert the option (it is not sticky across category changes). MixWithOthers is only valid with AVAudioSessionCategoryPlayAndRecord, AVAudioSessionCategoryPlayback, and AVAudioSessionCategoryMultiRoute
DuckOthers	This allows an application to set whether or not other active audio apps will be ducked when when your app's audio session goes active. An example of this is the Nike app, which provides periodic updates to its user (it reduces the volume of any music currently being played while it provides its status). This defaults to off. Note that the other audio will be ducked for as long as the current session is active. You will need to deactivate your audio session when you want full volume playback of the other audio. If your category is AVAudioSessionCategoryPlayback, AVAudioSessionCategoryPlayAndRecord, or AVAudioSessionCategoryMultiRoute, by default the audio session will be non-mixable and non-ducking. Setting this option will also make your category mixable with others (AVAudioSessionCategoryOptionMixWithOthers will be set). DuckOthers is only valid with AVAudioSessionCategoryAmbient, AVAudioSessionCategoryPlayAndRecord, AVAudioSessionCategoryPlayback, and AVAudioSessionCategoryMultiRoute
AllowBluetooth	This allows an application to change the default behaviour of some audio session categories with regards to showing bluetooth Hands-Free Profile (HFP) devices as available routes. The current category behavior is: (1) AVAudioSessionCategoryPlayAndRecord this will default to false, but can be set to true. This will allow a paired bluetooth HFP device to show up as an available route for input, while playing through the category-appropriate output (2) AVAudioSessionCategoryRecord this will default to false, but can be set to true. This will allow a paired bluetooth HFP device to show up as an available route for input (3) Other categories this defaults to false and cannot be changed (that is, enabling bluetooth for input in these categories is not allowed) An application must be prepared for setting this option to fail as behaviour may change in future releases. If an application changes their category or mode, they should reassert the override (it is not sticky across category and mode changes). AllowBluetooth is only valid with AVAudioSessionCategoryRecord and AVAudioSessionCategoryPlayAndRecord
DefaultToSpeaker	This allows an application to change the default behaviour of some audio session categories with regards to the audio route. The current category behavior is: (1) AVAudioSessionCategoryPlayAndRecord category this will default to false, but can be set to true. this will route to Speaker (instead of Receiver) when no other audio route is connected. (2) Other categories this defaults to false and cannot be changed (that is, the default to speaker setting of these categories cannot be overridden An application must be prepared for setting this property to fail as behaviour may change in future releases. If an application changes their category, they should reassert the override (it is not sticky across category and mode changes). DefaultToSpeaker is only valid with AVAudioSessionCategoryPlayAndRecord

2.5.1.3 AudioSessionMode

```
enum AudioSessionMode [strong]
```

Enumerator

Default	Modes modify the audio category in order to introduce behavior that is tailored to the specific use of audio within an application. Available in iOS 5.0 and greater. The default mode API_AVAILABLE(ios(5.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
VoiceChat	Only valid with AVAudioSessionCategoryPlayAndRecord. Appropriate for Voice over IP (VoIP) applications. Reduces the number of allowable audio routes to be only those that are appropriate for VoIP applications and may engage appropriate system-supplied signal processing. Has the side effect of setting AVAudioSessionCategoryOptionAllowBluetooth API_AVAILABLE(ios(5.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
VideoRecording	Only valid with AVAudioSessionCategoryPlayAndRecord or AVAudioSessionCategoryRecord. Modifies the audio routing options and may engage appropriate system-supplied signal processing. API_AVAILABLE(ios(5.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
Measurement	Appropriate for applications that wish to minimize the effect of system-supplied signal processing for input and/or output audio signals. API_AVAILABLE(ios(5.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
MoviePlayback	Engages appropriate output signal processing for movie playback scenarios. Currently only applied during playback over built-in speaker. API_AVAILABLE(ios(6.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);
VideoChat	Only valid with kAudioSessionCategory_PlayAndRecord. Reduces the number of allowable audio routes to be only those that are appropriate for video chat applications. May engage appropriate system-supplied signal processing. Has the side effect of setting AVAudioSessionCategoryOptionAllowBluetooth and AVAudioSessionCategoryOptionDefaultToSpeaker. API_AVAILABLE(ios(7.0), watchos(2.0), tvos(9.0)) API_UNAVAILABLE(macos);

2.6 Photon.Voice.MacOS Namespace Reference

Classes

- class AudioInChangeNotifier
- class AudioInEnumerator

  Enumerates microphones available on device.
- class AudioInPusher
- class AudioInReader
- class MonoPInvokeCallbackAttribute

2.7 Photon.Voice.PUN Namespace Reference

Classes

- class PhotonVoiceView

Component that should be attached to a networked PUN prefab that has PhotonView. It will bind remote Recorder with local Speaker of the same networked prefab. This component makes automatic voice stream routing easy for players' characters/avatars.

- class PunVoiceClient

  This class can be used to automatically sync client states between PUN and Voice. It also finds the Speaker component for a character's voice. For this to work attach a PhotonVoiceView next to the PhotonView of your player's prefab.

2.8 Photon.Voice.PUN.UtilityScripts Namespace Reference

Classes

- class VoiceDebugScript

  Utility script to be attached next to PhotonVoiceView & PhotonView on the player prefab to be network instantiated. Call voiceDebugScript.CantHearYou() on the networked object of the remote (or local) player if you can't hear the corresponding player.

2.9 Photon.Voice.Unity Namespace Reference

Classes

- class AndroidAudioInAEC
- class AndroidAudioInParameters
- class AudioChangesHandler

  This component is useful to handle audio device and config changes.
- class AudioClipWrapper
- class AudioInEnumerator
- class AudioOutCapture
- class Logger
- class MicWrapper
- class MicWrapperPusher
- class MicWrapperPusherOnAudioFilterRead
- class PhotonVoiceCreatedParams
- class Recorder

  Component representing outgoing audio stream in scene.
- class RecorderPreset
- class RemoteVoiceLink
- class Speaker
- class SpeakerAudioFilterRead
- class UnityAudioOut
- class UnityLogger
- class UnityMicrophone

  A wrapper around UnityEngine.Microphone to be able to safely use Microphone and compile for WebGL.
- class UnityVoiceClient

  Component that represents a Voice client and manages a simple Unity integration: a single Recorder and multiple remote speakers.
- class VideoInEnumerator
- class VoiceComponent
- class VoiceComponentImpl
- class VoiceConnection

  Component that represents a Voice client.
- class VoiceLogger
- class WebRtcAudioDsp

2.10 Photon.Voice.Unity.FMOD Namespace Reference

Classes

- class FMODRecorderSetup
- class SpeakerFMOD

2.11 Photon.Voice.Unity.UtilityScripts Namespace Reference

Classes

- class ConnectAndJoin
- class MicAmplifier
- class MicAmplifierFloat
- class MicAmplifierShort
- class MicrophonePermission

  Helper to request Microphone permission on Android or iOS.
- class PhotonVoiceLagSimulationGui
- class PhotonVoiceStatsGui

  Basic GUI to show traffic and health statistics of the connection to Photon, toggled by shift+tab.
- class SaveIncomingStreamToFile
- class SaveOutgoingStreamToFile
- class TestTone
- class WaveWriter

2.12 Photon.Voice.UWP Namespace Reference

Classes

- class AudioInEnumerator
- class AudioInPusher
- class CaptureDevice
- class DeviceEnumerator
- class VideoInEnumerator

Functions

- delegate void **MediaCaptureInitConmpleted** (MediaCapture mediaCpture, bool ok)

2.13 Photon.Voice.Windows Namespace Reference

Classes

- class AudioInEnumerator

  Enumerates microphones available on device.
- class MonoPInvokeCallbackAttribute
- class WindowsAudioInPusher

Chapter 3

Class Documentation

3.1 AndroidAudioInAEC Class Reference

Inherits IAudioPusher< short >, and IResettable.

Public Member Functions

- **AndroidAudioInAEC** (Voice.ILogger logger, bool enableAEC=false, bool enableAGC=false, bool enableNS=false)
- void **SetCallback** (Action< short[ ]> callback, ObjectFactory< short[ ], int > bufferFactory)
- void **Reset** ()
- void **Dispose** ()

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.2 AndroidAudioInParameters Class Reference

Public Attributes

- bool **EnableAEC** = false
- bool **EnableAGC** = false
- bool **EnableNS** = false

3.3 AudioChangesHandler Class Reference

This component is useful to handle audio device and config changes.

Inherits VoiceComponent.

Public Attributes

- bool HandleDeviceChange = true

  Try to react to device change notification when Recorder is started.
- bool HandleDeviceChangeIOS

  iOS: Try to react to device change notification when Recorder is started.
- bool HandleDeviceChangeAndroid

  Android: Try to react to device change notification when Recorder is started.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

3.3.1 Detailed Description

This component is useful to handle audio device and config changes.

3.3.2 Member Data Documentation

3.3.2.1 HandleDeviceChange

```
bool HandleDeviceChange = true
```

Try to react to device change notification when Recorder is started.

3.3.2.2 HandleDeviceChangeAndroid

```
bool HandleDeviceChangeAndroid
```

Android: Try to react to device change notification when Recorder is started.

3.3.2.3 HandleDeviceChangeIOS

```
bool HandleDeviceChangeIOS
```

iOS: Try to react to device change notification when Recorder is started.

3.4 AudioClipWrapper Class Reference

Inherits IAudioReader< float >.

Public Member Functions

- **AudioClipWrapper** (AudioClip audioClip)
- bool **Read** (float[ ] buffer)
- void **Dispose** ()

Properties

- bool **Loop** `[get, set]`
- int **SamplingRate** `[get]`
- int **Channels** `[get]`
- string **Error** `[get]`

3.5 AudioDesc Class Reference

Inherits IAudioDesc.

Public Member Functions

- **AudioDesc** (int samplingRate, int channels, string error)
- void **Dispose** ()

Properties

- int **SamplingRate** `[get]`
- int **Channels** `[get]`
- string **Error** `[get]`

3.6 AudioInChangeNotifier Class Reference

Inherits IAudioInChangeNotifier.

Public Member Functions

- **AudioInChangeNotifier** (Action callback, ILogger logger)
- void Dispose ()

  Disposes enumerator. Call it to free native resources.

Public Attributes

- bool **IsSupported** => true

Properties

- string Error [get]

  If not null, the enumerator is in invalid state.

3.6.1 Member Function Documentation

3.6.1.1 Dispose()

```
void Dispose ( )
```

Disposes enumerator. Call it to free native resources.

3.6.2 Property Documentation

3.6.2.1 Error

```
string Error [get]
```

If not null, the enumerator is in invalid state.

3.7 AudioInChangeNotifier Class Reference

Inherits IAudioInChangeNotifier.

Classes

- class MonoPInvokeCallbackAttribute

Public Member Functions

- **AudioInChangeNotifier** (Action callback, ILogger logger)
- void Dispose ()

  Disposes enumerator. Call it to free native resources.

Public Attributes

- bool **IsSupported** => true

Properties

- string Error [get]

  If not null, the enumerator is in invalid state.

3.7.1 Member Function Documentation

3.7.1.1 Dispose()

```
void Dispose ( )
```

Disposes enumerator. Call it to free native resources.

3.7.2 Property Documentation

3.7.2.1 Error

```
string Error [get]
```

If not null, the enumerator is in invalid state.

3.8 AudioInChangeNotifierNotSupported Class Reference

Inherits IAudioInChangeNotifier.

Public Member Functions

- **AudioInChangeNotifierNotSupported** (Action callback, ILogger logger)
- void **Dispose** ()

Public Attributes

- bool **IsSupported** => false

Properties

- string **Error** [get]

3.9 AudioInEnumerator Class Reference

Enumerates microphones available on device.

Inherits DeviceEnumeratorBase.

Public Member Functions

- **AudioInEnumerator** (ILogger logger)
- override void Refresh ()

  Refreshes the microphones list.
- override void Dispose ()

  Disposes enumerator. Call it to free native resources.

Additional Inherited Members

3.9.1 Detailed Description

Enumerates microphones available on device.

3.9.2 Member Function Documentation

3.9.2.1 Dispose()

```
override void Dispose ( )  [virtual]
```

Disposes enumerator. Call it to free native resources.

Implements DeviceEnumeratorBase.

3.9.2.2 Refresh()

```
override void Refresh ( )  [virtual]
```

Refreshes the microphones list.

Implements DeviceEnumeratorBase.

3.10 AudioInEnumerator Class Reference

Inherits DeviceEnumeratorBase.

Public Member Functions

- **AudioInEnumerator** (FMODLib.System coreSystem, ILogger logger)
- override void **Refresh** ()
- override void **Dispose** ()

Additional Inherited Members

3.11 AudioInEnumerator Class Reference

Inherits DeviceEnumeratorBase.

Public Member Functions

- **AudioInEnumerator** (ILogger logger)
- override void **Refresh** ()
- override void **Dispose** ()

Public Attributes

- override bool **IsSupported** => false

Properties

- override string **Error** `[get]`

Additional Inherited Members

3.12 AudioInEnumerator Class Reference

Inherits DeviceEnumerator.

Public Member Functions

- **AudioInEnumerator** (ILogger logger)

Additional Inherited Members

3.13 AudioInEnumerator Class Reference

Enumerates microphones available on device.

Inherits DeviceEnumeratorBase.

Public Member Functions

- **AudioInEnumerator** (ILogger logger)
- override void Refresh ()

  Refreshes the microphones list.
- override void Dispose ()

  Disposes enumerator. Call it to free native resources.

Additional Inherited Members

3.13.1 Detailed Description

Enumerates microphones available on device.

3.13.2 Member Function Documentation

3.13.2.1 Dispose()

```
override void Dispose ( )  [virtual]
```

Disposes enumerator. Call it to free native resources.

Implements DeviceEnumeratorBase.

3.13.2.2 Refresh()

```
override void Refresh ( )  [virtual]
```

Refreshes the microphones list.

Implements DeviceEnumeratorBase.

3.14 AudioInPusher Class Reference

Inherits IAudioPusher< float >, and IResettable.

Public Member Functions

- **AudioInPusher** (AudioSessionParameters sessParam, ILogger logger)
- void **SetCallback** (Action< float[ ]> callback, ObjectFactory< float[ ], int > bufferFactory)
- void **Reset** ()
- void **Dispose** ()

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.15 AudioInPusher Class Reference

Inherits IAudioPusher< float >.

Public Member Functions

- **AudioInPusher** (int deviceID, ILogger logger)
- void **SetCallback** (Action< float[ ]> callback, ObjectFactory< float[ ], int > bufferFactory)
- void **Dispose** ()

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.16 AudioInPusher Class Reference

Inherits IAudioPusher< short >.

Public Member Functions

- **AudioInPusher** (ILogger logger, int samplingRate, int channels, string deviceID)
- void **SetCallback** (Action< short[ ]> callback, ObjectFactory< short[ ], int > bufferFactory)
- ArraySegment< byte > **DequeueOutput** (out FrameFlags flags)
- void **EndOfStream** ()
- I **GetPlatformAPI< I >** ()
- void **Dispose** ()

Properties

- int **SamplingRate** `[get]`
- int Channels `[get]`

  Number of channels in the audio signal.
- string **Error** `[get]`
- bool **ErrorAccess** `[get]`

3.16.1 Property Documentation

3.16.1.1 Channels

```
int Channels  [get]
```

Number of channels in the audio signal.

3.17 AudioInReader Class Reference

Inherits IAudioReader< float >, and IResettable.

Public Member Functions

- **AudioInReader** (AudioSessionParameters sessParam, ILogger logger)
- void **Reset** ()
- void **Dispose** ()
- bool **Read** (float[ ] buf)

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.18 AudioInReader Class Reference

Inherits IAudioReader< float >.

Public Member Functions

- **AudioInReader** (int deviceID, ILogger logger)
- void **Dispose** ()
- bool **Read** (float[ ] buf)

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.19 AudioInReader< T > Class Template Reference

Inherits IAudioReader< T >.

Public Member Functions

- **AudioInReader** (FMODLib.System coreSystem, int device, int suggestedFrequency, ILogger logger)
- void **Dispose** ()
- bool Read (T[ ] readBuf)

  Fill full given frame buffer with source uncompressed data or return false if not enough such data.

Public Attributes

- bool **isRecording**

Properties

- int? **SamplingRate** `[get]`
- int? **Channels** `[get]`
- string **Error** `[get]`

3.19.1 Member Function Documentation

3.19.1.1 Read()

```
bool Read (
            T[ ] buffer )
```

Fill full given frame buffer with source uncompressed data or return false if not enough such data.

Parameters

buffer	Buffer to fill.

Returns

True if buffer was filled successfully, false otherwise.

Implements IDataReader< T >.

3.20 AudioOut< T > Class Template Reference

Inherits AudioOutDelayControl< T >.

Inherited by AudioOutEvent< T >.

Public Member Functions

- **AudioOut** (FMODLib.System coreSystem, PlayDelayConfig playDelayConfig, ILogger logger, string logPrefix, bool debugInfo)
- override void **OutCreate** (int samplingRate, int channels, int bufferSamples)
- override void **OutStart** ()
- override void **OutWrite** (T[ ] frame, int offsetSamples)
- override void **Stop** ()

Protected Attributes

- readonly int **sizeofT** = Marshal.SizeOf(default(T))

Properties

- FMODLib.Sound **Sound** `[get]`
- FMODLib.Channel **Channel** `[get]`
- override long **OutPos** `[get]`
- string **Error** `[get]`

3.21 AudioOutCapture Class Reference

Inherits MonoBehaviour.

Events

- Action< float[ ], int > **OnAudioFrame**

3.22 AudioOutDelayControl Class Reference

Inherited by AudioOutDelayControl< T >.

Classes

- struct PlayDelayConfig

3.23 AudioOutDelayControl Class Reference

Inherited by AudioOutDelayControl< T >.

Classes

- struct PlayDelayConfig

3.24 AudioOutEvent< T > Class Template Reference

Inherits AudioOut< T >.

Public Member Functions

- **AudioOutEvent** (FMODLib.System coreSystem, FMODLib.Studio.EventInstance fmodEvent, PlayDelayConfig playDelayConfig, ILogger logger, string logPrefix, bool debugInfo)
- override void **OutStart** ()
- override void **Stop** ()

Properties

- override long **OutPos** `[get]`

Additional Inherited Members

3.25 AudioSessionParameters Struct Reference

Public Member Functions

- int **CategoryOptionsToInt** ()
- override string **ToString** ()

Public Attributes

- AudioSessionCategory **Category**
- AudioSessionMode **Mode**
- AudioSessionCategoryOption[ ] **CategoryOptions**

3.26 AudioSessionParametersPresets Class Reference

Static Public Attributes

- static AudioSessionParameters **Game**
- static AudioSessionParameters **VoIP**

3.26.1 Member Data Documentation

3.26.1.1 Game

```
AudioSessionParameters Game  [static]
```

Initial value:

```
= new AudioSessionParameters()
        {
            Category = AudioSessionCategory.PlayAndRecord,
            Mode = AudioSessionMode.Default,
            CategoryOptions = new AudioSessionCategoryOption[] {
      AudioSessionCategoryOption.DefaultToSpeaker, AudioSessionCategoryOption.AllowBluetooth }
        }
```

3.26.1.2 VoIP

```
AudioSessionParameters VoIP  [static]
```

Initial value:

```
= new AudioSessionParameters()
        {
            Category = AudioSessionCategory.PlayAndRecord,
            Mode = AudioSessionMode.VoiceChat,


            CategoryOptions = new AudioSessionCategoryOption[] { AudioSessionCategoryOption.AllowBluetooth }
        }
```

3.27 AudioSyncBuffer< T > Class Template Reference

Inherits IAudioOut< T >.

Public Member Functions

- **AudioSyncBuffer** (int playDelayMs, ILogger logger, string logPrefix, bool debugInfo)
- void **Start** (int sampleRate, int channels, int frameSamples)
- void **Service** ()
- void **Read** (T[ ] outBuf, int outChannels, int outSampleRate)
- void **Push** (T[ ] frame)
- void **Flush** ()
- void **Stop** ()

Static Public Attributes

- const int **FRAME_POOL_CAPACITY** = 50

Properties

- int **Lag** `[get]`
- bool **IsPlaying** `[get]`

3.28 AudioUtil Class Reference

Collection of Audio Utility functions and classes.

Classes

- class GeneratorPusher

  IAudioPusher that provides a constant tone signal.
- class GeneratorReader
- interface ILevelMeter

  Audio Level Metering interface.
- interface IVoiceDetector

  Voice Activity Detector interface.
- class LevelMeter

  Audio Level Meter.
- class LevelMeterDummy

  Dummy Audio Level Meter that doesn't actually do anything.
- class LevelMeterFloat

  LevelMeter specialization for float audio.
- class LevelMeterShort

  LevelMeter specialization for short audio.
- class Resampler

  Sample-rate conversion Audio Processor.
- class TempoUp
- class ToneAudioPusher

  IAudioPusher that provides a constant tone signal.
- class ToneAudioReader

  IAudioReader that provides a constant tone signal.
- class VoiceDetector

  Simple voice activity detector triggered by signal level.
- class VoiceDetectorCalibration

  Calibration Utility for Voice Detector
- class VoiceDetectorDummy

  Dummy VoiceDetector that doesn't actually do anything.
- class VoiceDetectorFloat

  VoiceDetector specialization for float audio.
- class VoiceDetectorShort

  VoiceDetector specialization for float audio.
- class VoiceLevelDetectCalibrate

  Utility Audio Processor Voice Detection Calibration.
- class WaveformAudioPusher

  IAudioPusher that provides the given waveform.
- class WaveformAudioReader

  IAudioReader that provides the given waveform.

Static Public Member Functions

- static int **ToneToBuf**< **T** > (T[ ] buf, long timeSamples, int channels, double amp, double k, double phase↩ Mod=0)
- static int **ToneToBuf**< **T** > (T[ ] buf, int offset, int length, long timeSamples, int channels, double amp, double k, double phaseMod=0)
- static int **WaveformToBuf**< **T** > (T[ ] buf, T[ ] waveform, long timePos)
- static void Resample< T > (T[ ] src, T[ ] dst, int dstCount, int channels)

  Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer.
- static void **Resample**< **T** > (T[ ] src, int srcOffset, int srcCount, T[ ] dst, int dstOffset, int dstCount, int channels)
- static void **Resample**< **T** > (T[ ] src, int srcOffset, int srcCount, int srcChannels, T[ ] dst, int dstOffset, int dstCount, int dstChannels)
- static void ResampleAndConvert (short[ ] src, float[ ] dst, int dstCount, int channels)

  Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer, and convert short to float samples along the way.
- static void ResampleAndConvert (float[ ] src, short[ ] dst, int dstCount, int channels)

  Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer, and convert float to short samples along the way.
- static void Convert (float[ ] src, short[ ] dst, int dstCount)

  Convert audio buffer from float to short samples.
- static void Convert (short[ ] src, float[ ] dst, int dstCount)

  Convert audio buffer from short to float samples.
- static void ForceToStereo< T > (T[ ] src, T[ ] dst, int srcChannels)

  Convert audio buffer with arbitrary number of channels to stereo.

3.28.1 Detailed Description

Collection of Audio Utility functions and classes.

3.28.2 Member Function Documentation

3.28.2.1 Convert() [1/2]

```
static void Convert (
              float[ ] src,
              short[ ] dst,
              int dstCount )  [static]
```

Convert audio buffer from float to short samples.

Parameters

src	Source buffer.
dst	Destination buffer.
dstCount	Size of destination buffer (in total samples), source buffer must be of same length or longer.

3.28.2.2 Convert() [2/2]

```
static void Convert (
             short[] src,
             float[] dst,
             int dstCount )  [static]
```

Convert audio buffer from short to float samples.

Parameters

src	Source buffer.
dst	Destination buffer.
dstCount	Size of destination buffer (in total samples), source buffer must be of same length or longer.

3.28.2.3 ForceToStereo< T >()

```
static void ForceToStereo< T > (
             T[] src,
             T[] dst,
             int srcChannels )  [static]
```

Convert audio buffer with arbitrary number of channels to stereo.

For mono sources (srcChannels==1), the signal will be copied to both Left and Right stereo channels. For all others, the first two available channels will be used, any other channels will be discarded.

Parameters

src	Source buffer.
dst	Destination buffer.
srcChannels	Number of (interleaved) channels in src.

3.28.2.4 Resample< T >()

```
static void Resample< T > (
             T[] src,
             T[] dst,
             int dstCount,
             int channels )  [static]
```

Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer.

This implements a primitive nearest-neighbor resampling algorithm for an arbitrary number of channels.

Parameters

src	Source buffer.
dst	Destination buffer.
dstCount	Target size of destination buffer (in samples per channel).
channels	Number of channels in the signal (1=mono, 2=stereo). Must be $>$ 0.

3.28.2.5 ResampleAndConvert() [1/2]

```
static void ResampleAndConvert (
             float[] src,
             short[] dst,
             int dstCount,
             int channels )  [static]
```

Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer, and convert float to short samples along the way.

This implements a primitive nearest-neighbor resampling algorithm for an arbitrary number of channels.

Parameters

src	Source buffer.
dst	Destination buffer.
dstCount	Target size of destination buffer (in samples per channel).
channels	Number of channels in the signal (1=mono, 2=stereo). Must be $>$ 0.

3.28.2.6 ResampleAndConvert() [2/2]

```
static void ResampleAndConvert (
             short[] src,
             float[] dst,
             int dstCount,
             int channels )  [static]
```

Resample audio data so that the complete src buffer fits into dstCount samples in the dst buffer, and convert short to float samples along the way.

This implements a primitive nearest-neighbor resampling algorithm for an arbitrary number of channels.

Parameters

src	Source buffer.
dst	Destination buffer.
dstCount	Target size of destination buffer (in samples per channel).
channels	Number of channels in the signal (1=mono, 2=stereo). Must be $>$ 0.

3.29 BufferReaderPushAdapter< T > Class Template Reference

Simple BufferReaderPushAdapterBase<T> implementation using a single buffer and synchronous LocalVoiceFramed<T>.PushData

Inherits BufferReaderPushAdapterBase< T >.

Public Member Functions

- BufferReaderPushAdapter (LocalVoice localVoice, IDataReader< T > reader)

  Create a new BufferReaderPushAdapter instance
- override void Service (LocalVoice localVoice)

  Do the actual data read/push.

Protected Attributes

- T[ ] **buffer**

3.29.1 Detailed Description

Simple BufferReaderPushAdapterBase<T> implementation using a single buffer and synchronous LocalVoiceFramed<T>.PushData

3.29.2 Constructor & Destructor Documentation

3.29.2.1 BufferReaderPushAdapter()

```
BufferReaderPushAdapter (
            LocalVoice localVoice,
            IDataReader< T > reader )
```

Create a new BufferReaderPushAdapter instance

Parameters

localVoice	LocalVoice instance to push data to.
reader	DataReader to read from.

3.29.3 Member Function Documentation

3.29.3.1 Service()

```
override void Service (
            LocalVoice localVoice )  [virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to.

Implements BufferReaderPushAdapterBase< T >.

3.30 BufferReaderPushAdapterAsyncPool< T > Class Template Reference

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync.

Inherits BufferReaderPushAdapterBase< T >.

Public Member Functions

- BufferReaderPushAdapterAsyncPool (LocalVoice localVoice, IDataReader< T > reader)

    Create a new BufferReaderPushAdapter instance
- override void Service (LocalVoice localVoice)

    Do the actual data read/push.

Additional Inherited Members

3.30.1 Detailed Description

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync.

Acquires a buffer from pool before each Read, releases buffer after last Read (brings Acquire/Release overhead). Expects localVoice to be a LocalVoiceFramed<T> of same T.

3.30.2 Constructor & Destructor Documentation

3.30.2.1 BufferReaderPushAdapterAsyncPool()

```
BufferReaderPushAdapterAsyncPool (
            LocalVoice localVoice,
            IDataReader< T > reader )
```

Create a new BufferReaderPushAdapter instance

Parameters

localVoice	LocalVoice instance to push data to.
reader	DataReader to read from.

3.30.3 Member Function Documentation

3.30.3.1 Service()

```
override void Service (
            LocalVoice localVoice )  [virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to. Must be a LocalVoiceFramed<T> of same T.

Implements BufferReaderPushAdapterBase< T >.

3.31 BufferReaderPushAdapterAsyncPoolCopy< T > Class Template Reference

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync(T[ ]) and data copy.

Inherits BufferReaderPushAdapterBase< T >.

Public Member Functions

- BufferReaderPushAdapterAsyncPoolCopy (LocalVoice localVoice, IDataReader< T > reader)

  Create a new BufferReaderPushAdapter instance
- override void Service (LocalVoice localVoice)

  Do the actual data read/push.

Protected Attributes

- T[ ] **buffer**

3.31.1 Detailed Description

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync(T[ ]) and data copy.

Reads data to preallocated buffer, copies it to buffer from pool before pushing. Compared with BufferReaderPushAdapterAsyncPool<T this avoids one pool Acquire/Release cycle at the cost of a buffer copy. Expects localVoice to be a LocalVoiceFramed<T> of same T.

3.31.2 Constructor & Destructor Documentation

3.31.2.1 BufferReaderPushAdapterAsyncPoolCopy()

```
BufferReaderPushAdapterAsyncPoolCopy (
             LocalVoice localVoice,
             IDataReader< T > reader )
```

Create a new BufferReaderPushAdapter instance

Parameters

localVoice	LocalVoice instance to push data to.
reader	DataReader to read from.

3.31.3 Member Function Documentation

3.31.3.1 Service()

```
override void Service (
             LocalVoice localVoice )  [virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to. Must be a LocalVoiceFramed<T> of same T.

Implements BufferReaderPushAdapterBase< T >.

3.32 BufferReaderPushAdapterAsyncPoolFloatToShort Class Reference

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync, converting float samples to short.

Inherits BufferReaderPushAdapterBase< float >.

Public Member Functions

- BufferReaderPushAdapterAsyncPoolFloatToShort (LocalVoice localVoice, IDataReader< float > reader)

  Create a new BufferReaderPushAdapter instance
- override void Service (LocalVoice localVoice)

  Do the actual data read/push.

Additional Inherited Members

3.32.1 Detailed Description

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync, converting float samples to short.

This adapter works exactly like BufferReaderPushAdapterAsyncPool<T>, but it converts float samples to short. Acquires a buffer from pool before each Read, releases buffer after last Read.

Expects localVoice to be a LocalVoiceFramed<T> of same T.

3.32.2 Constructor & Destructor Documentation

3.32.2.1 BufferReaderPushAdapterAsyncPoolFloatToShort()

```
BufferReaderPushAdapterAsyncPoolFloatToShort (
            LocalVoice localVoice,
            IDataReader< float > reader )
```

Create a new BufferReaderPushAdapter instance

Parameters

localVoice	LocalVoice instance to push data to.
reader	DataReader to read from.

3.32.3 Member Function Documentation

3.32.3.1 Service()

```
override void Service (
            LocalVoice localVoice )  [virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to. Must be a LocalVoiceFramed<T> of same T.

Implements BufferReaderPushAdapterBase< float >.

3.33 BufferReaderPushAdapterAsyncPoolShortToFloat Class Reference

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync, converting short samples to float.

Inherits BufferReaderPushAdapterBase< short >.

Public Member Functions

- BufferReaderPushAdapterAsyncPoolShortToFloat (LocalVoice localVoice, IDataReader< short > reader)

  Create a new BufferReaderPushAdapter instance
- override void Service (LocalVoice localVoice)

  Do the actual data read/push.

Additional Inherited Members

3.33.1 Detailed Description

BufferReaderPushAdapter<T> implementation using asynchronous LocalVoiceFramed<T>.PushDataAsync, converting short samples to float.

This adapter works exactly like BufferReaderPushAdapterAsyncPool<T>, but it converts short samples to float. Acquires a buffer from pool before each Read, releases buffer after last Read.

Expects localVoice to be a LocalVoiceFramed<T> of same T.

3.33.2 Constructor & Destructor Documentation

3.33.2.1 BufferReaderPushAdapterAsyncPoolShortToFloat()

```
BufferReaderPushAdapterAsyncPoolShortToFloat (
            LocalVoice localVoice,
            IDataReader< short > reader )
```

Create a new BufferReaderPushAdapter instance

Parameters

localVoice	LocalVoice instance to push data to.
reader	DataReader to read from.

3.33.3 Member Function Documentation

3.33.3.1 Service()

```
override void Service (
            LocalVoice localVoice )  [virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to. Must be a LocalVoiceFramed<T> of same T.

Implements BufferReaderPushAdapterBase< short >.

3.34 BufferReaderPushAdapterBase< T > Class Template Reference

Adapter base reading data from IDataReader<T>.Read and pushing it to LocalVoice.

Inherits IServiceable.

Inherited by BufferReaderPushAdapter< T >, BufferReaderPushAdapterAsyncPool< T >, and BufferReaderPushAdapterAsyncPool

Public Member Functions

- abstract void Service (LocalVoice localVoice)

  Do the actual data read/push.
- BufferReaderPushAdapterBase (IDataReader< T > reader)

  Create a new BufferReaderPushAdapterBase instance
- void Dispose ()

  Release resources associated with this instance.

Protected Attributes

- IDataReader< T > **reader**

3.34.1 Detailed Description

Adapter base reading data from IDataReader<T>.Read and pushing it to LocalVoice.

Use this with a LocalVoice of same T type.

3.34.2 Constructor & Destructor Documentation

3.34.2.1 BufferReaderPushAdapterBase()

```
BufferReaderPushAdapterBase (
             IDataReader< T > reader )
```

Create a new BufferReaderPushAdapterBase instance

Parameters

reader	DataReader to read from.

3.34.3 Member Function Documentation

3.34.3.1 Dispose()

```
void Dispose ( )
```

Release resources associated with this instance.

3.34.3.2 Service()

```
abstract void Service (
             LocalVoice localVoice )  [pure virtual]
```

Do the actual data read/push.

Parameters

localVoice	LocalVoice instance to push data to.

Implements IServiceable.

Implemented in BufferReaderPushAdapterAsyncPoolShortToFloat, BufferReaderPushAdapterAsyncPoolFloatToShort, BufferReaderPushAdapterAsyncPoolCopy< T >, BufferReaderPushAdapterAsyncPool< T >, and BufferReaderPushAdapter< T >.

3.35 CaptureDevice Class Reference

Public Types

- enum **Media**

Public Member Functions

- **CaptureDevice** (ILogger logger, Media media, string deviceID)
- void **Initialize** ()
- void **InitializeAsync** ()
- async Task CleanUpAsync ()

  Asynchronous method cleaning up resources and stopping recording if necessary.
- async Task< IMediaEncodingProperties > SelectPreferredCameraStreamSettingAsync (MediaStreamType mediaStreamType, Func< IMediaEncodingProperties, bool > filterSettings)

  Allow selection of camera settings.
- async Task StartRecordingAsync (MediaEncodingProfile encodingProfile, Action< byte[ ], FrameFlags > encoderCallback)

  Starts media recording asynchronously
- async Task StopRecordingAsync ()

  Stops recording asynchronously

Static Public Member Functions

- static async Task< bool > **CheckForRecordingDeviceAsync** ()

Properties

- MediaCapture CaptureSource `[get]`

  Creates url object from MediaCapture

Events

- EventHandler< MediaCaptureFailedEventArgs > **CaptureFailed**

3.35.1 Member Function Documentation

3.35.1.1 CleanUpAsync()

```
async Task CleanUpAsync ( )
```

Asynchronous method cleaning up resources and stopping recording if necessary.

3.35.1.2 SelectPreferredCameraStreamSettingAsync()

```
async Task<IMediaEncodingProperties> SelectPreferredCameraStreamSettingAsync (
             MediaStreamType mediaStreamType,
             Func< IMediaEncodingProperties, bool > filterSettings )
```

Allow selection of camera settings.

Parameters

mediaStreamType	Type of a the media stream.
filterSettings	A predicate function, which will be called to filter the correct settings.

3.35.1.3 StartRecordingAsync()

```
async Task StartRecordingAsync (
             MediaEncodingProfile encodingProfile,
             Action< byte[], FrameFlags > encoderCallback )
```

Starts media recording asynchronously

Parameters

encodingProfile	Encoding profile used for the recording session

3.35.1.4 StopRecordingAsync()

```
async Task StopRecordingAsync ( )
```

Stops recording asynchronously

3.35.2 Property Documentation

3.35.2.1 CaptureSource

```
MediaCapture CaptureSource  [get]
```

Creates url object from MediaCapture

3.36 ConnectAndJoin Class Reference

Inherits MonoBehaviour, IConnectionCallbacks, and IMatchmakingCallbacks.

Public Member Functions

- void **ConnectNow** ()
- void **OnCreatedRoom** ()
- void **OnCreateRoomFailed** (short returnCode, string message)
- void **OnFriendListUpdate** (List< FriendInfo > friendList)
- void **OnJoinedRoom** ()
- void **OnJoinRandomFailed** (short returnCode, string message)
- void **OnJoinRoomFailed** (short returnCode, string message)
- void **OnLeftRoom** ()
- void **OnConnected** ()
- void **OnConnectedToMaster** ()
- void **OnDisconnected** (DisconnectCause cause)
- void **OnRegionListReceived** (RegionHandler regionHandler)
- void **OnCustomAuthenticationResponse** (Dictionary< string, object > data)
- void **OnCustomAuthenticationFailed** (string debugMessage)

Public Attributes

- bool **RandomRoom** = true
- string **RoomName**

Properties

- bool **IsConnected** `[get]`

3.37 VoiceClient.CreateOptions Struct Reference

Public Attributes

- byte **VoiceIDMin**
- byte **VoiceIDMax**

Static Public Attributes

- static CreateOptions **Default**

3.37.1 Member Data Documentation

3.37.1.1 Default

```
CreateOptions Default [static]
```

Initial value:

```
= new CreateOptions()
            {
                VoiceIDMin = 1,
                VoiceIDMax = 15
            }
```

3.38 OpusCodec.Decoder< T > Class Template Reference

Inherits IDecoder.

Public Member Functions

- **Decoder** (Action< FrameOut< T >> output, ILogger logger)
- void Open (VoiceInfo i)

  Open (initialize) the decoder.
- void **Dispose** ()
- void Input (ref FrameBuffer buf)

  Consumes the given encoded data.

Protected Attributes

- OpusDecoder< T > **decoder**

Properties

- string **Error** [get]

3.38.1 Member Function Documentation

3.38.1.1 Input()

```
void Input (
            ref FrameBuffer buf )
```

Consumes the given encoded data.

The callee can call buf.Retain() to prevent the caller from disposing the buffer. In this case, the callee should call buf.Release() when buffer is no longer needed.

Implements IDecoder.

3.38.1.2 Open()

```
void Open (
            VoiceInfo info )
```

Open (initialize) the decoder.

Parameters

info	Properties of the data stream to decode.

Implements IDecoder.

3.39 RawCodec.Decoder< T > Class Template Reference

Inherits IDecoder.

Public Member Functions

- **Decoder** (Action< FrameOut< T >> output)
- void Open (VoiceInfo info)

  Open (initialize) the decoder.
- void Input (ref FrameBuffer byteBuf)

  Consumes the given encoded data.
- void **Dispose** ()

Properties

- string **Error** `[get]`

3.39.1 Member Function Documentation

3.39.1.1 Input()

```
void Input (
            ref FrameBuffer buf )
```

Consumes the given encoded data.

The callee can call buf.Retain() to prevent the caller from disposing the buffer. In this case, the callee should call buf.Release() when buffer is no longer needed.

Implements IDecoder.

3.39.1.2 Open()

```
void Open (
            VoiceInfo info )
```

Open (initialize) the decoder.

Parameters

info	Properties of the data stream to decode.

Implements IDecoder.

3.40 OpusCodec.DecoderFactory Class Reference

Static Public Member Functions

- static IEncoder **Create**< **T** > (VoiceInfo i, ILogger logger)

3.41 DeviceEnumerator Class Reference

Inherits DeviceEnumeratorBase.

Inherited by AudioInEnumerator, and VideoInEnumerator.

Public Member Functions

- **DeviceEnumerator** (ILogger logger, Windows.Devices.Enumeration.DeviceClass deviceClass)
- override void **Refresh** ()
- override void **Dispose** ()

Additional Inherited Members

3.42 DeviceEnumeratorBase Class Reference

Inherits IDeviceEnumerator.

Inherited by DeviceEnumeratorNotSupported, AudioInEnumerator, AudioInEnumerator, AudioInEnumerator, VideoInEnumerator, DeviceEnumerator, and AudioInEnumerator.

Public Member Functions

- **DeviceEnumeratorBase** (ILogger logger)
- IEnumerator< DeviceInfo > **GetEnumerator** ()
- abstract void **Refresh** ()
- abstract void **Dispose** ()

Public Attributes

- virtual bool **IsSupported** => true

Protected Attributes

- List< DeviceInfo > **devices** = new List<DeviceInfo>()
- ILogger **logger**

Properties

- virtual string **Error** `[get, protected set]`

3.43 DeviceFeatures Class Reference

Public Member Functions

- **DeviceFeatures** (CameraFacing facing)

Properties

- CameraFacing **CameraFacing** `[get]`

3.44 DeviceInfo Struct Reference

Public Member Functions

- **DeviceInfo** (int id, string name, DeviceFeatures features=null)
- **DeviceInfo** (string id, string name, DeviceFeatures features=null)
- **DeviceInfo** (string name, DeviceFeatures features=null)
- override bool **Equals** (object obj)
- override int **GetHashCode** ()
- override string **ToString** ()

Static Public Member Functions

- static bool **operator==** (DeviceInfo d1, DeviceInfo d2)
- static bool **operator!=** (DeviceInfo d1, DeviceInfo d2)

Public Attributes

- DeviceFeatures **Features** => features == null ? DeviceFeatures.Default : features
- DeviceFeatures **features**

Static Public Attributes

- static readonly DeviceInfo **Default** = new DeviceInfo(true, -128, "", "[Default]")

Properties

- bool **IsDefault** `[get]`
- int **IDInt** `[get]`
- string **IDString** `[get]`
- string **Name** `[get]`

3.45 RecorderPreset.DSP Struct Reference

Public Attributes

- bool **AEC**
- bool **VAD**

3.46 OpusCodec.Encoder< T > Class Template Reference

Inherits IEncoderDirect< T[ ]>.

Public Member Functions

- void **Input** (T[ ] buf)
- void **EndOfStream** ()
- ArraySegment< byte > **DequeueOutput** (out FrameFlags flags)
- I **GetPlatformAPI< I >** ()
- void **Dispose** ()

Protected Member Functions

- **Encoder** (VoiceInfo i, ILogger logger)
- abstract ArraySegment< byte > **encodeTyped** (T[ ] buf)

Protected Attributes

- OpusEncoder **encoder**
- bool **disposed**

Properties

- string **Error** `[get]`
- Action< ArraySegment< byte >, FrameFlags > **Output** `[get, set]`

3.47 RawCodec.Encoder< T > Class Template Reference

Inherits IEncoderDirect< T[ ]>.

Public Member Functions

- ArraySegment< byte > **DequeueOutput** (out FrameFlags flags)
- void **EndOfStream** ()
- I **GetPlatformAPI< I >** ()
- void **Dispose** ()
- void **Input** (T[ ] buf)

Properties

- string **Error** `[get]`
- Action< ArraySegment< byte >, FrameFlags > **Output** `[get, set]`

3.48 OpusCodec.EncoderFloat Class Reference

Inherits OpusCodec.Encoder< float >.

Protected Member Functions

- override ArraySegment< byte > **encodeTyped** (float[ ] buf)

Additional Inherited Members

3.49 OpusCodec.EncoderShort Class Reference

Inherits OpusCodec.Encoder< short >.

Protected Member Functions

- override ArraySegment< byte > **encodeTyped** (short[ ] buf)

Additional Inherited Members

3.50 OpusCodec.Factory Class Reference

Static Public Member Functions

- static IEncoder **CreateEncoder< B >** (VoiceInfo i, ILogger logger)

3.51 FactoryPrimitiveArrayPool< T > Class Template Reference

PrimitiveArrayPool<T> as wrapped in object factory interface.

Inherits ObjectFactory< T[ ], int >.

Public Member Functions

- **FactoryPrimitiveArrayPool** (int capacity, string name)
- **FactoryPrimitiveArrayPool** (int capacity, string name, int info)
- T[ ] **New** ()
- T[ ] **New** (int size)
- void **Free** (T[ ] obj)
- void **Free** (T[ ] obj, int info)
- void **Dispose** ()

Properties

- int **Info** `[get]`

3.51.1 Detailed Description

PrimitiveArrayPool<T> as wrapped in object factory interface.

Template Parameters

T	Array element type.

3.52 FactoryReusableArray< T > Class Template Reference

Array factory returnig the same array instance as long as it requested with the same array length. If length changes, new array instance created.

Inherits ObjectFactory< T[ ], int >.

Public Member Functions

- **FactoryReusableArray** (int size)
- T[ ] **New** ()
- T[ ] **New** (int size)
- void **Free** (T[ ] obj)
- void **Free** (T[ ] obj, int info)
- void **Dispose** ()

Properties

- int **Info** `[get]`

3.52.1 Detailed Description

Array factory returnig the same array instance as long as it requested with the same array length. If length changes, new array instance created.

Template Parameters

T	Array element type.

3.53 Flip Struct Reference

Public Member Functions

- override bool **Equals** (object obj)
- override int **GetHashCode** ()

Static Public Member Functions

- static bool **operator==** (Flip f1, Flip f2)
- static bool **operator!=** (Flip f1, Flip f2)
- static Flip **operator*** (Flip f1, Flip f2)

Static Public Attributes

- static Flip **None**
- static Flip **Vertical** = new Flip() { IsVertical = true }
- static Flip **Horizontal** = new Flip() { IsHorizontal = true }
- static Flip **Both** = Vertical * Horizontal

Properties

- bool **IsVertical** `[get]`
- bool **IsHorizontal** `[get]`

3.54 FMODRecorderSetup Class Reference

Inherits VoiceComponent.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

3.55 FrameBuffer Struct Reference

Public Member Functions

- **FrameBuffer** (byte[ ] array, int offset, int count, FrameFlags flags, IDisposable disposer)
- **FrameBuffer** (byte[ ] array, FrameFlags flags)
- void **Retain** ()
- void **Release** ()

Public Attributes

- readonly byte[ ] **array**
- readonly int **offset**
- readonly int **count**
- readonly IDisposable **disposer**
- bool **disposed**
- int **refCnt**
- GCHandle **gcHandle**
- IntPtr **ptr**
- bool **pinned**

Properties

- IntPtr **Ptr** `[get]`
- byte[ ] **Array** `[get]`
- int **Length** `[get]`
- int **Offset** `[get]`
- FrameFlags **Flags** `[get]`

3.56 FrameOut< T > Class Template Reference

Public Member Functions

- **FrameOut** (T[ ] buf, bool endOfStream)
- FrameOut< T > **Set** (T[ ] buf, bool endOfStream)

Properties

- T[ ] **Buf** `[get]`
- bool **EndOfStream** `[get]`

3.57 Framer< T > Class Template Reference

Utility class to re-frame audio packets.

Public Member Functions

- Framer (int frameSize)

  Create new Framer instance.
- int Count (int bufLen)

  Get the number of frames available after adding bufLen samples.
- IEnumerable< T[ ]> Frame (T[ ] buf)

  Append arbitrary-sized buffer and return available full frames.

3.57.1 Detailed Description

Utility class to re-frame audio packets.

3.57.2 Constructor & Destructor Documentation

3.57.2.1 Framer()

```
Framer (
            int frameSize )
```

Create new Framer instance.

3.57.3 Member Function Documentation

3.57.3.1 Count()

```
int Count (
            int bufLen )
```

Get the number of frames available after adding bufLen samples.

Parameters

bufLen	Number of samples that would be added.

Returns

Number of full frames available when adding bufLen samples.

3.57.3.2 Frame()

```
IEnumerable<T[]> Frame (
            T[] buf )
```

Append arbitrary-sized buffer and return available full frames.

Parameters

buf	Array of samples to add.

Returns

Enumerator of full frames (might be none).

3.58 FusionVoiceClient Class Reference

Inherits VoiceConnection, and INetworkRunnerCallbacks.

Public Attributes

- bool UseFusionAppSettings = true

  Whether or not to use the Voice AppId and all the other AppSettings from Fusion's RealtimeAppSettings ScriptableObject singleton in the Voice client/app.
- bool UseFusionAuthValues = true

  Whether or not to use the same AuthenticationValues used in Fusion client/app in Voice client/app as well. This means that the same UserID will be used in both clients. If custom authentication is used and setup in Fusion AppId from dashboard, the same configuration should be done for the Voice AppId.

Protected Member Functions

- void **Start** ()
- override void **Awake** ()
- override void **OnDestroy** ()
- override Speaker **InstantiateSpeakerForRemoteVoice** (int playerId, byte voiceId, object userData)

Additional Inherited Members

3.58.1 Member Data Documentation

3.58.1.1 UseFusionAppSettings

```
bool UseFusionAppSettings = true
```

Whether or not to use the Voice AppId and all the other AppSettings from Fusion's RealtimeAppSettings ScriptableObject singleton in the Voice client/app.

3.58.1.2 UseFusionAuthValues

```
bool UseFusionAuthValues = true
```

Whether or not to use the same AuthenticationValues used in Fusion client/app in Voice client/app as well. This means that the same UserID will be used in both clients. If custom authentication is used and setup in Fusion AppId from dashboard, the same configuration should be done for the Voice AppId.

3.59 AudioUtil.GeneratorPusher< T > Class Template Reference

IAudioPusher that provides a constant tone signal.

Inherits IAudioPusher< T >.

Inherited by AudioUtil.ToneAudioPusher< T >, and AudioUtil.WaveformAudioPusher< T >.

Public Member Functions

- **GeneratorPusher** (int bufSizeMs=100, int samplingRate=48000, int channels=1)
- void SetCallback (Action< T[ ]> callback, ObjectFactory< T[ ], int > bufferFactory)

  Set the callback function used for pushing data
- void **Dispose** ()

Protected Member Functions

- abstract int **Gen** (T[ ] buf, long timeSamples)

Protected Attributes

- long **timeSamples**

Properties

- int **Channels** [get]
- int **SamplingRate** [get]
- string **Error** [get]

3.59.1 Detailed Description

IAudioPusher that provides a constant tone signal.

3.59.2 Member Function Documentation

3.59.2.1 SetCallback()

```
void SetCallback (
            Action< T[ ]> callback,
            ObjectFactory< T[ ], int > bufferFactory )
```

Set the callback function used for pushing data

Parameters

callback	Callback function to use
bufferFactory	Buffer factory used to create the buffer that is pushed to the callback

Implements IAudioPusher< T >.

3.60 AudioUtil.GeneratorReader< T > Class Template Reference

Inherits IAudioReader< T >.

Inherited by AudioUtil.ToneAudioReader< T >, and AudioUtil.WaveformAudioReader< T >.

Public Member Functions

- **GeneratorReader** (Func< double > clockSec=null, int samplingRate=48000, int channels=1)
- void **Dispose** ()
- bool Read (T[ ] buf)

  Fill full given frame buffer with source uncompressed data or return false if not enough such data.

Protected Member Functions

- abstract int **Gen** (T[ ] buf, long timeSamples)

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

3.60.1 Member Function Documentation

3.60.1.1 Read()

```
bool Read (
             T[ ] buffer )
```

Fill full given frame buffer with source uncompressed data or return false if not enough such data.

Parameters

buffer	Buffer to fill.

Returns

True if buffer was filled successfully, false otherwise.

Implements IDataReader< T >.

3.61 IAudioDesc Interface Reference

Audio Source interface.

Inherits IDisposable.

Inherited by AudioDesc, IAudioPusher< T >, and IAudioReader< T >.

Properties

- int SamplingRate `[get]`

  Sampling rate of the audio signal (in Hz).
- int Channels `[get]`

  Number of channels in the audio signal.
- string Error `[get]`

  If not null, audio object is in invalid state.

3.61.1 Detailed Description

Audio Source interface.

3.61.2 Property Documentation

3.61.2.1 Channels

```
int Channels [get]
```

Number of channels in the audio signal.

3.61.2.2 Error

```
string Error [get]
```

If not null, audio object is in invalid state.

3.61.2.3 SamplingRate

```
int SamplingRate  [get]
```

Sampling rate of the audio signal (in Hz).

3.62 IAudioInChangeNotifier Interface Reference

Inherits IDisposable.

Inherited by AudioInChangeNotifierNotSupported, AudioInChangeNotifier, and AudioInChangeNotifier.

Properties

- bool **IsSupported** `[get]`
- string **Error** `[get]`

3.63 IAudioOut< T > Interface Template Reference

Inherited by AudioOutDelayControl< T >, and AudioSyncBuffer< T >.

Public Member Functions

- void **Start** (int frequency, int channels, int frameSamplesPerChannel)
- void **Flush** ()
- void **Stop** ()
- void **Push** (T[ ] frame)
- void **Service** ()

Properties

- bool **IsPlaying** `[get]`
- int **Lag** `[get]`

3.64 IAudioPusher< T > Interface Template Reference

Audio Pusher interface.

Inherits IAudioDesc.

Inherited by AudioUtil.GeneratorPusher< T >.

Public Member Functions

- void SetCallback (Action< T[ ]> callback, ObjectFactory< T[ ], int > bufferFactory)

  Set the callback function used for pushing data.

Additional Inherited Members

3.64.1 Detailed Description

Audio Pusher interface.

Opposed to an IAudioReader (which will deliver audio data when it is "pulled"), an IAudioPusher will push its audio data whenever it is ready,

3.64.2 Member Function Documentation

3.64.2.1 SetCallback()

```
void SetCallback (
            Action< T[ ]> callback,
            ObjectFactory< T[ ], int > bufferFactory )
```

Set the callback function used for pushing data.

Parameters

callback	Callback function to use.
bufferFactory	Buffer factory used to create the buffer that is pushed to the callback

Implemented in AudioUtil.GeneratorPusher< T >.

3.65 IAudioReader< T > Interface Template Reference

Audio Reader interface.

Inherits IDataReader< T >, and IAudioDesc.

Inherited by AudioUtil.GeneratorReader< T >, and AudioInReader< T >.

Additional Inherited Members

3.65.1 Detailed Description

Audio Reader interface.

Opposed to an IAudioPusher (which will push its audio data whenever it is ready), an IAudioReader will deliver audio data when it is "pulled" (it's Read function is called).

3.66 IDataReader< T > Interface Template Reference

Interface for pulling data, in case this is more appropriate than pushing it.

Inherits IDisposable.

Inherited by IAudioReader< T >.

Public Member Functions

- bool Read (T[ ] buffer)

  Fill full given frame buffer with source uncompressed data or return false if not enough such data.

3.66.1 Detailed Description

Interface for pulling data, in case this is more appropriate than pushing it.

3.66.2 Member Function Documentation

3.66.2.1 Read()

```
bool Read (
          T[ ] buffer )
```

Fill full given frame buffer with source uncompressed data or return false if not enough such data.

Parameters

buffer	Buffer to fill.

Returns

True if buffer was filled successfully, false otherwise.

Implemented in AudioInReader< T >, and AudioUtil.GeneratorReader< T >.

3.67 IDecoder Interface Reference

Generic decoder interface.

Inherits IDisposable.

Inherited by IDecoderDirect< B >, OpusCodec.Decoder< T >, and RawCodec.Decoder< T >.

Public Member Functions

- void Open (VoiceInfo info)

  Open (initialize) the decoder.
- void Input (ref FrameBuffer buf)

  Consumes the given encoded data.

Properties

- string Error `[get]`

  If not null, the object is in invalid state.

3.67.1 Detailed Description

Generic decoder interface.

3.67.2 Member Function Documentation

3.67.2.1 Input()

```
void Input (
             ref FrameBuffer buf )
```

Consumes the given encoded data.

The callee can call buf.Retain() to prevent the caller from disposing the buffer. In this case, the callee should call buf.Release() when buffer is no longer needed.

Implemented in RawCodec.Decoder< T >, and OpusCodec.Decoder< T >.

3.67.2.2 Open()

```
void Open (
             VoiceInfo info )
```

Open (initialize) the decoder.

Parameters

info	Properties of the data stream to decode.

Implemented in RawCodec.Decoder< T >, and OpusCodec.Decoder< T >.

3.67.3 Property Documentation

3.67.3.1 Error

```
string Error  [get]
```

If not null, the object is in invalid state.

3.68 IDecoderDirect< B > Interface Template Reference

Interface for an decoder which outputs data via explicit call.

Inherits IDecoder.

Properties

- Action< B > Output `[get, set]`

  Callback to call when a new decoded data buffer is available.

Additional Inherited Members

3.68.1 Detailed Description

Interface for an decoder which outputs data via explicit call.

3.68.2 Property Documentation

3.68.2.1 Output

```
Action<B> Output  [get], [set]
```

Callback to call when a new decoded data buffer is available.

3.69 IDeviceEnumerator Interface Reference

Inherits IDisposable, and IEnumerable< DeviceInfo >.

Inherited by DeviceEnumeratorBase.

Public Member Functions

- void **Refresh** ()

Properties

- bool **IsSupported** `[get]`
- string **Error** `[get]`

3.70 IEncoder Interface Reference

Generic encoder interface.

Inherits IDisposable.

Inherited by IEncoderDirect< B >.

Public Member Functions

- ArraySegment< byte > DequeueOutput (out FrameFlags flags)

  Returns next encoded data frame (if such output supported).
- void EndOfStream ()

  Forces an encoder to flush and produce frame with EndOfStream flag (in output queue).
- I GetPlatformAPI< I > ()

  Returns an platform-specific interface.

Properties

- string Error `[get]`

  If not null, the object is in invalid state.
- Action< ArraySegment< byte >, FrameFlags > Output `[set]`

  Set callback encoder calls on each encoded data frame (if such output supported).

3.70.1 Detailed Description

Generic encoder interface.

Depending on implementation, encoder should either call Output on eaach data frame or return next data frame in DequeueOutput() call.

3.70.2 Member Function Documentation

3.70.2.1 DequeueOutput()

```
ArraySegment<byte> DequeueOutput (
             out FrameFlags flags )
```

Returns next encoded data frame (if such output supported).

3.70.2.2 EndOfStream()

```
void EndOfStream ( )
```

Forces an encoder to flush and produce frame with EndOfStream flag (in output queue).

3.70.2.3 GetPlatformAPI< I >()

```
I GetPlatformAPI< I > ( )
```

Returns an platform-specific interface.

Type Constraints

I* : *class

3.70.3 Property Documentation

3.70.3.1 Error

```
string Error  [get]
```

If not null, the object is in invalid state.

3.70.3.2 Output

```
Action<ArraySegment<byte>, FrameFlags> Output  [set]
```

Set callback encoder calls on each encoded data frame (if such output supported).

3.71 IEncoderDirect< B > Interface Template Reference

Interface for an encoder which consumes input data via explicit call.

Inherits IEncoder.

Public Member Functions

- void Input (B buf)

  Consumes the given raw data.

Additional Inherited Members

3.71.1 Detailed Description

Interface for an encoder which consumes input data via explicit call.

3.71.2 Member Function Documentation

3.71.2.1 Input()

```
void Input (
            B buf )
```

Consumes the given raw data.

Parameters

buf	Array containing raw data (e.g. audio samples).

3.72 IEncoderDirectImage Interface Reference

Interface for an encoder which consumes images via explicit call.

Inherits IEncoderDirect< ImageBufferNative >.

Properties

- ImageFormat ImageFormat `[get]`

  Recommended encoder input image format. Encoder may support other formats.

Additional Inherited Members

3.72.1 Detailed Description

Interface for an encoder which consumes images via explicit call.

3.72.2 Property Documentation

3.72.2.1 ImageFormat

```
ImageFormat ImageFormat  [get]
```

Recommended encoder input image format. Encoder may support other formats.

3.73 AudioUtil.ILevelMeter Interface Reference

Audio Level Metering interface.

Inherited by AudioUtil.LevelMeter< T >, and AudioUtil.LevelMeterDummy.

Public Member Functions

- void ResetAccumAvgPeakAmp ()

  Reset AccumAvgPeakAmp.

Properties

- float CurrentAvgAmp `[get]`

  Average amplitude value over last half second.
- float CurrentPeakAmp `[get]`

  Maximum amplitude value over last half second sec.
- float AccumAvgPeakAmp `[get]`

  Average of CurrentPeakAmps since last reset.

3.73.1 Detailed Description

Audio Level Metering interface.

3.73.2 Member Function Documentation

3.73.2.1 ResetAccumAvgPeakAmp()

```
void ResetAccumAvgPeakAmp ( )
```

Reset AccumAvgPeakAmp.

Implemented in AudioUtil.LevelMeter< T >, and AudioUtil.LevelMeterDummy.

3.73.3 Property Documentation

3.73.3.1 AccumAvgPeakAmp

```
float AccumAvgPeakAmp  [get]
```

Average of CurrentPeakAmps since last reset.

3.73.3.2 CurrentAvgAmp

```
float CurrentAvgAmp  [get]
```

Average amplitude value over last half second.

3.73.3.3 CurrentPeakAmp

```
float CurrentPeakAmp  [get]
```

Maximum amplitude value over last half second sec.

3.74 ILocalVoiceAudio Interface Reference

Interface for an outgoing audio stream.

Inherited by LocalVoiceAudio< T >, and LocalVoiceAudioDummy.

Public Member Functions

- void VoiceDetectorCalibrate (int durationMs, Action< float > onCalibrated=null)

  Trigger voice detector calibration process.

Properties

- AudioUtil.IVoiceDetector VoiceDetector `[get]`

  The VoiceDetector in use.
- AudioUtil.ILevelMeter LevelMeter `[get]`

  The LevelMeter utility in use.
- bool VoiceDetectorCalibrating `[get]`

  If true, voice detector calibration is in progress.

3.74.1 Detailed Description

Interface for an outgoing audio stream.

A LocalVoice always brings a LevelMeter and a VoiceDetector, which you can access using this interface.

3.74.2 Member Function Documentation

3.74.2.1 VoiceDetectorCalibrate()

```
void VoiceDetectorCalibrate (
            int durationMs,
            Action< float > onCalibrated = null )
```

Trigger voice detector calibration process.

While calibrating, keep silence. Voice detector sets threshold based on measured backgroud noise level.

Parameters

durationMs	Duration of calibration (in milliseconds).
onCalibrated	Called when calibration is complete. Parameter is new threshold value.

Implemented in LocalVoiceAudioDummy, and LocalVoiceAudio< T >.

3.74.3 Property Documentation

3.74.3.1 LevelMeter

```
AudioUtil.ILevelMeter LevelMeter  [get]
```

The LevelMeter utility in use.

3.74.3.2 VoiceDetector

```
AudioUtil.IVoiceDetector VoiceDetector  [get]
```

The VoiceDetector in use.

Use it to enable or disable voice detector and set its parameters.

3.74.3.3 VoiceDetectorCalibrating

```
bool VoiceDetectorCalibrating  [get]
```

If true, voice detector calibration is in progress.

3.75 ILogger Interface Reference

Inherited by LoadBalancingTransport, and Logger.

Public Member Functions

- void **LogError** (string fmt, params object[ ] args)
- void **LogWarning** (string fmt, params object[ ] args)
- void **LogInfo** (string fmt, params object[ ] args)
- void **LogDebug** (string fmt, params object[ ] args)

3.76 ImageBufferInfo Struct Reference

Classes

- struct StrideSet

Public Member Functions

- **ImageBufferInfo** (int width, int height, StrideSet stride, ImageFormat format)

Properties

- int **Width** `[get]`
- int **Height** `[get]`
- StrideSet **Stride** `[get]`
- ImageFormat **Format** `[get]`
- Rotation **Rotation** `[get, set]`
- Flip **Flip** `[get, set]`

3.77 ImageBufferNative Class Reference

Inherited by ImageBufferNativeAlloc, and ImageBufferNativeGCHandleSinglePlane.

Classes

- struct PlaneSet

Public Member Functions

- **ImageBufferNative** (ImageBufferInfo info)
- **ImageBufferNative** (IntPtr buf, int width, int height, int stride, ImageFormat imageFormat)
- virtual void **Release** ()
- virtual void **Dispose** ()

Public Attributes

- ImageBufferInfo **Info**
- PlaneSet **Planes**

3.78 ImageBufferNativeAlloc Class Reference

Inherits ImageBufferNative, and IDisposable.

Public Member Functions

- **ImageBufferNativeAlloc** (ImageBufferNativePool< ImageBufferNativeAlloc > pool, ImageBufferInfo info)
- override void **Release** ()
- override void **Dispose** ()

Additional Inherited Members

3.79 ImageBufferNativeGCHandleSinglePlane Class Reference

Inherits ImageBufferNative, and IDisposable.

Public Member Functions

- **ImageBufferNativeGCHandleSinglePlane** (ImageBufferNativePool< ImageBufferNativeGCHandleSinglePlane > pool, ImageBufferInfo info)
- void **PinPlane** (byte[ ] plane)
- override void **Release** ()
- override void **Dispose** ()

Additional Inherited Members

3.80 ImageBufferNativePool< T > Class Template Reference

Inherits ObjectPool< T, ImageBufferInfo >.

Public Member Functions

- delegate T **Factory** (ImageBufferNativePool< T > pool, ImageBufferInfo info)
- **ImageBufferNativePool** (int capacity, Factory factory, string name)
- **ImageBufferNativePool** (int capacity, Factory factory, string name, ImageBufferInfo info)

Protected Member Functions

- override T **createObject** (ImageBufferInfo info)
- override void **destroyObject** (T obj)
- override bool **infosMatch** (ImageBufferInfo i0, ImageBufferInfo i1)

Additional Inherited Members

3.81 IProcessor< T > Interface Template Reference

Audio Processor interface.

Inherits IDisposable.

Inherited by AudioUtil.LevelMeter< T >, AudioUtil.Resampler< T >, AudioUtil.VoiceDetector< T >, AudioUtil.VoiceDetectorCalibrati and AudioUtil.VoiceLevelDetectCalibrate< T >.

Public Member Functions

- T[ ] Process (T[ ] buf)

  Process a frame of audio data.

3.81.1 Detailed Description

Audio Processor interface.

3.81.2 Member Function Documentation

3.81.2.1 Process()

```
T [ ] Process (
            T[ ] buf )
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implemented in AudioUtil.VoiceLevelDetectCalibrate< T >, AudioUtil.VoiceDetector< T >, AudioUtil.VoiceDetectorCalibration< T >, AudioUtil.LevelMeter< T >, and AudioUtil.Resampler< T >.

3.82 IResettable Interface Reference

Inherited by AudioInPusher, AudioInReader, and AndroidAudioInAEC.

Public Member Functions

- void **Reset** ()

3.83 IServiceable Interface Reference

Interface for classes that want their Service() function to be called regularly in the context of a LocalVoice.

Inherited by BufferReaderPushAdapterBase< T >.

Public Member Functions

- void Service (LocalVoice localVoice)

  Service function that should be called regularly.

3.83.1 Detailed Description

Interface for classes that want their Service() function to be called regularly in the context of a LocalVoice.

3.83.2 Member Function Documentation

3.83.2.1 Service()

```
void Service (
            LocalVoice localVoice )
```

Service function that should be called regularly.

Implemented in BufferReaderPushAdapterAsyncPoolCopy< T >, BufferReaderPushAdapterAsyncPool< T >, BufferReaderPushAdapter< T >, and BufferReaderPushAdapterBase< T >.

3.84 AudioUtil.IVoiceDetector Interface Reference

Voice Activity Detector interface.

Inherited by AudioUtil.VoiceDetector< T >, and AudioUtil.VoiceDetectorDummy.

Properties

- bool On `[get, set]`

  If true, voice detection enabled.
- float Threshold `[get, set]`

  Voice detected as soon as signal level exceeds threshold.
- bool Detected `[get]`

  If true, voice detected.
- DateTime DetectedTime `[get]`

  Last time when switched to detected state.
- int ActivityDelayMs `[get, set]`

  Keep detected state during this time after signal level dropped below threshold.

Events

- Action OnDetected

  Called when switched to detected state.

3.84.1 Detailed Description

Voice Activity Detector interface.

3.84.2 Property Documentation

3.84.2.1 ActivityDelayMs

```
int ActivityDelayMs [get], [set]
```

Keep detected state during this time after signal level dropped below threshold.

3.84.2.2 Detected

```
bool Detected [get]
```

If true, voice detected.

3.84.2.3 DetectedTime

```
DateTime DetectedTime [get]
```

Last time when switched to detected state.

3.84.2.4 On

```
bool On [get], [set]
```

If true, voice detection enabled.

3.84.2.5 Threshold

```
float Threshold [get], [set]
```

Voice detected as soon as signal level exceeds threshold.

3.84.3 Event Documentation

3.84.3.1 OnDetected

```
Action OnDetected
```

Called when switched to detected state.

3.85 IVoiceTransport Interface Reference

Inherited by LoadBalancingTransport.

Public Member Functions

- bool **IsChannelJoined** (int channelId)
- void **SendVoicesInfo** (IEnumerable< LocalVoice > voices, int channelId, int targetPlayerId)
- void **SendVoiceRemove** (LocalVoice voice, int channelId, int targetPlayerId)
- void **SendFrame** (ArraySegment< byte > data, FrameFlags flags, byte evNumber, byte voiceId, int channelId, int targetPlayerId, bool reliable, LocalVoice localVoice)
- string **ChannelIdStr** (int channelId)
- string **PlayerIdStr** (int playerId)

3.86 AudioUtil.LevelMeter< T > Class Template Reference

Audio Level Meter.

Inherits IProcessor< T >, and AudioUtil.ILevelMeter.

Public Member Functions

- void ResetAccumAvgPeakAmp ()

  Reset AccumAvgPeakAmp.
- abstract T[ ] Process (T[ ] buf)

  Process a frame of audio data.
- void **Dispose** ()

Protected Attributes

- float **ampSum**
- float **ampPeak**
- int **bufferSize**
- float[ ] **prevValues**
- int **prevValuesHead**
- float **accumAvgPeakAmpSum**
- int **accumAvgPeakAmpCount**
- float **currentPeakAmp**
- float **norm**

Properties

- float **CurrentAvgAmp** `[get]`
- float **CurrentPeakAmp** `[get, protected set]`
- float? **AccumAvgPeakAmp** `[get]`

3.86.1 Detailed Description

Audio Level Meter.

3.86.2 Member Function Documentation

3.86.2.1 Process()

```
abstract T [] Process (
            T[] buf )  [pure virtual]
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implements IProcessor< T >.

3.86.2.2 ResetAccumAvgPeakAmp()

```
void ResetAccumAvgPeakAmp ( )
```

Reset AccumAvgPeakAmp.

Implements AudioUtil.ILevelMeter.

3.87 AudioUtil.LevelMeterDummy Class Reference

Dummy Audio Level Meter that doesn't actually do anything.

Inherits AudioUtil.ILevelMeter.

Public Member Functions

- void ResetAccumAvgPeakAmp ()

  Reset AccumAvgPeakAmp.

Properties

- float **CurrentAvgAmp** `[get]`
- float **CurrentPeakAmp** `[get]`
- float **AccumAvgPeakAmp** `[get]`

3.87.1 Detailed Description

Dummy Audio Level Meter that doesn't actually do anything.

3.87.2 Member Function Documentation

3.87.2.1 ResetAccumAvgPeakAmp()

```
void ResetAccumAvgPeakAmp ( )
```

Reset AccumAvgPeakAmp.

Implements AudioUtil.ILevelMeter.

3.88 AudioUtil.LevelMeterFloat Class Reference

LevelMeter specialization for float audio.

Inherits AudioUtil.LevelMeter< float >.

Public Member Functions

- LevelMeterFloat (int samplingRate, int numChannels)

  Create new LevelMeterFloat instance.
- override float[ ] **Process** (float[ ] buf)

Additional Inherited Members

3.88.1 Detailed Description

LevelMeter specialization for float audio.

3.88.2 Constructor & Destructor Documentation

3.88.2.1 LevelMeterFloat()

```
LevelMeterFloat (
              int samplingRate,
              int numChannels )
```

Create new LevelMeterFloat instance.

Parameters

samplingRate	Sampling rate of the audio signal (in Hz).
numChannels	Number of channels in the audio signal.

3.89 AudioUtil.LevelMeterShort Class Reference

LevelMeter specialization for short audio.

Inherits AudioUtil.LevelMeter< short >.

Public Member Functions

- LevelMeterShort (int samplingRate, int numChannels)

  Create new LevelMeterShort instance.
- override short[ ] **Process** (short[ ] buf)

Additional Inherited Members

3.89.1 Detailed Description

LevelMeter specialization for short audio.

3.89.2 Constructor & Destructor Documentation

3.89.2.1 LevelMeterShort()

```
LevelMeterShort (
              int samplingRate,
              int numChannels )
```

Create new LevelMeterShort instance.

Parameters

samplingRate	Sampling rate of the audio signal (in Hz).
numChannels	Number of channels in the audio signal.

3.90 LoadBalancingTransport Class Reference

Extends LoadBalancingClient with media streaming functionality.

Inherits LoadBalancingClient, IVoiceTransport, ILogger, and IDisposable.

Inherited by LoadBalancingTransport2.

Public Member Functions

- void **LogError** (string fmt, params object[ ] args)
- void **LogWarning** (string fmt, params object[ ] args)
- void **LogInfo** (string fmt, params object[ ] args)
- void **LogDebug** (string fmt, params object[ ] args)
- bool **IsChannelJoined** (int channelId)
- LoadBalancingTransport (ILogger logger=null, ConnectionProtocol connectionProtocol=ConnectionProtocol.Udp)

  Initializes a new LoadBalancingTransport.
- new void Service ()

  This method dispatches all available incoming commands and then sends this client's outgoing commands. Call this method regularly (2 to 20 times a second).
- virtual bool **ChangeAudioGroups** (byte[ ] groupsToRemove, byte[ ] groupsToAdd)
- void **SendVoicesInfo** (IEnumerable< LocalVoice > voices, int channelId, int targetPlayerId)
- void **SendVoiceRemove** (LocalVoice voice, int channelId, int targetPlayerId)
- virtual void **SendFrame** (ArraySegment< byte > data, FrameFlags flags, byte evNumber, byte voiceId, int channelId, int targetPlayerId, bool reliable, LocalVoice localVoice)
- string **ChannelIdStr** (int channelId)
- string **PlayerIdStr** (int playerId)
- void Dispose ()

  Releases all resources used by the LoadBalancingTransport instance.

Protected Member Functions

- virtual void **onEventActionVoiceClient** (EventData ev)

Protected Attributes

- VoiceClient **voiceClient**

Properties

- VoiceClient VoiceClient `[get]`

  The VoiceClient implementation associated with this LoadBalancingTransport.
- byte **GlobalAudioGroup** `[get, set]`
- byte GlobalInterestGroup `[get, set]`

  Set global interest group for this client. This call sets InterestGroup for existing local voices and for created later to given value. Client set as listening to this group only until LoadBalancingPeer.OpChangeGroups() called. This method can be called any time.

3.90.1 Detailed Description

Extends LoadBalancingClient with media streaming functionality.

Use your normal LoadBalancing workflow to join a Voice room. All standard LoadBalancing features are available. Use VoiceClient to work with media streams.

3.90.2 Constructor & Destructor Documentation

3.90.2.1 LoadBalancingTransport()

```
LoadBalancingTransport (
            ILogger logger = null,
            ConnectionProtocol connectionProtocol = ConnectionProtocol.Udp )
```

Initializes a new LoadBalancingTransport.

Parameters

logger	ILogger instance. If null, this instance LoadBalancingClient.DebugReturn implementation is used.ConnectionProtocol
connectionProtocol	Connection protocol (UDP or TCP). ConnectionProtocol

3.90.3 Member Function Documentation

3.90.3.1 Dispose()

```
void Dispose ( )
```

Releases all resources used by the LoadBalancingTransport instance.

3.90.3.2 Service()

```
new void Service ( )
```

This method dispatches all available incoming commands and then sends this client's outgoing commands. Call this method regularly (2 to 20 times a second).

3.90.4 Property Documentation

3.90.4.1 GlobalInterestGroup

```
byte GlobalInterestGroup  [get], [set]
```

Set global interest group for this client. This call sets InterestGroup for existing local voices and for created later to given value. Client set as listening to this group only until LoadBalancingPeer.OpChangeGroups() called. This method can be called any time.

LocalVoice.InterestGroup LoadBalancingPeer.OpChangeGroups(byte[ ], byte[ ])

3.90.4.2 VoiceClient

```
VoiceClient VoiceClient  [get]
```

The VoiceClient implementation associated with this LoadBalancingTransport.

3.91 LoadBalancingTransport2 Class Reference

Variant of LoadBalancingTransport. Aims to be non-alloc at the cost of breaking compatibility with older clients.

Inherits LoadBalancingTransport.

Public Member Functions

- **LoadBalancingTransport2** (ILogger logger=null, ConnectionProtocol connectionProtocol=Connection←Protocol.Udp)
- override void **SendFrame** (ArraySegment< byte > data, FrameFlags flags, byte evNumber, byte voiceId, int channelId, int targetPlayerId, bool reliable, LocalVoice localVoice)

Protected Member Functions

- override void **onEventActionVoiceClient** (EventData ev)

Additional Inherited Members

3.91.1 Detailed Description

Variant of LoadBalancingTransport. Aims to be non-alloc at the cost of breaking compatibility with older clients.

3.92 LocalVoice Class Reference

Represents outgoing data stream.

Inherits IDisposable.

Inherited by LocalVoiceAudioDummy, and LocalVoiceFramedBase.

Public Member Functions

- void **SendSpacingProfileStart** ()
- void RemoveSelf ()

  Remove this voice from it's VoiceClient (using VoiceClient.RemoveLocalVoice
- virtual void **Dispose** ()

Static Public Attributes

- const int **DATA_POOL_CAPACITY** = 50

Protected Attributes

- VoiceInfo **info**
- IEncoder **encoder**
- VoiceClient **voiceClient**
- ArraySegment< byte > **configFrame**
- volatile bool **disposed**
- object **disposeLock** = new object()

Properties

- byte **Group** `[get, set]`
- byte InterestGroup `[get, set]`

  If InterestGroup != 0, voice's data is sent only to clients listening to this group (if supported by transport).
- VoiceInfo Info `[get]`

  Returns Info structure assigned on local voice cration.
- bool TransmitEnabled `[get, set]`

  If true, stream data broadcasted.
- bool IsCurrentlyTransmitting `[get]`

  Returns true if stream broadcasts.
- int FramesSent `[get]`

  Sent frames counter.
- int FramesSentBytes `[get]`

  Sent frames bytes counter.
- bool Reliable `[get, set]`

  Send data reliable.
- bool Encrypt `[get, set]`

  Send data encrypted.
- IServiceable LocalUserServiceable `[get, set]`

  Optional user object attached to LocalVoice. its Service() will be called at each VoiceClient.Service() call.

- bool DebugEchoMode `[get, set]`

  If true, outgoing stream routed back to client via server same way as for remote client's streams. Can be swithed any time. OnRemoteVoiceInfoAction and OnRemoteVoiceRemoveAction are triggered if required. This functionality availability depends on transport.
- string **SendSpacingProfileDump** `[get]`
- int SendSpacingProfileMax `[get]`

  Logs input frames time spacing profiling results. Do not call frequently.
- byte **ID** `[get]`
- byte **EvNumber** `[get]`
- string **shortName** `[get]`
- string **Name** `[get]`
- string **LogPrefix** `[get]`

3.92.1 Detailed Description

Represents outgoing data stream.

3.92.2 Member Function Documentation

3.92.2.1 RemoveSelf()

```
void RemoveSelf ( )
```

Remove this voice from it's VoiceClient (using VoiceClient.RemoveLocalVoice

3.92.3 Property Documentation

3.92.3.1 DebugEchoMode

```
bool DebugEchoMode  [get], [set]
```

If true, outgoing stream routed back to client via server same way as for remote client's streams. Can be swithed any time. OnRemoteVoiceInfoAction and OnRemoteVoiceRemoveAction are triggered if required. This functionality availability depends on transport.

3.92.3.2 Encrypt

```
bool Encrypt  [get], [set]
```

Send data encrypted.

3.92.3.3 FramesSent

```
int FramesSent  [get]
```

Sent frames counter.

3.92.3.4 FramesSentBytes

```
int FramesSentBytes  [get]
```

Sent frames bytes counter.

3.92.3.5 Info

```
VoiceInfo Info  [get]
```

Returns Info structure assigned on local voice cration.

3.92.3.6 InterestGroup

```
byte InterestGroup  [get], [set]
```

If InterestGroup != 0, voice's data is sent only to clients listening to this group (if supported by transport).

3.92.3.7 IsCurrentlyTransmitting

```
bool IsCurrentlyTransmitting  [get]
```

Returns true if stream broadcasts.

3.92.3.8 LocalUserServiceable

```
IServiceable LocalUserServiceable  [get], [set]
```

Optional user object attached to LocalVoice. its Service() will be called at each VoiceClient.Service() call.

3.92.3.9 Reliable

```
bool Reliable  [get], [set]
```

Send data reliable.

3.92.3.10 SendSpacingProfileMax

```
int SendSpacingProfileMax  [get]
```

Logs input frames time spacing profiling results. Do not call frequently.

3.92.3.11 TransmitEnabled

```
bool TransmitEnabled  [get], [set]
```

If true, stream data broadcasted.

3.93 LocalVoiceAudio< T > Class Template Reference

Outgoing audio stream.

Inherits LocalVoiceFramed< T >, and ILocalVoiceAudio.

Public Member Functions

- void VoiceDetectorCalibrate (int durationMs, Action< float > onCalibrated=null)

  Trigger voice detector calibration process.

Static Public Member Functions

- static LocalVoiceAudio< T > Create (VoiceClient voiceClient, byte voiceId, IEncoder encoder, VoiceInfo voiceInfo, IAudioDesc audioSourceDesc, int channelId)

  Create a new LocalVoiceAudio{T} instance.

Protected Member Functions

- void **initBuiltinProcessors** ()

Protected Attributes

- AudioUtil.VoiceDetector< T > **voiceDetector**
- AudioUtil.VoiceDetectorCalibration< T > **voiceDetectorCalibration**
- AudioUtil.LevelMeter< T > **levelMeter**
- int **channels**
- bool **resampleSource**

Properties

- virtual AudioUtil.IVoiceDetector **VoiceDetector** [get]
- virtual AudioUtil.ILevelMeter **LevelMeter** [get]
- bool VoiceDetectorCalibrating [get]

  True if the VoiceDetector is currently calibrating.

Additional Inherited Members

3.93.1 Detailed Description

Outgoing audio stream.

3.93.2 Member Function Documentation

3.93.2.1 Create()

```
static LocalVoiceAudio<T> Create (
            VoiceClient voiceClient,
            byte voiceId,
            IEncoder encoder,
            VoiceInfo voiceInfo,
            IAudioDesc audioSourceDesc,
            int channelId )  [static]
```

Create a new LocalVoiceAudio{T} instance.

Parameters

voiceClient	The VoiceClient to use for this outgoing stream.
voiceId	Numeric ID for this voice.
encoder	Encoder to use for this voice.
voiceInfo	Outgoing stream parameters.
audioSourceDesc	Audio source parameters.
channelId	Voice transport channel ID to use for this voice.

Returns

The new LocalVoiceAudio{T} instance.

3.93.2.2 VoiceDetectorCalibrate()

```
void VoiceDetectorCalibrate (
             int durationMs,
             Action< float > onCalibrated = null )
```

Trigger voice detector calibration process.

While calibrating, keep silence. Voice detector sets threshold basing on measured backgroud noise level.

Parameters

durationMs	Duration of calibration in milliseconds.
onCalibrated	Called when calibration is complete. Parameter is new threshold value.

Implements ILocalVoiceAudio.

3.93.3 Property Documentation

3.93.3.1 VoiceDetectorCalibrating

```
bool VoiceDetectorCalibrating  [get]
```

True if the VoiceDetector is currently calibrating.

3.94 LocalVoiceAudioDummy Class Reference

Dummy LocalVoiceAudio

Inherits LocalVoice, and ILocalVoiceAudio.

Public Member Functions

- void VoiceDetectorCalibrate (int durationMs, Action< float > onCalibrated=null)

  Trigger voice detector calibration process.

Static Public Attributes

- static LocalVoiceAudioDummy Dummy = new LocalVoiceAudioDummy()

  A Dummy LocalVoiceAudio instance.

Properties

- AudioUtil.IVoiceDetector **VoiceDetector** `[get]`
- AudioUtil.ILevelMeter **LevelMeter** `[get]`
- bool **VoiceDetectorCalibrating** `[get]`

Additional Inherited Members

3.94.1 Detailed Description

Dummy LocalVoiceAudio

For testing, this LocalVoiceAudio implementation features a AudioUtil.VoiceDetectorDummy and a AudioUtil.LevelMeterDummy

3.94.2 Member Function Documentation

3.94.2.1 VoiceDetectorCalibrate()

```
void VoiceDetectorCalibrate (
             int durationMs,
             Action< float > onCalibrated = null )
```

Trigger voice detector calibration process.

While calibrating, keep silence. Voice detector sets threshold based on measured backgroud noise level.

Parameters

durationMs	Duration of calibration (in milliseconds).
onCalibrated	Called when calibration is complete. Parameter is new threshold value.

Implements ILocalVoiceAudio.

3.94.3 Member Data Documentation

3.94.3.1 Dummy

```
LocalVoiceAudioDummy Dummy = new LocalVoiceAudioDummy()  [static]
```

A Dummy LocalVoiceAudio instance.

3.95 LocalVoiceAudioFloat Class Reference

Specialization of LocalVoiceAudio< T > for float audio

Inherits LocalVoiceAudio< float >.

Additional Inherited Members

3.95.1 Detailed Description

Specialization of LocalVoiceAudio< T > for float audio

3.96 LocalVoiceAudioShort Class Reference

Specialization of LocalVoiceAudio< T > for short audio

Inherits LocalVoiceAudio< short >.

Additional Inherited Members

3.96.1 Detailed Description

Specialization of LocalVoiceAudio< T > for short audio

3.97 LocalVoiceFramed< T > Class Template Reference

Typed re-framing LocalVoice

Inherits LocalVoiceFramedBase.

Inherited by LocalVoiceAudio< T >.

Public Member Functions

- void AddPostProcessor (params IProcessor< T >[ ] processors)

  Adds processors after any built-in processors and everything added with AddPreProcessor.
- void AddPreProcessor (params IProcessor< T >[ ] processors)

  Adds processors before built-in processors and everything added with AddPostProcessor.
- void RemoveProcessor (params IProcessor< T >[ ] processors)

  Adds processors before built-in processors and everything added with AddPostProcessor.
- void ClearProcessors ()

  Clears all processors in pipeline including built-in resampling. User should add at least resampler processor after call.
- void PushDataAsync (T[ ] buf)

  Asynchronously push data into this stream.
- void PushData (T[ ] buf)

  Synchronously push data into this stream.
- override void Dispose ()

  Releases resources used by the LocalVoiceFramed<T> instance. Buffers used for asynchronous push will be disposed in encoder thread's 'finally'.

Protected Member Functions

- T[ ] **processFrame** (T[ ] buf)

Properties

- FactoryPrimitiveArrayPool< T > **BufferFactory** `[get]`
- bool PushDataAsyncReady `[get]`

  Wether this LocalVoiceFramed has capacity for more data buffers to be pushed asynchronously.

Additional Inherited Members

3.97.1 Detailed Description

Typed re-framing LocalVoice

Consumes data in array buffers of arbitrary length. Repacks them in frames of constant length for further processing and encoding.

3.97.2 Member Function Documentation

3.97.2.1 AddPostProcessor()

```
void AddPostProcessor (
            params IProcessor< T >[ ] processors )
```

Adds processors after any built-in processors and everything added with AddPreProcessor.

Parameters

processors	

3.97.2.2 AddPreProcessor()

```
void AddPreProcessor (
            params IProcessor< T >[] processors )
```

Adds processors before built-in processors and everything added with AddPostProcessor.

Parameters

processors	

3.97.2.3 ClearProcessors()

```
void ClearProcessors ( )
```

Clears all processors in pipeline including built-in resampling. User should add at least resampler processor after call.

3.97.2.4 Dispose()

```
override void Dispose ( )  [virtual]
```

Releases resources used by the LocalVoiceFramed<T> instance. Buffers used for asynchronous push will be disposed in encoder thread's 'finally'.

Reimplemented from LocalVoice.

3.97.2.5 PushData()

```
void PushData (
            T[] buf )
```

Synchronously push data into this stream.

3.97.2.6 PushDataAsync()

```
void PushDataAsync (
             T[] buf )
```

Asynchronously push data into this stream.

3.97.2.7 RemoveProcessor()

```
void RemoveProcessor (
             params IProcessor< T >[] processors )
```

Adds processors before built-in processors and everything added with AddPostProcessor.

Parameters

processors	

3.97.3 Property Documentation

3.97.3.1 PushDataAsyncReady

```
bool PushDataAsyncReady  [get]
```

Wether this LocalVoiceFramed has capacity for more data buffers to be pushed asynchronously.

3.98 LocalVoiceFramedBase Class Reference

Typed re-framing LocalVoice

Inherits LocalVoice.

Inherited by LocalVoiceFramed< T >.

Properties

- int FrameSize `[get]`

  Data flow will be repacked to frames of this size. May differ from input voiceInfo.FrameSize. Processors should resample in this case.

Additional Inherited Members

3.98.1 Detailed Description

Typed re-framing LocalVoice

Base class for typed re-framing LocalVoice implementation (LocalVoiceFramed<T>)

3.98.2 Property Documentation

3.98.2.1 FrameSize

```
int FrameSize  [get]
```

Data flow will be repacked to frames of this size. May differ from input voiceInfo.FrameSize. Processors should resample in this case.

3.99 Logger Class Reference

Inherits ILogger.

Public Member Functions

- void **LogError** (string fmt, params object[ ] args)
- void **LogWarning** (string fmt, params object[ ] args)
- void **LogInfo** (string fmt, params object[ ] args)
- void **LogDebug** (string fmt, params object[ ] args)

3.100 MicAmplifier Class Reference

Inherits VoiceComponent.

Properties

- float **AmplificationFactor** `[get, set]`

Additional Inherited Members

3.101 MicAmplifierFloat Class Reference

Inherits IProcessor< float >.

Public Member Functions

- **MicAmplifierFloat** (float amplificationFactor)
- float[ ] **Process** (float[ ] buf)
- void **Dispose** ()

Properties

- float **AmplificationFactor** `[get, set]`
- bool **Disabled** `[get, set]`

3.102 MicAmplifierShort Class Reference

Inherits IProcessor< short >.

Public Member Functions

- **MicAmplifierShort** (float amplificationFactor)
- short[ ] **Process** (short[ ] buf)
- void **Dispose** ()

Properties

- float **AmplificationFactor** `[get, set]`
- bool **Disabled** `[get, set]`

3.103 MicrophonePermission Class Reference

Helper to request Microphone permission on Android or iOS.

Inherits VoiceComponent.

Public Member Functions

- void **InitVoice** ()

Protected Member Functions

- override void **Awake** ()

Properties

- bool? **HasPermission** `[get]`

Events

- static Action< bool > **MicrophonePermissionCallback**

Additional Inherited Members

3.103.1 Detailed Description

Helper to request Microphone permission on Android or iOS.

3.104 MicWrapper Class Reference

Inherits IAudioReader< float >.

Public Member Functions

- **MicWrapper** (string device, int suggestedFrequency, ILogger logger)
- void **Dispose** ()
- bool **Read** (float[ ] buffer)

Properties

- int? **SamplingRate** `[get]`
- int? **Channels** `[get]`
- string **Error** `[get]`

3.105 MicWrapperPusher Class Reference

Inherits IAudioPusher< float >.

Public Member Functions

- **MicWrapperPusher** (GameObject parent, string device, int suggestedFrequency, ILogger logger)
- void **SetCallback** (Action< float[ ]> callback, ObjectFactory< float[ ], int > bufferFactory)
- void **Dispose** ()

Properties

- int? **SamplingRate** `[get]`
- int? **Channels** `[get]`
- string **Error** `[get]`

3.106 MicWrapperPusherOnAudioFilterRead Class Reference

Inherits MonoBehaviour.

Events

- Action< float[ ], int > **OnAudioFrame**

3.107 AudioInChangeNotifier.MonoPInvokeCallbackAttribute Class Reference

Inherits Attribute.

Public Member Functions

- **MonoPInvokeCallbackAttribute** (Type t)

3.108 MonoPInvokeCallbackAttribute Class Reference

Inherits Attribute.

Public Member Functions

- **MonoPInvokeCallbackAttribute** (Type t)

3.109 MonoPInvokeCallbackAttribute Class Reference

Inherits Attribute.

Public Member Functions

- **MonoPInvokeCallbackAttribute** (Type t)

3.110 MonoPInvokeCallbackAttribute Class Reference

Inherits Attribute.

Public Member Functions

- **MonoPInvokeCallbackAttribute** (Type t)

3.111 ObjectFactory< TType, TInfo > Interface Template Reference

Uniform interface to ObjectPool<TType, TInfo> and single reusable object.

Inherits IDisposable.

Public Member Functions

- TType **New** ()
- TType **New** (TInfo info)
- void **Free** (TType obj)
- void **Free** (TType obj, TInfo info)

Properties

- TInfo **Info** `[get]`

3.111.1 Detailed Description

Uniform interface to ObjectPool<TType, TInfo> and single reusable object.

Template Parameters

TType	Object type.
TInfo	Type of property used to check 2 objects identity (like integral length of array).

3.112 ObjectPool< TType, TInfo > Class Template Reference

Generic Pool to re-use objects of a certain type (TType) that optionally match a certain property or set of properties (TInfo).

Inherits IDisposable.

Public Member Functions

- ObjectPool (int capacity, string name)

  Create a new ObjectPool instance. Does not call Init().
- ObjectPool (int capacity, string name, TInfo info)

  Create a new ObjectPool instance with the given info structure. Calls Init().
- void Init (TInfo info)

  (Re-)Initializes this ObjectPool.
- TType AcquireOrCreate ()

  Acquire an existing object, or create a new one if none are available.
- TType AcquireOrCreate (TInfo info)

  Acquire an existing object (if info matches), or create a new one from the passed info.

- virtual bool Release (TType obj, TInfo objInfo)

  Returns object to pool.
- virtual bool Release (TType obj)

  Returns object to pool, or destroys it if the pool is full.
- void Dispose ()

  Free resources assoicated with this ObjectPool

Protected Member Functions

- abstract TType **createObject** (TInfo info)
- abstract void **destroyObject** (TType obj)
- abstract bool **infosMatch** (TInfo i0, TInfo i1)

Protected Attributes

- int **capacity**
- TInfo **info**
- int **pos**
- string **name**

Properties

- TInfo Info `[get]`

  The property (info) that objects in this Pool must match.

3.112.1 Detailed Description

Generic Pool to re-use objects of a certain type (TType) that optionally match a certain property or set of properties (TInfo).

Template Parameters

TType	Object type.
TInfo	Type of parameter used to check 2 objects identity (like integral length of array).

3.112.2 Constructor & Destructor Documentation

3.112.2.1 ObjectPool() `[1/2]`

```
ObjectPool (
              int capacity,
              string name )
```

Create a new ObjectPool instance. Does not call Init().

Parameters

capacity	Capacity (size) of the object pool.
name	Name of the object pool.

3.112.2.2 ObjectPool() `[2/2]`

```
ObjectPool (
             int capacity,
             string name,
             TInfo info )
```

Create a new ObjectPool instance with the given info structure. Calls Init().

Parameters

capacity	Capacity (size) of the object pool.
name	Name of the object pool.
info	Info about this Pool's objects.

3.112.3 Member Function Documentation

3.112.3.1 AcquireOrCreate() `[1/2]`

```
TType AcquireOrCreate ( )
```

Acquire an existing object, or create a new one if none are available.

If it fails to get one from the pool, this will create from the info given in this pool's constructor.

3.112.3.2 AcquireOrCreate() `[2/2]`

```
TType AcquireOrCreate (
             TInfo info )
```

Acquire an existing object (if info matches), or create a new one from the passed info.

Parameters

info	Info structure to match, or create a new object with.

3.112.3.3 Dispose()

```
void Dispose ( )
```

Free resources assoicated with this ObjectPool

3.112.3.4 Init()

```
void Init (
            TInfo info )
```

(Re-)Initializes this ObjectPool.

If there are objects available in this Pool, they will be destroyed. Allocates (Capacity) new Objects.

Parameters

info	Info about this Pool's objects.

3.112.3.5 Release() [1/2]

```
virtual bool Release (
            TType obj )  [virtual]
```

Returns object to pool, or destroys it if the pool is full.

Parameters

obj	The object to return to the pool.

3.112.3.6 Release() [2/2]

```
virtual bool Release (
            TType obj,
            TInfo objInfo )  [virtual]
```

Returns object to pool.

Parameters

obj	The object to return to the pool.
objInfo	The info structure about obj.

obj is returned to the pool only if objInfo matches this pool's info. Else, it is destroyed.

3.112.4 Property Documentation

3.112.4.1 Info

```
TInfo Info  [get]
```

The property (info) that objects in this Pool must match.

3.113 OpusCodec Class Reference

Classes

- class Decoder
- class DecoderFactory
- class Encoder
- class EncoderFloat
- class EncoderShort
- class Factory
- class Util

Public Types

- enum **FrameDuration**

Properties

- static string **Version** `[get]`

3.114 PhotonAppSettings Class Reference

Collection of connection-relevant settings, used internally by PhotonNetwork.ConnectUsingSettings.

Inherits ScriptableObject.

Public Member Functions

- void UseCloud (string cloudAppid, string code="")

  Sets appid and region code in the AppSettings. Used in Editor.
- override string ToString ()

  String summary of the AppSettings.

Static Public Member Functions

- static void **LoadOrCreateSettings** ()

Public Attributes

- AppSettings **AppSettings**

Properties

- static PhotonAppSettings **Instance** `[get]`

3.114.1 Detailed Description

Collection of connection-relevant settings, used internally by PhotonNetwork.ConnectUsingSettings.

Includes the AppSettings class from the Realtime APIs plus some other, PUN-relevant, settings.

3.114.2 Member Function Documentation

3.114.2.1 ToString()

```
override string ToString ( )
```

String summary of the AppSettings.

3.114.2.2 UseCloud()

```
void UseCloud (
             string cloudAppid,
             string code = "" )
```

Sets appid and region code in the AppSettings. Used in Editor.

3.115 PhotonVoiceCreatedParams Class Reference

Properties

- Voice.LocalVoice **Voice** `[get, set]`
- Voice.IAudioDesc **AudioDesc** `[get, set]`

3.116 PhotonVoiceLagSimulationGui Class Reference

Inherits MonoBehaviour.

Public Member Functions

- void **OnEnable** ()

3.117 PhotonVoiceStatsGui Class Reference

Basic GUI to show traffic and health statistics of the connection to Photon, toggled by shift+tab.

Inherits MonoBehaviour.

3.117.1 Detailed Description

Basic GUI to show traffic and health statistics of the connection to Photon, toggled by shift+tab.

The shown health values can help identify problems with connection losses or performance. Example: If the time delta between two consecutive SendOutgoingCommands calls is a second or more, chances rise for a disconnect being caused by this (because acknowledgments to the server need to be sent in due time).

3.118 PhotonVoiceView Class Reference

Component that should be attached to a networked PUN prefab that has PhotonView. It will bind remote Recorder with local Speaker of the same networked prefab. This component makes automatic voice stream routing easy for players' characters/avatars.

Inherits VoiceComponent.

Protected Member Functions

- override void **Awake** ()

Properties

- Recorder RecorderInUse `[get]`

  The Recorder component currently used by this PhotonVoiceView
- Speaker SpeakerInUse `[get]`

  The Speaker component currently used by this PhotonVoiceView
- bool IsSpeaking `[get]`

  If true, this PhotonVoiceView has a Speaker that is currently playing received audio frames from remote audio source
- bool IsRecording `[get]`

  If true, this PhotonVoiceView has a Recorder that is currently transmitting audio stream from local audio source

Additional Inherited Members

3.118.1 Detailed Description

Component that should be attached to a networked PUN prefab that has PhotonView. It will bind remote Recorder with local Speaker of the same networked prefab. This component makes automatic voice stream routing easy for players' characters/avatars.

3.118.2 Property Documentation

3.118.2.1 IsRecording

```
bool IsRecording  [get]
```

If true, this PhotonVoiceView has a Recorder that is currently transmitting audio stream from local audio source

3.118.2.2 IsSpeaking

```
bool IsSpeaking  [get]
```

If true, this PhotonVoiceView has a Speaker that is currently playing received audio frames from remote audio source

3.118.2.3 RecorderInUse

```
Recorder RecorderInUse  [get]
```

The Recorder component currently used by this PhotonVoiceView

3.118.2.4 SpeakerInUse

```
Speaker SpeakerInUse  [get]
```

The Speaker component currently used by this PhotonVoiceView

3.119 ImageBufferNative.PlaneSet Struct Reference

Public Member Functions

- **PlaneSet** (int length, IntPtr p0=default(IntPtr), IntPtr p1=default(IntPtr), IntPtr p2=default(IntPtr), IntPtr p3=default(IntPtr))

Properties

- IntPtr **this[int key]** `[get, set]`
- int **Length** `[get]`

3.120 Platform Class Reference

Static Public Member Functions

- static IDeviceEnumerator **CreateAudioInEnumerator** (ILogger logger)
- static IAudioInChangeNotifier **CreateAudioInChangeNotifier** (Action callback, ILogger logger)
- static IEncoder **CreateDefaultAudioEncoder**< **T** > (ILogger logger, VoiceInfo info)
- static IAudioDesc **CreateDefaultAudioSource** (ILogger logger, DeviceInfo dev, int samplingRate, int channels, object otherParams=null)
- static IDeviceEnumerator **CreateVideoInEnumerator** (ILogger logger)
- static IEncoderDirectImage **CreateDefaultVideoEncoder** (ILogger logger, VoiceInfo info)
- static IDecoderDirect< ImageBufferNative > **CreateDefaultVideoDecoder** (ILogger logger, VoiceInfo info)
- static IVideoRecorder **CreateDefaultVideoRecorder** (ILogger logger, VoiceInfo info, DeviceInfo camDevice, Action< IVideoRecorder > onReady)
- static IVideoPlayer **CreateDefaultVideoPlayer** (ILogger logger, VoiceInfo info, Action< IVideoPlayer > on↩ Ready)
- static IPreviewManager **CreateDefaultPreviewManager** (ILogger logger)
- static IVideoRecorder **CreateVideoRecorderUnityTexture** (ILogger logger, VoiceInfo info, DeviceInfo cam↩ Device, Action< IVideoRecorder > onReady)
- static IVideoPlayer **CreateVideoPlayerUnityTexture** (ILogger logger, VoiceInfo info, Action< IVideoPlayer > onReady)
- static IPreviewManager **CreatePreviewManagerUnityTexture** (ILogger logger)

3.121 AudioOutDelayControl.PlayDelayConfig Struct Reference

Public Attributes

- int **Low**
- int **High**
- int **Max**
- int **SpeedUpPerc**

Static Public Attributes

- static PlayDelayConfig **Default**

3.121.1 Member Data Documentation

3.121.1.1 Default

```
PlayDelayConfig Default  [static]
```

Initial value:

```
= new PlayDelayConfig()
            {
                Low = 200,
                High = 400,
                Max = 1000,
                SpeedUpPerc = 5,
            }
```

3.122 PrimitiveArrayPool< T > Class Template Reference

Pool of Arrays with components of type T, with ObjectPool info being the array's size.

Inherits ObjectPool< T[ ], int >.

Public Member Functions

- **PrimitiveArrayPool** (int capacity, string name)
- **PrimitiveArrayPool** (int capacity, string name, int info)

Protected Member Functions

- override T[ ] **createObject** (int info)
- override void **destroyObject** (T[ ] obj)
- override bool **infosMatch** (int i0, int i1)

Additional Inherited Members

3.122.1 Detailed Description

Pool of Arrays with components of type T, with ObjectPool info being the array's size.

Template Parameters

T	Array element type.

3.123 PunVoiceClient Class Reference

This class can be used to automatically sync client states between PUN and Voice. It also finds the Speaker component for a character's voice. For this to work attach a PhotonVoiceView next to the PhotonView of your player's prefab.

Inherits VoiceConnection.

Public Member Functions

- bool ConnectAndJoinRoom ()

  Connect voice client to Photon servers and join a Voice room
- void Disconnect ()

  Disconnect voice client from all Photon servers

Public Attributes

- bool AutoConnectAndJoin = true

  Auto connect voice client and join a voice room when PUN client is joined to a PUN room
- bool AutoLeaveAndDisconnect = true

  Auto disconnect voice client when PUN client is not joined to a PUN room

Static Public Attributes

- const string VoiceRoomNameSuffix = "_voice_"

  Suffix for voice room names appended to PUN room names.

Protected Member Functions

- void **Start** ()
- override void **OnDisable** ()
- override void **OnDestroy** ()
- override void **OnVoiceStateChanged** (ClientState fromState, ClientState toState)
- override Speaker **InstantiateSpeakerForRemoteVoice** (int playerId, byte voiceId, object userData)

Properties

- static PunVoiceClient Instance `[get]`

  Singleton instance for PunVoiceClient
- bool UsePunAppSettings `[get, set]`

  Whether or not to use the Voice AppId and all the other AppSettings from PUN's PhotonServerSettings ScriptableObject singleton in the Voice client/app.
- bool UsePunAuthValues `[get, set]`

  Whether or not to use the same PhotonNetwork.AuthValues in PunVoiceClient.Instance.Client.AuthValues. This means that the same UserID will be used in both clients. If custom authentication is used and setup in PUN app, the same configuration should be done for the Voice app.

Additional Inherited Members

3.123.1 Detailed Description

This class can be used to automatically sync client states between PUN and Voice. It also finds the Speaker component for a character's voice. For this to work attach a PhotonVoiceView next to the PhotonView of your player's prefab.

3.123.2 Member Function Documentation

3.123.2.1 ConnectAndJoinRoom()

```
bool ConnectAndJoinRoom ( )
```

Connect voice client to Photon servers and join a Voice room

Returns

If true, connection command send from client

3.123.2.2 Disconnect()

```
void Disconnect ( )
```

Disconnect voice client from all Photon servers

3.123.3 Member Data Documentation

3.123.3.1 AutoConnectAndJoin

```
bool AutoConnectAndJoin = true
```

Auto connect voice client and join a voice room when PUN client is joined to a PUN room

3.123.3.2 AutoLeaveAndDisconnect

```
bool AutoLeaveAndDisconnect = true
```

Auto disconnect voice client when PUN client is not joined to a PUN room

3.123.3.3 VoiceRoomNameSuffix

```
const string VoiceRoomNameSuffix = "_voice_"  [static]
```

Suffix for voice room names appended to PUN room names.

3.123.4 Property Documentation

3.123.4.1 Instance

```
PunVoiceClient Instance  [static], [get]
```

Singleton instance for PunVoiceClient

3.123.4.2 UsePunAppSettings

```
bool UsePunAppSettings  [get], [set]
```

Whether or not to use the Voice AppId and all the other AppSettings from PUN's PhotonServerSettings Scriptable←Object singleton in the Voice client/app.

3.123.4.3 UsePunAuthValues

```
bool UsePunAuthValues  [get], [set]
```

Whether or not to use the same PhotonNetwork.AuthValues in PunVoiceClient.Instance.Client.AuthValues. This means that the same UserID will be used in both clients. If custom authentication is used and setup in PUN app, the same configuration should be done for the Voice app.

3.124 RawCodec Class Reference

Classes

- class Decoder
- class Encoder
- class ShortToFloat

3.125 Recorder Class Reference

Component representing outgoing audio stream in scene.

Inherits VoiceComponent.

Public Types

- enum **InputSourceType**
- enum **MicType**

Public Member Functions

- bool RestartRecording ()

  Restarts recording if Recorder.IsRecoring is true
- void VoiceDetectorCalibrate (int durationMs, Action< float > detectionEndedCallback=null)

  Trigger voice detector calibration process. While calibrating, keep silence. Voice detector sets threshold basing on measured background noise level.
- bool SetIosAudioSessionParameters (IOS.AudioSessionParameters asp)

  Sets the AudioSessionParameters for iOS audio initialization when Photon MicrophoneType is used.
- bool SetIosAudioSessionParameters (IOS.AudioSessionCategory category, IOS.AudioSessionMode mode, IOS.AudioSessionCategoryOption[ ] options)

  Sets the AudioSessionParameters for iOS audio initialization when Photon MicrophoneType is used.
- bool SetAndroidNativeMicrophoneSettings (bool aec=false, bool agc=false, bool ns=false)

  Sets the native Android audio input settings when the Photon microphone type is used.
- bool ResetLocalAudio ()

  Resets audio session and parameters locally to fix broken recording due to system configuration modifications or audio interruptions or audio routing changes.

Static Public Attributes

- const int **MIN_OPUS_BITRATE** = 6000
- const int **MAX_OPUS_BITRATE** = 510000

Protected Member Functions

- virtual void **SendPhotonVoiceCreatedMessage** ()
- void **Update** ()

Properties

- bool TransmitEnabled `[get, set]`

  If true, audio transmission is enabled.
- bool Encrypt `[get, set]`

  If true, voice stream is sent encrypted.
- bool DebugEchoMode `[get, set]`

  If true, outgoing stream routed back to client via server same way as for remote client's streams.
- bool ReliableMode `[get, set]`

  If true, stream data sent in reliable mode.
- bool VoiceDetection `[get, set]`

  If true, voice detection enabled.
- float VoiceDetectionThreshold `[get, set]`

  Voice detection threshold (0..1, where 1 is full amplitude).
- int VoiceDetectionDelayMs `[get, set]`

  Keep detected state during this time after signal level dropped below threshold. Default is 500ms
- object UserData `[get, set]`

  Custom user object to be sent in the voice stream info event.
- Func< IAudioDesc > InputFactory `[get, set]`

  Set the method returning new Voice.IAudioDesc instance to be assigned to a new voice created with Source set to Factory
- AudioUtil.IVoiceDetector? VoiceDetector `[get]`

  Returns voice activity detector for recorder's audio stream.
- byte InterestGroup `[get, set]`

  Target interest group that will receive transmitted audio.
- bool IsCurrentlyTransmitting `[get]`

  Returns true if audio stream broadcasts.
- AudioUtil.ILevelMeter? LevelMeter `[get]`

  Level meter utility.
- bool VoiceDetectorCalibrating `[get]`

  If true, voice detector calibration is in progress.
- ILocalVoiceAudio **voiceAudio** `[get]`
- InputSourceType SourceType `[get, set]`

  Audio data source.
- MicType MicrophoneType `[get, set]`

  Which microphone API to use when the Source is set to Microphone.
- AudioClip AudioClip `[get, set]`

  Source audio clip.
- bool LoopAudioClip `[get, set]`

  Loop playback for audio clip sources.
- SamplingRate SamplingRate `[get, set]`

  Outgoing audio stream sampling rate.
- OpusCodec.FrameDuration FrameDuration `[get, set]`

  Outgoing audio stream encoder delay.
- int Bitrate `[get, set]`

  Outgoing audio stream bitrate.
- bool RecordingEnabled `[get, set]`

  Gets or sets whether this Recorder is recording audio to be transmitted.
- bool StopRecordingWhenPaused `[get, set]`

  If true, stop recording when paused resume/restart when un-paused.
- bool UseOnAudioFilterRead `[get, set]`

If true, recording will make use of Unity's OnAudioFitlerRead callback from a muted local AudioSource.

- bool UseMicrophoneTypeFallback `[get, set]`

  If true, if recording fails to start with Unity microphone type, Photon microphone type is used -if available- as a fallback and vice versa.

- bool RecordWhenJoined `[get, set]`

  If true, recording starts when joining the room and stops when leaving the room.

- DeviceInfo **MicrophoneDevice** `[get, set]`

Additional Inherited Members

3.125.1 Detailed Description

Component representing outgoing audio stream in scene.

3.125.2 Member Function Documentation

3.125.2.1 ResetLocalAudio()

```
bool ResetLocalAudio ( )
```

Resets audio session and parameters locally to fix broken recording due to system configuration modifications or audio interruptions or audio routing changes.

Returns

If reset is done.

3.125.2.2 RestartRecording()

```
bool RestartRecording ( )
```

Restarts recording if Recorder.IsRecoring is true

3.125.2.3 SetAndroidNativeMicrophoneSettings()

```
bool SetAndroidNativeMicrophoneSettings (
            bool aec = false,
            bool agc = false,
            bool ns = false )
```

Sets the native Android audio input settings when the Photon microphone type is used.

Parameters

aec	Acoustic Echo Cancellation
agc	Automatic Gain Control
ns	Noise Suppression

Returns

If a change has been made.

3.125.2.4 SetIosAudioSessionParameters() [1/2]

```
bool SetIosAudioSessionParameters (
            IOS.AudioSessionCategory category,
            IOS.AudioSessionMode mode,
            IOS.AudioSessionCategoryOption[] options )
```

Sets the AudioSessionParameters for iOS audio initialization when Photon MicrophoneType is used.

Parameters

category	Audio session category to be used.
mode	Audio session mode to be used.
options	Audio session category options to be used

Returns

If a change has been made.

3.125.2.5 SetIosAudioSessionParameters() [2/2]

```
bool SetIosAudioSessionParameters (
            IOS.AudioSessionParameters asp )
```

Sets the AudioSessionParameters for iOS audio initialization when Photon MicrophoneType is used.

Parameters

asp	You can use custom value or one from presets, IOS.AudioSessionParametersPresets

Returns

If a change has been made.

3.125.2.6 VoiceDetectorCalibrate()

```
void VoiceDetectorCalibrate (
             int durationMs,
             Action< float > detectionEndedCallback = null )
```

Trigger voice detector calibration process. While calibrating, keep silence. Voice detector sets threshold basing on measured background noise level.

Parameters

durationMs	Duration of calibration in milliseconds.
detectionEndedCallback	Callback when VAD calibration ends.

3.125.3 Property Documentation

3.125.3.1 AudioClip

```
AudioClip AudioClip [get], [set]
```

Source audio clip.

3.125.3.2 Bitrate

```
int Bitrate [get], [set]
```

Outgoing audio stream bitrate.

3.125.3.3 DebugEchoMode

```
bool DebugEchoMode [get], [set]
```

If true, outgoing stream routed back to client via server same way as for remote client's streams.

3.125.3.4 Encrypt

```
bool Encrypt [get], [set]
```

If true, voice stream is sent encrypted.

3.125.3.5 FrameDuration

```
OpusCodec.FrameDuration FrameDuration  [get], [set]
```

Outgoing audio stream encoder delay.

3.125.3.6 InputFactory

```
Func<IAudioDesc> InputFactory  [get], [set]
```

Set the method returning new Voice.IAudioDesc instance to be assigned to a new voice created with Source set to Factory

3.125.3.7 InterestGroup

```
byte InterestGroup  [get], [set]
```

Target interest group that will receive transmitted audio.

If InterestGroup != 0, recorder's audio data is sent only to clients listening to this group.

3.125.3.8 IsCurrentlyTransmitting

```
bool IsCurrentlyTransmitting  [get]
```

Returns true if audio stream broadcasts.

3.125.3.9 LevelMeter

```
AudioUtil.ILevelMeter?  LevelMeter  [get]
```

Level meter utility.

3.125.3.10 LoopAudioClip

```
bool LoopAudioClip  [get], [set]
```

Loop playback for audio clip sources.

3.125.3.11 MicrophoneType

```
MicType MicrophoneType  [get], [set]
```

Which microphone API to use when the Source is set to Microphone.

3.125.3.12 RecordingEnabled

```
bool RecordingEnabled  [get], [set]
```

Gets or sets whether this Recorder is recording audio to be transmitted.

3.125.3.13 RecordWhenJoined

```
bool RecordWhenJoined  [get], [set]
```

If true, recording starts when joining the room and stops when leaving the room.

3.125.3.14 ReliableMode

```
bool ReliableMode  [get], [set]
```

If true, stream data sent in reliable mode.

3.125.3.15 SamplingRate

```
SamplingRate SamplingRate  [get], [set]
```

Outgoing audio stream sampling rate.

3.125.3.16 SourceType

```
InputSourceType SourceType  [get], [set]
```

Audio data source.

3.125.3.17 StopRecordingWhenPaused

```
bool StopRecordingWhenPaused [get], [set]
```

If true, stop recording when paused resume/restart when un-paused.

3.125.3.18 TransmitEnabled

```
bool TransmitEnabled [get], [set]
```

If true, audio transmission is enabled.

3.125.3.19 UseMicrophoneTypeFallback

```
bool UseMicrophoneTypeFallback [get], [set]
```

If true, if recording fails to start with Unity microphone type, Photon microphone type is used -if available- as a fallback and vice versa.

3.125.3.20 UseOnAudioFilterRead

```
bool UseOnAudioFilterRead [get], [set]
```

If true, recording will make use of Unity's OnAudioFitlerRead callback from a muted local AudioSource.

If enabled, 3D sounds and voice positioning can be lost.

3.125.3.21 UserData

```
object UserData [get], [set]
```

Custom user object to be sent in the voice stream info event.

3.125.3.22 VoiceDetection

```
bool VoiceDetection [get], [set]
```

If true, voice detection enabled.

3.125.3.23 VoiceDetectionDelayMs

```
int VoiceDetectionDelayMs  [get], [set]
```

Keep detected state during this time after signal level dropped below threshold. Default is 500ms

3.125.3.24 VoiceDetectionThreshold

```
float VoiceDetectionThreshold  [get], [set]
```

Voice detection threshold (0..1, where 1 is full amplitude).

3.125.3.25 VoiceDetector

```
AudioUtil.IVoiceDetector?  VoiceDetector  [get]
```

Returns voice activity detector for recorder's audio stream.

3.125.3.26 VoiceDetectorCalibrating

```
bool VoiceDetectorCalibrating  [get]
```

If true, voice detector calibration is in progress.

3.126 RecorderPreset Class Reference

Inherits VoiceComponent.

Classes

- struct DSP

Public Attributes

- RuntimePlatform **Platform**
- MicType **MicrophoneType**
- bool **DSPEnabled**
- DSP **DSPSettings**

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

3.127 RemoteVoiceInfo Class Reference

Information about a remote voice (incoming stream).

Properties

- VoiceInfo Info `[get]`

  Remote voice info.
- int ChannelId `[get]`

  ID of channel used for transmission.
- int PlayerId `[get]`

  Player ID of voice owner.
- byte VoiceId `[get]`

  Voice ID (unique in the room).

3.127.1 Detailed Description

Information about a remote voice (incoming stream).

3.127.2 Property Documentation

3.127.2.1 ChannelId

```
int ChannelId  [get]
```

ID of channel used for transmission.

3.127.2.2 Info

```
VoiceInfo Info  [get]
```

Remote voice info.

3.127.2.3 PlayerId

```
int PlayerId  [get]
```

Player ID of voice owner.

3.127.2.4 VoiceId

```
byte VoiceId  [get]
```

Voice ID (unique in the room).

3.128 RemoteVoiceLink Class Reference

Public Member Functions

- **RemoteVoiceLink** (VoiceInfo info, int playerId, byte voiceId, int channelId, ref RemoteVoiceOptions options)
- override string **ToString** ()

Public Attributes

- readonly VoiceInfo **VoiceInfo**
- readonly int **PlayerId**
- readonly byte **VoiceId**
- readonly int **ChannelId**

Events

- Action< FrameOut< float > > **FloatFrameDecoded**
- Action **RemoteVoiceRemoved**

3.129 RemoteVoiceOptions Struct Reference

Event Actions and other options for a remote voice (incoming stream).

Public Member Functions

- **RemoteVoiceOptions** (ILogger logger, string logPrefix, VoiceInfo voiceInfo)
- void SetOutput (Action< FrameOut< float >> output)

  Create default audio decoder and register a method to be called when a data frame is decoded.
- void SetOutput (Action< FrameOut< short >> output)

  Create default audio decoder and register a method to be called when a data frame is decoded.

Properties

- Action OnRemoteVoiceRemoveAction `[get, set]`

  Register a method to be called when the remote voice is removed.
- IDecoder Decoder `[get, set]`

  Remote voice data decoder. Use to set decoder options or override it with user decoder.

3.129.1 Detailed Description

Event Actions and other options for a remote voice (incoming stream).

3.129.2 Member Function Documentation

3.129.2.1 SetOutput() [1/2]

```
void SetOutput (
            Action< FrameOut< float >> output )
```

Create default audio decoder and register a method to be called when a data frame is decoded.

3.129.2.2 SetOutput() [2/2]

```
void SetOutput (
            Action< FrameOut< short >> output )
```

Create default audio decoder and register a method to be called when a data frame is decoded.

3.129.3 Property Documentation

3.129.3.1 Decoder

```
IDecoder Decoder  [get], [set]
```

Remote voice data decoder. Use to set decoder options or override it with user decoder.

3.129.3.2 OnRemoteVoiceRemoveAction

```
Action OnRemoteVoiceRemoveAction  [get], [set]
```

Register a method to be called when the remote voice is removed.

3.130 AudioUtil.Resampler< T > Class Template Reference

Sample-rate conversion Audio Processor.

Inherits IProcessor< T >.

Public Member Functions

- Resampler (int dstSize, int channels)

  Create a new Resampler instance.
- T[ ] Process (T[ ] buf)

  Process a frame of audio data.
- void **Dispose** ()

Protected Attributes

- T[ ] **frameResampled**

3.130.1 Detailed Description

Sample-rate conversion Audio Processor.

This processor converts the sample-rate of the source stream. Internally, it uses AudioUtil.Resample<T>(T[ ], T[ ], int, int).

3.130.2 Constructor & Destructor Documentation

3.130.2.1 Resampler()

```
Resampler (
              int dstSize,
              int channels )
```

Create a new Resampler instance.

Parameters

dstSize	Frame size of a destination frame. Determins output rate.
channels	Number of audio channels expected in both in- and output.

3.130.3 Member Function Documentation

3.130.3.1 Process()

```
T [] Process (
            T[] buf )
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implements IProcessor< T >.

3.131 SaveIncomingStreamToFile Class Reference

Inherits VoiceComponent.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

3.132 SaveOutgoingStreamToFile Class Reference

Inherits VoiceComponent.

Additional Inherited Members

3.133 RawCodec.ShortToFloat Class Reference

Public Member Functions

- **ShortToFloat** (Action< FrameOut< float >> output)
- void **Output** (FrameOut< short > shortBuf)

3.134 Speaker Class Reference

Inherits VoiceComponent.

Inherited by SpeakerFMOD, and SpeakerAudioFilterRead.

Public Member Functions

- void RestartPlayback ()

  Restarts the audio playback of the linked incoming remote audio stream via AudioSource component.

Protected Member Functions

- override void **Awake** ()
- virtual IAudioOut< float > **CreateAudioOut** ()
- void **Update** ()

Protected Attributes

- IAudioOut< float > **audioOutput**
- AudioOutDelayControl.PlayDelayConfig **playDelayConfig** = AudioOutDelayControl.PlayDelayConfig.Default
- bool **restartOnDeviceChange** = true

Properties

- bool IsPlaying `[get]`

  Is the speaker playing right now.
- int? Lag `[get]`

  Smoothed difference between (jittering) stream and (clock-driven) audioOutput.
- Action< Speaker > OnRemoteVoiceRemoveAction `[get, set]`

  Register a method to be called when remote voice removed.
- RemoteVoiceLink **RemoteVoice** `[get]`
- bool IsLinked `[get]`

  Whether or not this Speaker has been linked to a remote voice stream.
- AudioOutDelayControl.PlayDelayConfig PlayDelayConfig `[get, set]`

  Gets or sets jitter buffer config.
- int PlayDelay `[get, set]`

  Gets or sets jitter buffer size in ms.
- bool **RestartOnDeviceChange** `[get, set]`

Additional Inherited Members

3.134.1 Member Function Documentation

3.134.1.1 RestartPlayback()

```
void RestartPlayback ( )
```

Restarts the audio playback of the linked incoming remote audio stream via AudioSource component.

Returns

True if playback is successfully restarted.

3.134.2 Property Documentation

3.134.2.1 IsLinked

```
bool IsLinked [get]
```

Whether or not this Speaker has been linked to a remote voice stream.

3.134.2.2 IsPlaying

```
bool IsPlaying [get]
```

Is the speaker playing right now.

3.134.2.3 Lag

```
int? Lag [get]
```

Smoothed difference between (jittering) stream and (clock-driven) audioOutput.

3.134.2.4 OnRemoteVoiceRemoveAction

```
Action<Speaker> OnRemoteVoiceRemoveAction [get], [set]
```

Register a method to be called when remote voice removed.

3.134.2.5 PlayDelay

```
int PlayDelay  [get], [set]
```

Gets or sets jitter buffer size in ms.

The method updates PlayDelayConfig with reasonable values based on the single value provided. Use PlayDelayConfig for more precise control.

3.134.2.6 PlayDelayConfig

```
AudioOutDelayControl.PlayDelayConfig PlayDelayConfig  [get], [set]
```

Gets or sets jitter buffer config.

Make sure that the new value is fully initialized or built from AudioOutDelayControl.PlayDelayConfig.Default.

3.135 SpeakerAudioFilterRead Class Reference

Inherits Speaker.

Protected Member Functions

- override IAudioOut< float > **CreateAudioOut** ()

Additional Inherited Members

3.136 SpeakerFMOD Class Reference

Inherits Speaker.

Protected Member Functions

- override IAudioOut< float > **CreateAudioOut** ()

Additional Inherited Members

3.137 ImageBufferInfo.StrideSet Struct Reference

Public Member Functions

- **StrideSet** (int length, int s0=0, int s1=0, int s2=0, int s3=0)

Properties

- int **this[int key]** `[get, set]`
- int **Length** `[get]`

3.138 AudioUtil.TempoUp< T > Class Template Reference

Public Member Functions

- void **Begin** (int channels, int changePerc, int skipGroup)
- int **Process** (T[ ] s, T[ ] d)
- int **End** (T[ ] s)
- int **endFloat** (float[ ] s)
- int **endShort** (short[ ] s)

3.139 TestTone Class Reference

Inherits MonoBehaviour.

3.140 AudioUtil.ToneAudioPusher< T > Class Template Reference

IAudioPusher that provides a constant tone signal.

Inherits AudioUtil.GeneratorPusher< T >.

Public Member Functions

- ToneAudioPusher (int frequency=440, int bufSizeMs=100, int samplingRate=48000, int channels=1)

  Create a new ToneAudioReader instance

Protected Member Functions

- override int **Gen** (T[ ] buf, long timeSamples)

Additional Inherited Members

3.140.1 Detailed Description

IAudioPusher that provides a constant tone signal.

3.140.2 Constructor & Destructor Documentation

3.140.2.1 ToneAudioPusher()

```
ToneAudioPusher (
              int frequency = 440,
              int bufSizeMs = 100,
              int samplingRate = 48000,
              int channels = 1 )
```

Create a new ToneAudioReader instance

Parameters

frequency	Frequency of the generated tone (in Hz).
bufSizeMs	Size of buffers to push (in milliseconds).
samplingRate	Sampling rate of the audio signal (in Hz).
channels	Number of channels in the audio signal.

3.141 AudioUtil.ToneAudioReader< T > Class Template Reference

IAudioReader that provides a constant tone signal.

Inherits AudioUtil.GeneratorReader< T >.

Public Member Functions

- ToneAudioReader (Func< double > clockSec=null, double frequency=440, int samplingRate=48000, int channels=1)

  Create a new ToneAudioReader instance

Protected Member Functions

- override int **Gen** (T[ ] buf, long timeSamples)

Additional Inherited Members

3.141.1 Detailed Description

IAudioReader that provides a constant tone signal.

Because of current resampling algorithm, the tone is distorted if SamplingRate does not equal encoder sampling rate.

3.141.2 Constructor & Destructor Documentation

3.141.2.1 ToneAudioReader()

```
ToneAudioReader (
            Func< double > clockSec = null,
            double frequency = 440,
            int samplingRate = 48000,
            int channels = 1 )
```

Create a new ToneAudioReader instance

Parameters

clockSec	Function to get current time in seconds. In Unity, pass in '() => AudioSettings.dspTime' for better results.
frequency	Frequency of the generated tone (in Hz).
samplingRate	Sampling rate of the audio signal (in Hz).
channels	Number of channels in the audio signal.

3.142 UnityAudioOut Class Reference

Inherits AudioOutDelayControl< float >.

Public Member Functions

- **UnityAudioOut** (AudioSource audioSource, PlayDelayConfig playDelayConfig, ILogger logger, string logPrefix, bool debugInfo)
- override void **OutCreate** (int frequency, int channels, int bufferSamples)
- override void **OutStart** ()
- override void **OutWrite** (float[ ] data, int offsetSamples)
- override void **Stop** ()

Protected Attributes

- readonly AudioSource **source**
- AudioClip **clip**

Properties

- override long **OutPos** `[get]`

3.143 UnityLogger Class Reference

Static Public Member Functions

- static void **Log** (DebugLevel level, Object obj, string tag, string objName, string fmt, params object[ ] args)

3.144 UnityMicrophone Class Reference

A wrapper around UnityEngine.Microphone to be able to safely use Microphone and compile for WebGL.

Static Public Member Functions

- static void **End** (string deviceName)
- static void **GetDeviceCaps** (string deviceName, out int minFreq, out int maxFreq)
- static int **GetPosition** (string deviceName)
- static bool **IsRecording** (string deviceName)
- static AudioClip **Start** (string deviceName, bool loop, int lengthSec, int frequency)
- static string **CheckDevice** (Voice.ILogger logger, string logPref, string device, int suggestedFrequency, out int frequency)

Properties

- static string[ ] **devices** `[get]`

3.144.1 Detailed Description

A wrapper around UnityEngine.Microphone to be able to safely use Microphone and compile for WebGL.

3.145 UnityVoiceClient Class Reference

Component that represents a Voice client and manages a simple Unity integration: a single Recorder and multiple remote speakers.

Inherits VoiceConnection.

Public Member Functions

- override bool ConnectUsingSettings (AppSettings overwriteSettings=null)

  Connect to Photon server using Settings

Public Attributes

- override bool **AlwaysUsePrimaryRecorder** => true
- bool UseVoiceAppSettings = false

  Whether or not to use the Voice AppId and all the other AppSettings from Fusion's RealtimeAppSettings Scriptable↩ Object singleton in the Voice client/app.

Protected Member Functions

- void **Start** ()
- override Speaker **InstantiateSpeakerForRemoteVoice** (int playerId, byte voiceId, object userData)

Additional Inherited Members

3.145.1 Detailed Description

Component that represents a Voice client and manages a simple Unity integration: a single Recorder and multiple remote speakers.

3.145.2 Member Function Documentation

3.145.2.1 ConnectUsingSettings()

```
override bool ConnectUsingSettings (
            AppSettings overwriteSettings = null )  [virtual]
```

Connect to Photon server using Settings

Parameters

overwriteSettings	Overwrites Settings before connecting

Returns

If true voice connection command was sent from client

Reimplemented from VoiceConnection.

3.145.3 Member Data Documentation

3.145.3.1 UseVoiceAppSettings

```
bool UseVoiceAppSettings = false
```

Whether or not to use the Voice AppId and all the other AppSettings from Fusion's RealtimeAppSettings ScriptableObject singleton in the Voice client/app.

3.146 UnsupportedCodecException Class Reference

Exception thrown if an unsupported codec is encountered.

Inherits Exception.

Public Member Functions

- UnsupportedCodecException (string info, Codec codec)

  Create a new UnsupportedCodecException.

3.146.1 Detailed Description

Exception thrown if an unsupported codec is encountered.

3.146.2 Constructor & Destructor Documentation

3.146.2.1 UnsupportedCodecException()

```
UnsupportedCodecException (
            string info,
            Codec codec )
```

Create a new UnsupportedCodecException.

Parameters

info	The info prepending standard message.
codec	The codec actually encountered.

3.147 UnsupportedPlatformException Class Reference

Exception thrown if an unsupported platform is encountered.

Inherits Exception.

Public Member Functions

- UnsupportedPlatformException (string subject, string platform=null)

  Create a new UnsupportedPlatformException.

3.147.1 Detailed Description

Exception thrown if an unsupported platform is encountered.

3.147.2 Constructor & Destructor Documentation

3.147.2.1 UnsupportedPlatformException()

```
UnsupportedPlatformException (
            string subject,
            string platform = null )
```

Create a new UnsupportedPlatformException.

Parameters

subject	The info prepending standard message.

///

Parameters

platform	Optional platform name.

3.148 UnsupportedSampleTypeException Class Reference

Exception thrown if an unsupported audio sample type is encountered.

Inherits Exception.

Public Member Functions

- UnsupportedSampleTypeException (Type t)

  Create a new UnsupportedSampleTypeException.

3.148.1 Detailed Description

Exception thrown if an unsupported audio sample type is encountered.

PhotonVoice generally supports 32-bit floating point ("float") or 16-bit signed integer ("short") audio, but it usually won't be converted automatically due to the high CPU overhead (and potential loss of precision) involved.

3.148.2 Constructor & Destructor Documentation

3.148.2.1 UnsupportedSampleTypeException()

```
UnsupportedSampleTypeException (
            Type t )
```

Create a new UnsupportedSampleTypeException.

Parameters

t	The sample type actually encountered.

3.149 OpusCodec.Util Class Reference

3.150 VideoInEnumerator Class Reference

Inherits DeviceEnumerator.

Public Member Functions

- **VideoInEnumerator** (ILogger logger)

Additional Inherited Members

3.151 VideoInEnumerator Class Reference

Inherits DeviceEnumeratorBase.

Public Member Functions

- **VideoInEnumerator** (ILogger logger)
- override void **Refresh** ()
- override void **Dispose** ()

Properties

- override string **Error** `[get]`

Additional Inherited Members

3.152 VoiceClient Class Reference

Voice client interact with other clients on network via IVoiceTransport.

Inherits IDisposable.

Classes

- struct CreateOptions

Public Member Functions

- delegate void RemoteVoiceInfoDelegate (int channelId, int playerId, byte voiceId, VoiceInfo voiceInfo, ref RemoteVoiceOptions options)

  Remote voice info event delegate.
- IEnumerable< LocalVoice > LocalVoicesInChannel (int channelId)

  Iterates through copy of all local voices list of given channel.
- void **LogSpacingProfiles** ()
- void **LogStats** ()
- void **SetRemoteVoiceDelayFrames** (Codec codec, int delayFrames)
- VoiceClient (IVoiceTransport transport, ILogger logger, CreateOptions opt=default(CreateOptions))

  Creates VoiceClient instance
- void Service ()

  This method dispatches all available incoming commands and then sends this client's outgoing commands. Call this method regularly (2..20 times a second).
- LocalVoice CreateLocalVoice (VoiceInfo voiceInfo, int channelId=0, IEncoder encoder=null)

  Creates basic outgoing stream w/o data processing support. Provided encoder should generate output data stream.
- LocalVoiceFramed< T > CreateLocalVoiceFramed< T > (VoiceInfo voiceInfo, int frameSize, int channelId=0, IEncoder encoder=null)

  Creates outgoing stream consuming sequence of values passed in array buffers of arbitrary length which repacked in frames of constant length for further processing and encoding.
- LocalVoiceAudio< T > **CreateLocalVoiceAudio< T >** (VoiceInfo voiceInfo, IAudioDesc audioSourceDesc, IEncoder encoder, int channelId)
- LocalVoice CreateLocalVoiceAudioFromSource (VoiceInfo voiceInfo, IAudioDesc source, AudioSampleType sampleType, IEncoder encoder=null, int channelId=0)

  Creates outgoing audio stream of type automatically assigned and adds procedures (callback or serviceable) for consuming given audio source data. Adds audio specific features (e.g. resampling, level meter) to processing pipeline and to returning stream handler.
- LocalVoiceVideo CreateLocalVoiceVideo (VoiceInfo voiceInfo, IVideoRecorder recorder, int channelId=0)

  Creates outgoing video stream consuming sequence of image buffers.
- void RemoveLocalVoice (LocalVoice voice)

  Removes local voice (outgoing data stream).

Parameters

voice	*Handler of outgoing stream to be removed.*

- void **onJoinChannel** (int channel)
- void **onLeaveChannel** (int channel)
- void **onLeaveAllChannels** ()
- void **onPlayerJoin** (int channelId, int playerId)
- void **onPlayerLeave** (int channelId, int playerId)
- void **onVoiceInfo** (int channelId, int playerId, byte voiceId, byte eventNumber, VoiceInfo info)
- void **onVoiceRemove** (int channelId, int playerId, byte[ ] voiceIds)
- void **onFrame** (int channelId, int playerId, byte voiceId, byte evNumber, ref FrameBuffer receivedBytes, bool isLocalPlayer)
- void **Dispose** ()

Properties

- int FramesLost `[get, set]`

  Lost frames counter.
- int FramesReceived `[get]`

Received frames counter.

- int FramesSent `[get]`

  Sent frames counter.
- int FramesSentBytes `[get]`

  Sent frames bytes counter.
- int RoundTripTime `[get]`

  Average time required voice packet to return to sender.
- int RoundTripTimeVariance `[get]`

  Average round trip time variation.
- bool SuppressInfoDuplicateWarning `[get, set]`

  Do not log warning when duplicate info received.
- RemoteVoiceInfoDelegate OnRemoteVoiceInfoAction `[get, set]`

  Register a method to be called when remote voice info arrived (after join or new new remote voice creation). Metod parameters: (int channelId, int playerId, byte voiceId, VoiceInfo voiceInfo, ref RemoteVoiceOptions options);
- int DebugLostPercent `[get, set]`

  Lost frames simulation ratio.
- IEnumerable< LocalVoice > LocalVoices `[get]`

  Iterates through copy of all local voices list.
- IEnumerable< RemoteVoiceInfo > RemoteVoiceInfos `[get]`

  Iterates through all remote voices infos.

3.152.1 Detailed Description

Voice client interact with other clients on network via IVoiceTransport.

3.152.2 Constructor & Destructor Documentation

3.152.2.1 VoiceClient()

```
VoiceClient (
            IVoiceTransport transport,
            ILogger logger,
            CreateOptions opt = default(CreateOptions) )
```

Creates VoiceClient instance

3.152.3 Member Function Documentation

3.152.3.1 CreateLocalVoice()

```
LocalVoice CreateLocalVoice (
            VoiceInfo voiceInfo,
            int channelId = 0,
            IEncoder encoder = null )
```

Creates basic outgoing stream w/o data processing support. Provided encoder should generate output data stream.

Parameters

voiceInfo	Outgoing stream parameters.
channel↩ Id	Transport channel specific to transport.
encoder	Encoder producing the stream.

Returns

Outgoing stream handler.

3.152.3.2 CreateLocalVoiceAudioFromSource()

```
LocalVoice CreateLocalVoiceAudioFromSource (
             VoiceInfo voiceInfo,
             IAudioDesc source,
             AudioSampleType sampleType,
             IEncoder encoder = null,
             int channelId = 0 )
```

Creates outgoing audio stream of type automatically assigned and adds procedures (callback or serviceable) for consuming given audio source data. Adds audio specific features (e.g. resampling, level meter) to processing pipeline and to returning stream handler.

Parameters

voiceInfo	Outgoing stream parameters.
source	Streaming audio source.
sampleType	Voice's audio sample type. If does not match source audio sample type, conversion will occur.
channelId	Transport channel specific to transport.
encoder	Audio encoder. Set to null to use default Opus encoder.

Returns

Outgoing stream handler.

audioSourceDesc.SamplingRate and voiceInfo.SamplingRate may do not match. Automatic resampling will occur in this case.

3.152.3.3 CreateLocalVoiceFramed< T >()

```
LocalVoiceFramed<T> CreateLocalVoiceFramed< T > (
             VoiceInfo voiceInfo,
             int frameSize,
             int channelId = 0,
             IEncoder encoder = null )
```

Creates outgoing stream consuming sequence of values passed in array buffers of arbitrary length which repacked in frames of constant length for further processing and encoding.

Template Parameters

T	Type of data consumed by outgoing stream (element type of array buffers).

Parameters

voiceInfo	Outgoing stream parameters.
frameSize	Size of buffer LocalVoiceFramed repacks input data stream to.
channel↩ Id	Transport channel specific to transport.
encoder	Encoder compressing data stream in pipeline.

Returns

Outgoing stream handler.

3.152.3.4 CreateLocalVoiceVideo()

```
LocalVoiceVideo CreateLocalVoiceVideo (
            VoiceInfo voiceInfo,
            IVideoRecorder recorder,
            int channelId = 0 )
```

Creates outgoing video stream consuming sequence of image buffers.

Parameters

voiceInfo	Outgoing stream parameters.
recorder	Video recorder.
channel↩ Id	Transport channel specific to transport.

Returns

Outgoing stream handler.

3.152.3.5 LocalVoicesInChannel()

```
IEnumerable<LocalVoice> LocalVoicesInChannel (
            int channelId )
```

Iterates through copy of all local voices list of given channel.

3.152.3.6 RemoteVoiceInfoDelegate()

```
delegate void RemoteVoiceInfoDelegate (
             int channelId,
             int playerId,
             byte voiceId,
             VoiceInfo voiceInfo,
             ref RemoteVoiceOptions options )
```

Remote voice info event delegate.

3.152.3.7 RemoveLocalVoice()

```
void RemoveLocalVoice (
             LocalVoice voice )
```

Removes local voice (outgoing data stream).

Parameters

voice	Handler of outgoing stream to be removed.

3.152.3.8 Service()

```
void Service ( )
```

This method dispatches all available incoming commands and then sends this client's outgoing commands. Call this method regularly (2..20 times a second).

3.152.4 Property Documentation

3.152.4.1 DebugLostPercent

```
int DebugLostPercent  [get], [set]
```

Lost frames simulation ratio.

3.152.4.2 FramesLost

```
int FramesLost  [get], [set]
```

Lost frames counter.

3.152.4.3 FramesReceived

```
int FramesReceived  [get]
```

Received frames counter.

3.152.4.4 FramesSent

```
int FramesSent  [get]
```

Sent frames counter.

3.152.4.5 FramesSentBytes

```
int FramesSentBytes  [get]
```

Sent frames bytes counter.

3.152.4.6 LocalVoices

```
IEnumerable<LocalVoice> LocalVoices  [get]
```

Iterates through copy of all local voices list.

3.152.4.7 OnRemoteVoiceInfoAction

```
RemoteVoiceInfoDelegate OnRemoteVoiceInfoAction  [get], [set]
```

Register a method to be called when remote voice info arrived (after join or new new remote voice creation). Metod parameters: (int channelId, int playerId, byte voiceId, VoiceInfo voiceInfo, ref RemoteVoiceOptions options);

3.152.4.8 RemoteVoiceInfos

```
IEnumerable<RemoteVoiceInfo> RemoteVoiceInfos  [get]
```

Iterates through all remote voices infos.

3.152.4.9 RoundTripTime

```
int RoundTripTime  [get]
```

Average time required voice packet to return to sender.

3.152.4.10 RoundTripTimeVariance

```
int RoundTripTimeVariance  [get]
```

Average round trip time variation.

3.152.4.11 SuppressInfoDuplicateWarning

```
bool SuppressInfoDuplicateWarning  [get], [set]
```

Do not log warning when duplicate info received.

3.153 VoiceComponent Class Reference

Inherits MonoBehaviour.

Inherited by PhotonVoiceView, AudioChangesHandler, FMODRecorderSetup, Recorder, RecorderPreset, Speaker, MicAmplifier, MicrophonePermission, SaveIncomingStreamToFile, SaveOutgoingStreamToFile, and WebRtcAudioDsp.

Public Attributes

- VoiceLogger **VoiceLogger** => impl.VoiceLogger

Protected Member Functions

- virtual void **Awake** ()

Protected Attributes

- Voice.ILogger **Logger** => impl.Logger

Properties

- string **Name** `[set]`

3.154 VoiceComponentImpl Class Reference

Public Member Functions

- void **Awake** (MonoBehaviour mb)

Public Attributes

- Voice.ILogger **Logger** => logger
- VoiceLogger **VoiceLogger** => voiceLogger

Properties

- string **Name** `[set]`

3.155 VoiceConnection Class Reference

Component that represents a Voice client.

Inherits ConnectionHandler.

Inherited by FusionVoiceClient, PunVoiceClient, and UnityVoiceClient.

Public Member Functions

- virtual bool ConnectUsingSettings (AppSettings overwriteSettings=null)

  Connect to Photon server using Settings
- bool AddSpeaker (Speaker speaker, object userData)

  Tries to link local Speaker with remote voice stream using UserData. Useful if Speaker created after stream is started.
- Speaker InstantiateSpeakerPrefab (GameObject parent, bool destroyOnRemove)

  Instantiates SpeakerPrefab, optionally attaches it to the provided parent.
- bool **AddRecorder** (Recorder rec)
- void **RemoveRecorder** (Recorder rec)

Public Attributes

- virtual bool **AlwaysUsePrimaryRecorder** => false
- AppSettings Settings

  Settings to be used by this Voice Client
- VoiceLogger **VoiceLogger** => voiceComponentImpl.VoiceLogger
- bool UsePrimaryRecorder => this.usePrimaryRecorder

  Use VoiceConnection.PrimaryRecorder directly.

Protected Member Functions

- override void **Awake** ()
- virtual void **Update** ()
- virtual void **FixedUpdate** ()
- virtual void **OnDestroy** ()
- virtual Speaker **InstantiateSpeakerForRemoteVoice** (int playerId, byte voiceId, object userData)
- virtual void **OnVoiceStateChanged** (ClientState fromState, ClientState toState)
- void **CalcStatistics** ()
- virtual void **OnOperationResponseReceived** (OperationResponse operationResponse)

Protected Attributes

- Voice.ILogger **Logger** => voiceComponentImpl.Logger

Properties

- new LoadBalancingTransport **Client** `[get]`
- VoiceClient VoiceClient `[get]`

  Returns underlying Photon Voice client.
- ClientState ClientState `[get]`

  Returns Photon Voice client state.
- float FramesReceivedPerSecond `[get]`

  Number of frames received per second.
- float FramesLostPerSecond `[get]`

  Number of frames lost per second.
- float FramesLostPercent `[get]`

  Percentage of lost frames.
- GameObject SpeakerPrefab `[get, set]`

  Prefab that contains Speaker component to be instantiated when receiving a new remote audio source info
- Recorder PrimaryRecorder `[get, set]`

  Primary Recorder to be used by VoiceConnection implementations directly or via integration objects.
- string BestRegionSummaryInPreferences `[get, set]`

  Used to store and access the "Best Region Summary" in the Player Preferences.

Events

- Action< Speaker > SpeakerLinked

  Fires when a speaker has been linked to a remote audio stream
- Action< RemoteVoiceLink > RemoteVoiceAdded

  Fires when a remote voice stream is added

3.155.1 Detailed Description

Component that represents a Voice client.

3.155.2 Member Function Documentation

3.155.2.1 AddSpeaker()

```
bool AddSpeaker (
            Speaker speaker,
            object userData )
```

Tries to link local Speaker with remote voice stream using UserData. Useful if Speaker created after stream is started.

Parameters

speaker	Speaker ot try linking.
userData	UserData object used to bind local Speaker with remote voice stream.

Returns

3.155.2.2 ConnectUsingSettings()

```
virtual bool ConnectUsingSettings (
            AppSettings overwriteSettings = null )  [virtual]
```

Connect to Photon server using Settings

Parameters

overwriteSettings	Overwrites Settings before connecting

Returns

If true voice connection command was sent from client

Reimplemented in UnityVoiceClient.

3.155.2.3 InstantiateSpeakerPrefab()

```
Speaker InstantiateSpeakerPrefab (
            GameObject parent,
            bool destroyOnRemove )
```

Instantiates SpeakerPrefab, optionally attaches it to the provided parent.

VoiceConnection manages the instantiated object (destroys on OnRemoteVoiceRemoveAction).

Parameters

parent	The object to attach Steaker to.
destroyOnRemove	Automatically destroy instantiated prefab when remote voice is removed (the caller does not manages the instance).

Returns

Instantiated Speaker or null.

3.155.3 Member Data Documentation

3.155.3.1 Settings

```
AppSettings Settings
```

Settings to be used by this Voice Client

3.155.3.2 UsePrimaryRecorder

```
bool UsePrimaryRecorder => this.usePrimaryRecorder
```

Use VoiceConnection.PrimaryRecorder directly.

3.155.4 Property Documentation

3.155.4.1 BestRegionSummaryInPreferences

```
string BestRegionSummaryInPreferences [get], [set]
```

Used to store and access the "Best Region Summary" in the Player Preferences.

3.155.4.2 ClientState

```
ClientState ClientState  [get]
```

Returns Photon Voice client state.

3.155.4.3 FramesLostPercent

```
float FramesLostPercent  [get]
```

Percentage of lost frames.

3.155.4.4 FramesLostPerSecond

```
float FramesLostPerSecond  [get]
```

Number of frames lost per second.

3.155.4.5 FramesReceivedPerSecond

```
float FramesReceivedPerSecond  [get]
```

Number of frames received per second.

3.155.4.6 PrimaryRecorder

```
Recorder PrimaryRecorder  [get], [set]
```

Primary Recorder to be used by VoiceConnection implementations directly or via integration objects.

3.155.4.7 SpeakerPrefab

```
GameObject SpeakerPrefab  [get], [set]
```

Prefab that contains Speaker component to be instantiated when receiving a new remote audio source info

3.155.4.8 VoiceClient

```
VoiceClient VoiceClient  [get]
```

Returns underlying Photon Voice client.

3.155.5 Event Documentation

3.155.5.1 RemoteVoiceAdded

```
Action<RemoteVoiceLink> RemoteVoiceAdded
```

Fires when a remote voice stream is added

3.155.5.2 SpeakerLinked

```
Action<Speaker> SpeakerLinked
```

Fires when a speaker has been linked to a remote audio stream

3.156 VoiceDebugScript Class Reference

Utility script to be attached next to PhotonVoiceView & PhotonView on the player prefab to be network instantiated. Call voiceDebugScript.CantHearYou() on the networked object of the remote (or local) player if you can't hear the corresponding player.

Inherits MonoBehaviourPun.

Public Member Functions

- void **CantHearYou** ()

Public Attributes

- bool ForceRecordingAndTransmission

  Make sure recorder.TransmitEnabled and recorder.RecordingEnabled are true.
- AudioClip TestAudioClip

  Audio file to be broadcast when TestUsingAudioClip is enabled.
- bool TestUsingAudioClip

  Broadcast Audio file to make sure transmission over network works if microphone (audio input device/hardware) is not reliable. Requires setting AudioClip in TestAudioClip.
- bool DisableVad

  Disable recorder.VoiceDetection for easier testing.
- bool IncreaseLogLevels

  Set main voice component's log level to ALL (max).
- bool LocalDebug

  Debug DebugEcho mode (Can't Hear My Self?!).

3.156.1 Detailed Description

Utility script to be attached next to PhotonVoiceView & PhotonView on the player prefab to be network instantiated. Call voiceDebugScript.CantHearYou() on the networked object of the remote (or local) player if you can't hear the corresponding player.

3.156.2 Member Data Documentation

3.156.2.1 DisableVad

```
bool DisableVad
```

Disable recorder.VoiceDetection for easier testing.

3.156.2.2 ForceRecordingAndTransmission

```
bool ForceRecordingAndTransmission
```

Make sure recorder.TransmitEnabled and recorder.RecordingEnabled are true.

3.156.2.3 IncreaseLogLevels

```
bool IncreaseLogLevels
```

Set main voice component's log level to ALL (max).

3.156.2.4 LocalDebug

```
bool LocalDebug
```

Debug DebugEcho mode (Can't Hear My Self?!).

3.156.2.5 TestAudioClip

```
AudioClip TestAudioClip
```

Audio file to be broadcast when TestUsingAudioClip is enabled.

3.156.2.6 TestUsingAudioClip

```
bool TestUsingAudioClip
```

Broadcast Audio file to make sure transmission over network works if microphone (audio input device/hardware) is not reliable. Requires setting AudioClip in TestAudioClip.

3.157 AudioUtil.VoiceDetector< T > Class Template Reference

Simple voice activity detector triggered by signal level.

Inherits IProcessor< T >, and AudioUtil.IVoiceDetector.

Public Member Functions

- abstract T[ ] Process (T[ ] buf)

  Process a frame of audio data.
- void **Dispose** ()

Protected Attributes

- float **norm**
- float **threshold**
- int **activityDelay**
- int **autoSilenceCounter** = 0
- int **valuesCountPerSec**
- int **activityDelayValuesCount**

Properties

- bool On `[get, set]`

  If true, voice detection enabled.
- float Threshold `[get, set]`

  Voice detected as soon as signal level exceeds threshold.
- bool Detected `[get, protected set]`

  If true, voice detected.
- DateTime DetectedTime `[get]`

  Last time when switched to detected state.
- int ActivityDelayMs `[get, set]`

  Keep detected state during this time after signal level dropped below threshold.

Events

- Action OnDetected

  Called when switched to detected state.

3.157.1 Detailed Description

Simple voice activity detector triggered by signal level.

3.157.2 Member Function Documentation

3.157.2.1 Process()

```
abstract T [] Process (
            T[] buf )  [pure virtual]
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implements IProcessor< T >.

3.157.3 Property Documentation

3.157.3.1 ActivityDelayMs

```
int ActivityDelayMs  [get], [set]
```

Keep detected state during this time after signal level dropped below threshold.

3.157.3.2 Detected

```
bool Detected  [get], [protected set]
```

If true, voice detected.

3.157.3.3 DetectedTime

```
DateTime DetectedTime  [get]
```

Last time when switched to detected state.

3.157.3.4 On

```
bool On  [get], [set]
```

If true, voice detection enabled.

3.157.3.5 Threshold

```
float Threshold  [get], [set]
```

Voice detected as soon as signal level exceeds threshold.

3.157.4 Event Documentation

3.157.4.1 OnDetected

```
Action OnDetected
```

Called when switched to detected state.

3.158 AudioUtil.VoiceDetectorCalibration< T > Class Template Reference

Calibration Utility for Voice Detector

Inherits IProcessor< T >.

Public Member Functions

- VoiceDetectorCalibration (IVoiceDetector voiceDetector, ILevelMeter levelMeter, int samplingRate, int channels)

    Create new VoiceDetectorCalibration instance.
- void Calibrate (int durationMs, Action< float > onCalibrated=null)

    Start calibration.
- T[ ] Process (T[ ] buf)

    Process a frame of audio data.
- void **Dispose** ()

Protected Attributes

- int **calibrateCount**

Properties

- bool **IsCalibrating** [get]

3.158.1 Detailed Description

Calibration Utility for Voice Detector

. Using this audio processor, you can calibrate the IVoiceDetector.Threshold.

3.158.2 Constructor & Destructor Documentation

3.158.2.1 VoiceDetectorCalibration()

```
VoiceDetectorCalibration (
            IVoiceDetector voiceDetector,
            ILevelMeter levelMeter,
            int samplingRate,
            int channels )
```

Create new VoiceDetectorCalibration instance.

Parameters

voiceDetector	Voice Detector to calibrate.
levelMeter	Level Meter to look at for calibration.
samplingRate	Sampling rate of the audio signal (in Hz).
channels	Number of channels in the audio signal.

3.158.3 Member Function Documentation

3.158.3.1 Calibrate()

```
void Calibrate (
            int durationMs,
            Action< float > onCalibrated = null )
```

Start calibration.

Parameters

durationMs	Duration of the calibration procedure (in milliseconds).
onCalibrated	Optional callback that is called after calibration is complete.

This activates the Calibration process. It will reset the given LevelMeter's AccumAvgPeakAmp (accumulated average peak amplitude), and when the duration has passed, use it for the VoiceDetector's detection threshold.

3.158.3.2 Process()

```
T [] Process (
            T[] buf )
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implements IProcessor< T >.

3.159 AudioUtil.VoiceDetectorDummy Class Reference

Dummy VoiceDetector that doesn't actually do anything.

Inherits AudioUtil.IVoiceDetector.

Properties

- bool **On** `[get, set]`
- float **Threshold** `[get, set]`
- bool **Detected** `[get]`
- int **ActivityDelayMs** `[get, set]`
- DateTime **DetectedTime** `[get]`
- Action **OnDetected**

Additional Inherited Members

3.159.1 Detailed Description

Dummy VoiceDetector that doesn't actually do anything.

3.160 AudioUtil.VoiceDetectorFloat Class Reference

VoiceDetector specialization for float audio.

Inherits AudioUtil.VoiceDetector< float >.

Public Member Functions

- VoiceDetectorFloat (int samplingRate, int numChannels)

  Create a new VoiceDetectorFloat instance.
- override float[ ] **Process** (float[ ] buffer)

Additional Inherited Members

3.160.1 Detailed Description

VoiceDetector specialization for float audio.

3.160.2 Constructor & Destructor Documentation

3.160.2.1 VoiceDetectorFloat()

```
VoiceDetectorFloat (
              int samplingRate,
              int numChannels )
```

Create a new VoiceDetectorFloat instance.

Parameters

samplingRate	Sampling rate of the audio signal (in Hz).
numChannels	Number of channels in the audio signal.

3.161 AudioUtil.VoiceDetectorShort Class Reference

VoiceDetector specialization for float audio.

Inherits AudioUtil.VoiceDetector< short >.

Public Member Functions

- VoiceDetectorShort (int samplingRate, int numChannels)

  Create a new VoiceDetectorFloat instance
- override short[ ] **Process** (short[ ] buffer)

Additional Inherited Members

3.161.1 Detailed Description

VoiceDetector specialization for float audio.

3.161.2 Constructor & Destructor Documentation

3.161.2.1 VoiceDetectorShort()

```
VoiceDetectorShort (
              int samplingRate,
              int numChannels )
```

Create a new VoiceDetectorFloat instance

Parameters

samplingRate	Sampling rate of the audio signal (in Hz).
numChannels	Number of channels in the audio signal.

3.162 VoiceEvent Class Reference

Static Public Attributes

- const byte Code = 202

  Single event used for voice communications.
- const byte **FrameCode** = 203

3.162.1 Member Data Documentation

3.162.1.1 Code

```
const byte Code = 202 [static]
```

Single event used for voice communications.

Change if it conflicts with other event codes used in the same Photon room.

3.163 VoiceInfo Struct Reference

Describes stream properties.

Public Member Functions

- override string **ToString** ()

Static Public Member Functions

- static VoiceInfo CreateAudioOpus (POpusCodec.Enums.SamplingRate samplingRate, int channels, OpusCodec.FrameDuration frameDurationUs, int bitrate, object userdata=null)

  Create stream info for an Opus audio stream.
- static VoiceInfo CreateAudio (Codec codec, int samplingRate, int channels, int frameDurationUs, object userdata=null)

  Create stream info for an audio stream.
- static VoiceInfo CreateVideo (Codec codec, int bitrate, int width, int heigth, int fps, int keyFrameInt, object userdata=null)

  Create stream info for a video stream.

Properties

- Codec **Codec** `[get, set]`
- int SamplingRate `[get, set]`

  Audio sampling rate (frequency, in Hz).
- int Channels `[get, set]`

  Number of channels.
- int FrameDurationUs `[get, set]`

  Uncompressed frame (audio packet) size in microseconds.
- int Bitrate `[get, set]`

  Target bitrate (in bits/second).
- int Width `[get, set]`

  Video width.
- int Height `[get, set]`

  Video height
- int FPS `[get, set]`

  Video frames per second
- int KeyFrameInt `[get, set]`

  Video keyframe interval in frames
- object UserData `[get, set]`

  Optional user data. Should be serializable by Photon.
- int FrameDurationSamples `[get]`

  Uncompressed frame (data packet) size in samples.
- int FrameSize `[get]`

  Uncompressed frame (data packet) array size.

3.163.1 Detailed Description

Describes stream properties.

3.163.2 Member Function Documentation

3.163.2.1 CreateAudio()

```
static VoiceInfo CreateAudio (
            Codec codec,
            int samplingRate,
            int channels,
            int frameDurationUs,
            object userdata = null )  [static]
```

Create stream info for an audio stream.

Parameters

codec	Audio codec.
samplingRate	Audio sampling rate.
channels	Number of channels.
frameDurationUs	Uncompressed frame (audio packet) size in microseconds.
userdata	Optional user data. Should be serializable by Photon.

Returns

VoiceInfo instance.

3.163.2.2 CreateAudioOpus()

```
static VoiceInfo CreateAudioOpus (
            POpusCodec.Enums.SamplingRate samplingRate,
            int channels,
            OpusCodec.FrameDuration frameDurationUs,
            int bitrate,
            object userdata = null )  [static]
```

Create stream info for an Opus audio stream.

Parameters

samplingRate	Audio sampling rate.
channels	Number of channels.
frameDurationUs	Uncompressed frame (audio packet) size in microseconds.
bitrate	Stream bitrate (in bits/second).
userdata	Optional user data. Should be serializable by Photon.

Returns

VoiceInfo instance.

3.163.2.3 CreateVideo()

```
static VoiceInfo CreateVideo (
             Codec codec,
             int bitrate,
             int width,
             int heigth,
             int fps,
             int keyFrameInt,
             object userdata = null )  [static]
```

Create stream info for a video stream.

Parameters

codec	Video codec.
bitrate	Stream bitrate.
width	Streamed video width. If 0, width and height of video source used (no rescaling).
heigth	Streamed video height. If -1, aspect ratio preserved during rescaling.
fps	Streamed video frames per second.
keyFrameInt	Keyframes interval in frames.

///

Parameters

userdata	Optional user data. Should be serializable by Photon.

Returns

VoiceInfo instance.

3.163.3 Property Documentation

3.163.3.1 Bitrate

```
int Bitrate  [get], [set]
```

Target bitrate (in bits/second).

3.163.3.2 Channels

```
int Channels  [get], [set]
```

Number of channels.

3.163.3.3 FPS

```
int FPS  [get], [set]
```

Video frames per second

3.163.3.4 FrameDurationSamples

```
int FrameDurationSamples  [get]
```

Uncompressed frame (data packet) size in samples.

3.163.3.5 FrameDurationUs

```
int FrameDurationUs  [get], [set]
```

Uncompressed frame (audio packet) size in microseconds.

3.163.3.6 FrameSize

```
int FrameSize  [get]
```

Uncompressed frame (data packet) array size.

3.163.3.7 Height

```
int Height  [get], [set]
```

Video height

3.163.3.8 KeyFrameInt

```
int KeyFrameInt  [get], [set]
```

Video keyframe interval in frames

3.163.3.9 SamplingRate

```
int SamplingRate  [get], [set]
```

Audio sampling rate (frequency, in Hz).

3.163.3.10 UserData

```
object UserData  [get], [set]
```

Optional user data. Should be serializable by Photon.

3.163.3.11 Width

```
int Width  [get], [set]
```

Video width.

3.164 AudioUtil.VoiceLevelDetectCalibrate< T > Class Template Reference

Utility Audio Processor Voice Detection Calibration.

Inherits IProcessor< T >.

Public Member Functions

- VoiceLevelDetectCalibrate (int samplingRate, int channels)

  Create new VoiceLevelDetectCalibrate instance
- void Calibrate (int durationMs, Action< float > onCalibrated=null)

  Start calibration
- T[ ] Process (T[ ] buf)

  Process a frame of audio data.
- void **Dispose** ()

Properties

- ILevelMeter LevelMeter `[get]`

  The LevelMeter in use.
- IVoiceDetector VoiceDetector `[get]`

  The VoiceDetector in use
- bool **IsCalibrating** `[get]`

3.164.1 Detailed Description

Utility Audio Processor Voice Detection Calibration.

Encapsulates level meter, voice detector and voice detector calibrator in single instance.

3.164.2 Constructor & Destructor Documentation

3.164.2.1 VoiceLevelDetectCalibrate()

```
VoiceLevelDetectCalibrate (
            int samplingRate,
            int channels )
```

Create new VoiceLevelDetectCalibrate instance

Parameters

samplingRate	Sampling rate of the audio signal (in Hz).
channels	Number of channels in the audio signal.

3.164.3 Member Function Documentation

3.164.3.1 Calibrate()

```
void Calibrate (
            int durationMs,
            Action< float > onCalibrated = null )
```

Start calibration

Parameters

durationMs	Duration of the calibration procedure (in milliseconds).
onCalibrated	Called when calibration is complete. Parameter is new threshold value.

This activates the Calibration process. It will reset the given LevelMeter's AccumAvgPeakAmp (accumulated average peak amplitude), and when the duration has passed, use it for the VoiceDetector's detection threshold.

3.164.3.2 Process()

```
T [] Process (
            T[] buf )
```

Process a frame of audio data.

Parameters

buf	Buffer containing input audio data

Returns

Buffer containing output audio data or null if frame has been discarded (VAD)

Implements IProcessor< T >.

3.164.4 Property Documentation

3.164.4.1 LevelMeter

```
ILevelMeter LevelMeter  [get]
```

The LevelMeter in use.

3.164.4.2 VoiceDetector

```
IVoiceDetector VoiceDetector  [get]
```

The VoiceDetector in use

3.165 VoiceLogger Class Reference

Inherits MonoBehaviour.

Static Public Member Functions

- static VoiceLogger **FindLogger** (GameObject gameObject)
- static VoiceLogger **CreateRootLogger** ()

Public Attributes

- DebugLevel **LogLevel** = DebugLevel.WARNING

3.166 VoiceNetworkObject Class Reference

Inherits NetworkBehaviour.

Public Member Functions

- override void **Spawned** ()
- override void **Despawned** (NetworkRunner runner, bool hasState)

Public Attributes

- VoiceLogger **VoiceLogger** => voiceComponentImpl.VoiceLogger
- bool IsSpeaking => this.SpeakerInUse != null && this.SpeakerInUse.IsPlaying

  If true, this VoiceNetworkObject has a Speaker that is currently playing received audio frames from remote audio source
- bool IsRecording => this.RecorderInUse != null && this.RecorderInUse.IsCurrentlyTransmitting

  If true, this VoiceNetworkObject has a Recorder that is currently transmitting audio stream from local audio source

Protected Attributes

- Voice.ILogger **Logger** => voiceComponentImpl.Logger

Properties

- Recorder RecorderInUse `[get]`

  The Recorder component currently used by this VoiceNetworkObject
- Speaker SpeakerInUse `[get]`

  The Speaker component currently used by this VoiceNetworkObject

3.166.1 Member Data Documentation

3.166.1.1 IsRecording

```
bool IsRecording => this.RecorderInUse != null && this.RecorderInUse.IsCurrentlyTransmitting
```

If true, this VoiceNetworkObject has a Recorder that is currently transmitting audio stream from local audio source

3.166.1.2 IsSpeaking

```
bool IsSpeaking => this.SpeakerInUse != null && this.SpeakerInUse.IsPlaying
```

If true, this VoiceNetworkObject has a Speaker that is currently playing received audio frames from remote audio source

3.166.2 Property Documentation

3.166.2.1 RecorderInUse

```
Recorder RecorderInUse  [get]
```

The Recorder component currently used by this VoiceNetworkObject

3.166.2.2 SpeakerInUse

```
Speaker SpeakerInUse  [get]
```

The Speaker component currently used by this VoiceNetworkObject

3.167 AudioUtil.WaveformAudioPusher< T > Class Template Reference

IAudioPusher that provides the given waveform.

Inherits AudioUtil.GeneratorPusher< T >.

Public Member Functions

- **WaveformAudioPusher** (int bufSizeMs=100, int samplingRate=48000, int channels=1)

Protected Member Functions

- override int **Gen** (T[ ] buf, long timeSamples)

Properties

- T[ ] **Waveform** `[set]`

Additional Inherited Members

3.167.1 Detailed Description

IAudioPusher that provides the given waveform.

3.168 AudioUtil.WaveformAudioReader< T > Class Template Reference

IAudioReader that provides the given waveform.

Inherits AudioUtil.GeneratorReader< T >.

Public Member Functions

- **WaveformAudioReader** (Func< double > clockSec=null, int samplingRate=48000, int channels=1)

Protected Member Functions

- override int **Gen** (T[ ] buf, long timeSamples)

Properties

- T[ ] **Waveform** `[set]`

3.168.1 Detailed Description

IAudioReader that provides the given waveform.

3.169 WaveWriter Class Reference

Inherits IDisposable.

Public Member Functions

- **WaveWriter** (string fileName, int sampleRate, int bits, int channels)
- **WaveWriter** (Stream stream, int sampleRate, int bitsPerSample, int channels)
- void **Dispose** ()
- void **WriteSample** (float sample)
- void **WriteSamples** (float[ ] samples, int offset, int count)
- void **Write** (byte[ ] buffer, int offset, int count)
- void **Write** (byte value)
- void **Write** (short value)
- void **Write** (int value)
- void **Write** (float value)

Protected Member Functions

- virtual void **Dispose** (bool disposing)

3.170 WebRtcAudioDsp Class Reference

Inherits VoiceComponent.

Public Member Functions

- void **AdjustVoiceInfo** (ref VoiceInfo voiceInfo, ref AudioSampleType st)

Public Attributes

- bool **IsSupported**

Protected Member Functions

- override void **Awake** ()

Properties

- bool **AEC** `[get, set]`
- bool **AecHighPass** `[get, set]`
- int **ReverseStreamDelayMs** `[get, set]`
- bool **NoiseSuppression** `[get, set]`
- bool **HighPass** `[get, set]`
- bool **Bypass** `[get, set]`
- bool **AGC** `[get, set]`
- int **AgcCompressionGain** `[get, set]`
- int **AgcTargetLevel** `[get, set]`
- bool **VAD** `[get, set]`

Additional Inherited Members

3.170.1 Member Data Documentation

3.170.1.1 IsSupported

```
bool IsSupported
```

Initial value:

```
=>
        true
```

3.171 WebRTCAudioLib Class Reference

Inherited by WebRTCAudioProcessor.

Public Types

- enum **Error**
- enum **Param**

Public Member Functions

- static IntPtr **webrtc_audio_processor_create** (int samplingRate, int channels, int frameSize, int revSamplingRate, int revChannels)
- static int **webrtc_audio_processor_init** (IntPtr proc)
- static int **webrtc_audio_processor_set_param** (IntPtr proc, int param, int v)
- static int **webrtc_audio_processor_process** (IntPtr proc, short[ ] buffer, int offset, out bool voiceDetected)
- static int **webrtc_audio_processor_process_reverse** (IntPtr proc, short[ ] buffer, int bufferSize)
- static void **webrtc_audio_processor_destroy** (IntPtr proc)

3.172 WebRTCAudioProcessor Class Reference

Inherits WebRTCAudioLib, and IProcessor< short >.

Public Member Functions

- **WebRTCAudioProcessor** (ILogger logger, int frameSize, int samplingRate, int channels, int reverseSamplingRate, int reverseChannels)
- short[ ] **Process** (short[ ] buf)
- void **OnAudioOutFrameFloat** (float[ ] data)
- void **Dispose** ()

Static Public Attributes

- static readonly int[ ] **SupportedSamplingRates** = { 8000, 16000, 32000, 48000 }

Properties

- int **AECStreamDelayMs** `[set]`
- bool?? **AEC** `[set]`
- bool? **AECHighPass** `[set]`
- bool?? **AECMobile** `[set]`
- bool? **HighPass** `[set]`
- bool? **NoiseSuppression** `[set]`
- bool? **AGC** `[set]`
- int **AGCCompressionGain** `[set]`
- int **AGCTargetLevel** `[set]`
- bool? **AGC2** `[set]`
- bool? **VAD** `[set]`
- bool **Bypass** `[set]`

Additional Inherited Members

3.173 WindowsAudioInPusher Class Reference

Inherits IAudioPusher< short >.

Public Member Functions

- **WindowsAudioInPusher** (int deviceID, ILogger logger)
- void **SetCallback** (Action< short[ ]> callback, ObjectFactory< short[ ], int > bufferFactory)
- void **Dispose** ()

Properties

- int **Channels** `[get]`
- int **SamplingRate** `[get]`
- string **Error** `[get]`

Index