Google Developers

Google Tal X Search

melochale@gmail.com
Sign out

Products Google Talk for Developers

VoiceChannel Class

Wraps a **TransportChannel** and an associated **MediaEngine** object in a voice chat application. This object is created by the **ChannelManager** at the behest of the **Call** object when it creates a new **Session**. **VoiceChannel** is passes a **Session** object, but only uses it to create a **TransportChannel**.

VoiceChannel acts as the conduit for audio data between **TransportChannel** and **MediaChannel**, and can be used to stop or start the data flow (by muting or unmuting the data channel). Incoming data packets are received from **TransportChannel**, which calls **VoiceChannel::OnSocketRead**, which in turn calls **MediaChannel::OnPacketReceived**. Outbound packets are sent to **VoiceChannel::SendPacket**, which calls **TransportChannel::SendPacket**.

You should not need to modify override this class, although you could use it as a model for other rendered media session types (such as video).

Syntax

```
class VoiceChannel : public MessageHandler,
                     public sigslot::has_slots<>,
                     public MediaChannel::NetworkInterface
```

Methods

Name	Description
MediaChannel* **channel**()	Retrieves the **MediaChannel** object used to instantiate **VoiceChannel**.
void **Enable**(bool *enable*)	Enables or disables the voice channel, depending on the value passed to *enable*. True means enable, False means disable. (Enable means to send and receive audio data.)
int **GetInputLevel_w**()	Returns an input level value from the media engine. The range and meaning of this number is determined by the media engine implementation used.
int **GetOutputLevel_w**()	Returns an output level value from the media engine. The range and meaning of this number is determined by the media engine implementation used.
void **Mute**(bool *mute*)	Mutes and unmutes the outbound audio data, depending on the value passed in. True mutes the channel; False unmutes the channel. Note that incoming audio will still be rendered.
void **PauseMedia_w**()	Called by **ChannelManager** to temporarily stop the data flow so that it can change audio devices (**MediaEngine::SetSoundDevices**).
void **SendPacket**(const void **data*, size_t *len*)	Sends a packet of data of *len* bytes across the network. This is the implementation of the **MediaChannel::NetworkInterface::SendPacket** pure virtual method.
Session* **session**()	Returns the **Session** object associated with the **TransportChannel** managed by this object.
void **StartAudioMonitor**(int *cms*)	Starts sending audio information through **SignalAudioMonitor**. This is called

	by **Call::StartAudioMonitor**.
void **StartConnectionMonitor**(int *cms*)	Starts sending connection information through **SignalAudioMonitor**. This is called by **Call::StartConnectionMonitor**.
void **StopAudioMonitor**()	Stops sending audio information through **SignalAudioMonitor**. This is called by **Call::StopAudioMonitor**.
void **StopConnectionMonitor**()	Stops sending connection information through **SignalAudioMonitor**. This is called by **Call::StopConnectionMonitor**.
void **UnpauseMedia_w**()	Called by **ChannelManager** to restart sending audio data stopped by calling **PauseMedia_w**.
VoiceChannel(ChannelManager **manager*, Session **session*, MediaChannel **channel*)	Constructor. Creates a **TransportChannel**object using the **Session** object passed in.
~VoiceChannel	Destructor. Destroys the **TransportChannel**object that it created when it was instantiated.
Thread* **worker_thread**()	Returns the worker thread used to handle resource-intensive applications (used to send signals to this object's **OnMessage** method).

Signals

SignalConnectionMonitor < VoiceChannel *, const std::vector< ConnectionInfo > & >

Sends information about the current connection. To start this signaling, you must call **StartConnectionMonitor**.

SignalAudioMonitor < VoiceChannel *, const AudioInfo & >

Sends information about the current audio data. To start this signaling, you must call **StartAudioMonitor**.

> **Attributes:** public

> **Declaration file:** talk/session/phone/voicechannel.h

Last updated March 23, 2012.

Search ●

Explore

- Products
- Showcase
- Events
- Communities

Connect

- Blog
- Google+ Community
- YouTube Channel
- Report an Issue
- Jobs

Programs

- Groups
- Experts
- Startups
- Women Techmakers

Top Products

- Ads
- Analytics
- Android
- Apps
- Cast
- Chrome
- Cloud
- Glass
- Google+
- Maps
- Wallet
- YouTube

Terms of Use | Privacy Policy

English