

# Audio/video track status detection: muted/unmuted

- [Receiving stream status after stream subscription](#)
- [Mixer incoming stream status detection while playing a mixed stream](#)

Since Android SDK build [1.0.1.81](#) it is possible to receive stream publisher events while playing the stream. This can be used to detect if audio or video is muted by stream publisher: when publisher uses `muteAudio()/muteVideo()` method, a special event is sending to all the subscribers. To receive this event while playing a stream, define the function `Stream.onStreamEvent()` and check a value returned by `StreamEvent.getType()` method:

[code](#)

```
@Override
public void onStreamEvent(StreamEvent streamEvent) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            switch (streamEvent.getType()) {
                case audioMuted: mAudioMuteStatus.setText(getString(R.string.audio_mute_status)+"true"); break;
                case audioUnmuted: mAudioMuteStatus.setText(getString(R.string.audio_mute_status)+"false");
break;
                case videoMuted: mVideoMuteStatus.setText(getString(R.string.video_mute_status)+"true"); break;
                case videoUnmuted: mVideoMuteStatus.setText(getString(R.string.video_mute_status)+"false");
            }
        }
    });
}
```

## Receiving stream status after stream subscription

Since Android SDK build [1.0.1.85](#) it is possible to receive a stream status when a subscriber connects to this stream to play `inStreamStatusPlaying` event handler, using `Stream.getAudioState()` and `Stream.getVideoState()` methods

[code](#)

```
@Override
public void onStreamStatus(final Stream stream, final StreamStatus streamStatus) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            if (!StreamStatus.PLAYING.equals(streamStatus)) {
                ...
            } else {
                onPlayed(stream);
                ...
            }
        }
    });
}
...
private void onPlayed(Stream stream) {
    mPlayButton.setText(R.string.action_stop_play);
    mPlayButton.setTag(R.string.action_stop_play);
    mPlayButton.setEnabled(true);

    mAudioMuteStatus.setText(getString(R.string.audio_mute_status) + String.valueOf(stream.getAudioState().
isMuted()));
    mVideoMuteStatus.setText(getString(R.string.video_mute_status) + String.valueOf(stream.getVideoState().
isMuted()));
}
```

## Mixer incoming stream status detection while playing a mixed stream

Since Android SDK build [1.0.1.82](#) it is possible to detect mixer incoming stream status while playing a mixed stream. In this case, `Stream.onStreamEvent()` should be defined, in which `StreamEvent.payload` should be checked with a corresponding method. Then, if payload is not empty, the name of the muted/unmuted stream should be extracted

[code](#)

```
        @Override
        public void onStreamEvent(StreamEvent streamEvent) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    if (streamEvent.getPayload() != null) {
                        mMutedName.setText(getString(R.string.muted_name) + streamEvent.getPayload().
getStreamName());
                    }
                    switch (streamEvent.getType()) {
                        case audioMuted: mAudioMuteStatus.setText(getString(R.string.audio_mute_status)
+"true"); break;
                        case audioUnmuted: mAudioMuteStatus.setText(getString(R.string.
audio_mute_status)+"false"); break;
                        case videoMuted: mVideoMuteStatus.setText(getString(R.string.video_mute_status)
+"true"); break;
                        case videoUnmuted: mVideoMuteStatus.setText(getString(R.string.
video_mute_status)+"false");
                    }
                }
            });
        }
    }
```