

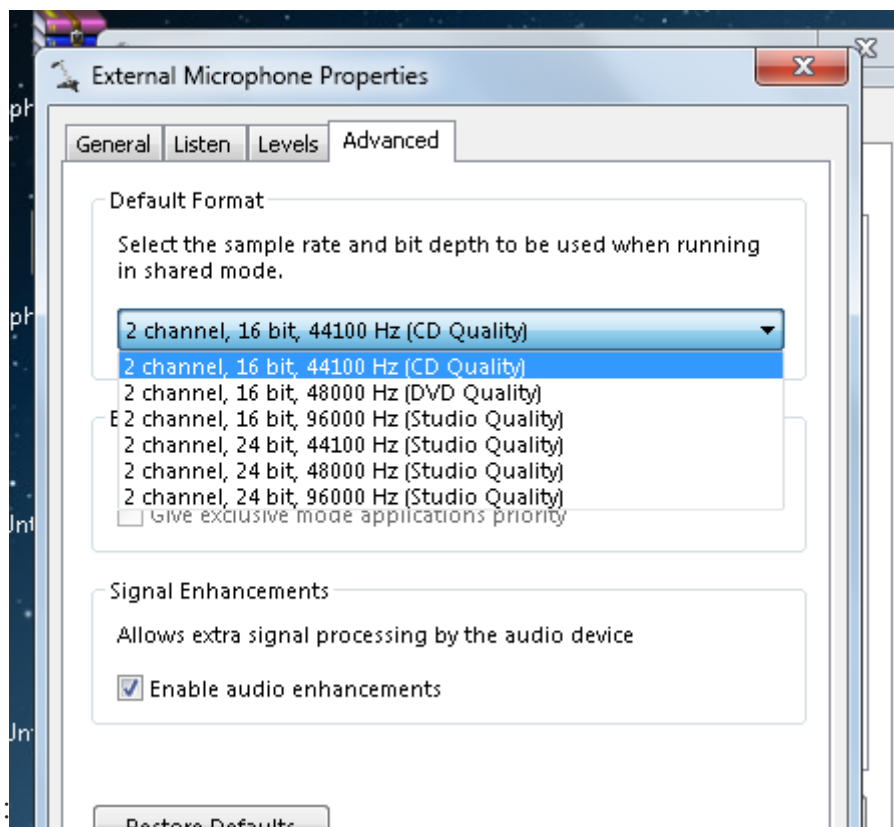
How are we doing? Please help us improve Stack Overflow. [Take our short survey](#)

Can't access microphone while running Dialog demo in sphinx4 5prealpha

Asked 7 years, 10 months ago Modified 2 years, 10 months ago Viewed 4k times

- ▲ I am trying to run the dialog demo of sphinx 4 pre alpha but it gives errors.
- 8 I am creating a live speech application.
- ▼ I imported the project using maven and followed this guide on stack overflow:
<https://stackoverflow.com/a/25963020/2653162>
- 🔖
- 🕒 The error says about issues regarding the 16 khz and channel being mono. So clearly its about the sampling stuff. And is also says about microphone.

I looked on how change the microphone settings to 16 khz and 16 bit but there is no such option in windows 7



The thing is that the HelloWorld and dialog demo worked fine in sphinx4 1.06 beta but after I tried the latest release it gives following errors:

```
Exception in thread "main" java.lang.IllegalStateException:
javax.sound.sampled.LineUnavailableException: line with format PCM_SIGNED 16000.0 Hz,
16 bit, mono, 2 bytes/frame, little-endian not supported.
    at edu.cmu.sphinx.api.Microphone.<init>(Microphone.java:38)
    at
    edu.cmu.sphinx.api.SpeechSourceProvider.getMicrophone(SpeechSourceProvider.java:18)
```

```

    at edu.cmu.sphinx.api.LiveSpeechRecognizer.<init>(LiveSpeechRecognizer.java:34)
    at edu.cmu.sphinx.demo.dialog.Dialog.main(Dialog.java:145)
Caused by: javax.sound.sampled.LineUnavailableException: line with format PCM_SIGNED
16000.0 Hz, 16 bit, mono, 2 bytes/frame, little-endian not supported.
    at
com.sun.media.sound.DirectAudioDevice$DirectDL.implOpen(DirectAudioDevice.java:513)
    at com.sun.media.sound.AbstractDataLine.open(AbstractDataLine.java:121)
    at com.sun.media.sound.AbstractDataLine.open(AbstractDataLine.java:413)
    at edu.cmu.sphinx.api.Microphone.<init>(Microphone.java:36)
    ... 3 more

```

Cant figure out what to do to resolve the issue.

java cmusphinx sphinx4

Share Improve this question

edited May 23, 2017 at 12:17

asked Mar 18, 2015 at 11:45

Follow



Community Bot
1 1



abc
117 2 11

@NikolayShmyrev sound card is Conexant SmartAudio HD – abc Mar 18, 2015 at 16:23

Ok, I'm sorry, this is a bug in sphinx4 that recognizer doesn't release the resource properly and Windows java doesn't allow to open the microphone the second time. – Nikolay Shmyrev Mar 18, 2015 at 19:46

Relevant issue in our tracker sourceforge.net/p/cmusphinx/bugs/412 – Nikolay Shmyrev Mar 18, 2015 at 19:48

I'll try to fix it in a coming days – Nikolay Shmyrev Mar 18, 2015 at 19:53

@NikolayShmyrev The demo did not work even once – abc Mar 19, 2015 at 4:58

Sorted by:

4 Answers

Highest score (default)



5



If you modify `SpeechSourceProvider` to return a constant microphone reference, it won't try to create multiple microphone references, which is the source of the issue.

```

public class SpeechSourceProvider {
    private static final Microphone mic = new Microphone(16000, 16, true, false);

    Microphone getMicrophone() {
        return mic;
    }
}

```

The problem here is that you don't want multiple threads trying to access a single resource, but for the demo, the recognizers are stopped and started as needed so that they aren't all competing for the microphone.



-
- 1 This worked for me too. As of 2019 I just cloned the github repo and faced the same issue.
– [jotadepicas](#) May 20, 2019 at 0:36
-



1



As Nickolay explains in the source forge forum ([here](#)) the microphone resource needs to be released by the recognizer currently using it for another recognizer to be able to use the microphone. While the API is being fixed, I made the following changes to certain classes in the sphinx API as a temporary workaround. This is probably not the best solution, guess until a better solution is proposed, this will work.

I created a class named `MicrophoneExtention` with the same source code as the `Microphone` class, and added the following methods:

```
public void closeLine(){
    line.close();
}
```

Similarly a `LiveSpeechRecognizerExtention` class with the source code of `LiveSpeechRecognizer` class, and made the following changes:

- use the `MicrohphoneExtention` class I defined:
`private final MicroPhoneExtention microphone;`
- inside the constructor,
`microphone =new MicrophoneExtention(16000, 16, true, false);`
- And add the following methods:

```
public void closeRecognitionLine(){
    microphone.closeLine();
}
```

Finally I edited the main method of the `DialogDemo` .

```
Configuration configuration = new Configuration();
configuration.setAcousticModelPath(ACOUSTIC_MODEL);
configuration.setDictionaryPath(DICTIONARY_PATH);
configuration.setGrammarPath(GRAMMAR_PATH);
configuration.setUseGrammar(true);
```

```
configuration.setGrammarName("dialog");
```

```

LiveSpeechRecognizerExtention recognizer =
new LiveSpeechRecognizerExtention(configuration);

Recognizer.startRecognition(true);
while (true) {
    System.out.println("Choose menu item:");
    System.out.println("Example: go to the bank account");
    System.out.println("Example: exit the program");
    System.out.println("Example: weather forecast");
    System.out.println("Example: digits\n");

    String utterance = recognizer.getResult().getHypothesis();

    if (utterance.startsWith("exit"))
        break;

    if (utterance.equals("digits")) {
        recognizer.stopRecognition();
        recognizer.closeRecognitionLine();
        configuration.setGrammarName("digits.grxml");
        recognizer=new LiveSpeechRecognizerExtention(configuration);
        recognizeDigits(recognizer);
        recognizer.closeRecognitionLine();
        configuration.setGrammarName("dialog");
        recognizer=new LiveSpeechRecognizerExtention(configuration);
        recognizer.startRecognition(true);
    }

    if (utterance.equals("bank account")) {
        recognizer.stopRecognition();
        recognizerBankAccount(Recognizer);
        recognizer.startRecognition(true);
    }

    if (utterance.endsWith("weather forecast")) {
        recognizer.stopRecognition();
        recognizer.closeRecognitionLine();
        configuration.setUseGrammar(false);
        configuration.setLanguageModelPath(LANGUAGE_MODEL);
        recognizer=new LiveSpeechRecognizerExtention(configuration);
        recognizeWeather(recognizer);
        recognizer.closeRecognitionLine();
        configuration.setUseGrammar(true);
        configuration.setGrammarName("dialog");
        recognizer=new LiveSpeechRecognizerExtention(configuration);
        recognizer.startRecognition(true);
    }
}

Recognizer.stopRecognition();

```

and obviously the method signatures in the `DialogDemo` needs changing... hope this helps...
and on a final note, I am not sure if what I did is exactly legal to start with. If i am doing something wrong, please be kind enough to point out my mistakes :D

Share Improve this answer Follow

answered Oct 17, 2015 at 17:06



Ahmed Shariff

763 6 12

I followed your modifications, when I run now I am getting this error: "java.lang.IllegalStateException: Expected state READY actual state DEALLOCATED at


edu.cmu.sphinx.recognizer.Recognizer.checkState(Recognizer.java:134)" – [Vinod](#) Oct 30, 2015 at 17:29

I am not sure where the problem could have occurred, mine works just fine, what is on the call stack? If i am reading this right, that exception gets thrown by the recognizer's decoder, not the line.

– [Ahmed Shariff](#) Oct 31, 2015 at 7:38 

Exception in thread "main" java.lang.IllegalStateException: Expected state READY actual state DEALLOCATED at edu.cmu.sphinx.recognizer.Recognizer.checkState(Recognizer.java:134) at edu.cmu.sphinx.recognizer.Recognizer.recognize(Recognizer.java:103) at edu.cmu.sphinx.recognizer.Recognizer.recognize(Recognizer.java:122) at edu.cmu.sphinx.api.AbstractSpeechRecognizer.getResult(AbstractSpeechRecognizer.java:63) at edu.cmu.sphinx.demo.dialog.DialogDemo.main(DialogDemo.java:153) – [Vinod](#) Oct 31, 2015 at 14:32

this is the console output I got – [Vinod](#) Oct 31, 2015 at 14:33

Make sure the order of the statements are correct. 1.stop recognition 2.close line 3.reassign recognizer 4.start recognition. as i said, this works for me just fine.. – [Ahmed Shariff](#) Oct 31, 2015 at 16:19 



0



The answer of aetherwalker worked for me - in more detail I overwrote the following files with my own implementations where I only changed the used SpeechSourceProvider:

First one is the AbstractSpeechRecognizer:

```
public class MaxAbstractSpeechRecognizer {
    protected final Context context;
    protected final Recognizer recognizer;

    protected ClusteredDensityFileData clusters;

    protected final MaxSpeechSourceProvider speechSourceProvider;

    /**
     * Constructs recognizer object using provided configuration.
     * @param configuration initial configuration
     * @throws IOException if IO went wrong
     */
    public MaxAbstractSpeechRecognizer(Configuration configuration)
        throws IOException
    {
        this(new Context(configuration));
    }

    protected MaxAbstractSpeechRecognizer(Context context) throws IOException {
        this.context = context;
        recognizer = context.getInstance(Recognizer.class);
        speechSourceProvider = new MaxSpeechSourceProvider();
    } .....
```

Then the LiveSpeechRecognizer:

```
public class MaxLiveSpeechRecognizer extends MaxAbstractSpeechRecognizer {

    private final Microphone microphone;

    /**
     * Constructs new live recognition object.
     *
     * @param configuration common configuration
```

```

* @throws IOException if model IO went wrong
*/
public MaxLiveSpeechRecognizer(Configuration configuration) throws IOException
{
    super(configuration);
    microphone = speechSourceProvider.getMicrophone();
    context.getInstance(StreamDataSource.class)
        .setInputStream(microphone.getStream());
}.....

```

And last but not least the SpeechSourceProvider:

```

import edu.cmu.sphinx.api.Microphone;

public class MaxSpeechSourceProvider {

    private static final Microphone mic = new Microphone(16000, 16, true, false);

    Microphone getMicrophone() {
        return mic;
    }
}

```

Share Improve this answer Follow

answered Mar 30, 2016 at 19:00



max

1,016 9 14



0



For me with this change, the problem was as long as i was staying in a context of cmusphinx it was good the line can be reused many times. But if i begin to reuse the mic for another work (like recording) it was not available!

I see that the stream was open in Microphone class but never close!



So first I change in class `Microphone` the following attributes from static to dynamic :



```

private TargetDataLine line;
private InputStream inputStream;

```

After i change the method `stopRecording` for closing stream before line:

```

/**
 * close the stream and line
 */
public void stopRecording() {

    if (inputStream != null )
        try {
            inputStream.close();
        } catch (IOException e) {
            throw new IllegalStateException(e);
        }

    line.stop();

}

```

And now with no more change (class `SpeechSourceProvider` is original), i can reuse alternatively mic for cmupsphinx and another recording task

Share Improve this answer Follow

edited Feb 20, 2020 at 8:51

answered Feb 19, 2020 at 9:04



thierry44

31 1 3