

# C# Speech Recognition - Is this what the user said?

Asked 16 years, 2 months ago    Modified 4 years, 7 months ago

Viewed 71k times



55



I have need to write an application which uses a speech recognition engine -- either the built in vista one, or a third party one -- that can display a word or phrase, and recognise when the user reads it (or an approximation of it). I also need to be able to switch quickly between languages, without changing the language of the operating system.

The users will be using the system for very short periods. The application needs to work without the requirement of first training the recognition engine to the users' voices.

It would also be fantastic if this could work on Windows XP or lesser versions of Windows Vista.

Optionally, the system needs to be able to read information on the screen back to the user, in the user's selected language. I can work around this specification using pre-recorded voice-overs, but the preferred method would be to use a text-to-speech engine.

Can anyone recommend something for me?

c#

speech-recognition

text-to-speech

Share

Improve this question

Follow

asked Oct 22, 2008 at 19:04



Richard C

2,216 ● 5 ● 31 ● 40

Please clarify... What do you mean? Do you mean a recognition engine? An structure for the application? If you should even attempt to do it? – [Jorge Córdoba](#) Oct 22, 2008 at 19:06

- 2 I am mainly looking for an engine to use. I need to be able to tell my managers whether or not the idea is feasible. I already have a rough idea of how to structure the application around the engine, all I need is to plug the engine in. – [Richard C](#) Oct 22, 2008 at 19:15

## 11 Answers

Sorted by:

Highest score (default)



59



A similar question was asked on Joel on Software a while back. You can use the [System.Speech.Recognition](#) namespace to do this...with some limitations. Add System.Speech (should be in the GAC) to your project. Here's some sample code for a WinForms app:



```
public partial class Form1 : Form
{
    SpeechRecognizer rec = new SpeechRecognizer();

    public Form1()
    {
        InitializeComponent();
    }
}
```

```

        rec.SpeechRecognized += rec_SpeechRecognized;
    }

    void rec_SpeechRecognized(object sender, SpeechRecogn
    {
        lblLetter.Text = e.Result.Text;
    }

    void Form1_Load(object sender, EventArgs e)
    {
        var c = new Choices();
        for (var i = 0; i <= 100; i++)
            c.Add(i.ToString());
        var gb = new GrammarBuilder(c);
        var g = new Grammar(gb);
        rec.LoadGrammar(g);
        rec.Enabled = true;
    }

```

This recognizes the numbers from 1 to 100, and displays the resulting number on the form. You'll need a form with a label called lblLetter on it.

System.Speech only works with a pre-defined list of words or phrases; it's not exactly NaturallySpeaking, either in versatility or in recognition quality. But you don't have to train it to the user's voice, and if you only have a few different things the user can say, it works reasonably well. And it's free! (if you have Visual Studio)

It won't work well if you use very short phrases; I made a program for my kid to say letters of the alphabet and see them on-screen, but it doesn't do that well since many of the letters sound alike (especially from the mouth of a four-year-old).

As for more flexible options...well, there's the aforementioned NaturallySpeaking, which has an SDK. But you have to contact sales to get any sort of access to it, and no pricing is listed, so it comes across as one of those "How much does it cost? Well, how much have you got?" kind of things. There doesn't seem to be a "download and play around with it" option. :(

As for text-to-speech, [System.Speech.Synthesis](#) does this. It's even easier than the speech recognition. I wrote a small program to let me type, hit Enter, and read the text aloud. My four-year-old gets mesmerized by it. :) ("Daddy, I wanna tawk to da wobot.")

Share Improve this answer

answered Oct 22, 2008 at 19:31

Follow




Ryan Lundy

210k ● 41 ● 184 ● 216

---

How would I adapt this code to recognise 1 - 100 in French or German without needing to change the OS display language? – [Richard C](#) Oct 23, 2008 at 7:07

- 
- 1 The only language possible is the one of your OS. I just read it from MSDN. – [Patrick Desjardins](#) Mar 17, 2009 at 13:18
- 
- 4 I think your comment that "only works with a pre-defined list of words or phrases" is not true. The desktop recognizer in Vista and later includes a dictation grammar that you can load. See [msdn.microsoft.com/en-us/library/...](http://msdn.microsoft.com/en-us/library/...) – [Michael Levy](#) Oct 18, 2010 at 19:59 
- 
- 1 +1 for horrible sales from Nuance / Dragon. We had to practically beg to get them to sell us their software (for a very non-trivial price). – [Matt Jordan](#) Jan 14, 2011 at 22:27
-

Hey... I tried to use this code... In my form I have only the label (is it correct?), I run the application, the speech recognizer start working, I say for example "four", and in the speech recognizer program always appears "what was that?" and nothing happens to the label. I even put a break point in Form1\_Load and it never gets there... any suggestions?  
thanks :) – [user990635](#) Apr 29, 2012 at 18:50

---



[Note: I was the development lead for the managed speech recognition API in .NET 3.0]

**16**



System.Speech is part of .NET 3.0, so it is available on both Vista and XP. In Vista you have the added benefit of having a speech recognition engine pre-installed by the OS. On XP you choices are: use the SAPI 5.1 SDK with a very old engine (but might work well enough for your command and control scenario), install Office 2003 which installs a newer version of the recognizer. There are a few SAPI 5 compliant speech recognition engines available as well.



If you need to switch languages, you will want to use the `System.Speech.Recognition.SpeechRecognitionEngine` class which allows you to choose the SR engine for the language you need to support. Note that engines are defined by a set of languages they support (they might be using the same binary, only swapping data files to support additional languages).

Comment if you need to know more.

# Philipp

Share Improve this answer

answered Nov 4, 2008 at 21:35

Follow



**Philipp Schmid**

5,828 ● 5 ● 46 ● 66

---

Phillip, if I want to use the engine and train it to learn and recognize spoken Croatian, as a way for transcribing various speakers, is it possible, and if is, where to start?

– [Daniel Mošmondor](#) Oct 23, 2010 at 21:43

---

There are 2 parts to a speech recognizer: acoustic models and language model. You can use the Vista Dictation Resource Kit (or something like that) to build a dictation language model that references Croatian words. There are currently no tools to train the acoustic models which you would want to do if there are sounds in Croatian that are not present in English (or whatever existing SR language you are using). You can specify custom pronunciations for your Croatian words to improve your recognition accuracy.

– [Philipp Schmid](#) Oct 29, 2010 at 16:17

---

Phillip, I have been playing with the libraries you suggested, but I have had very hard time getting it to recognize what I said. I do have a little accent, but I have not had this experience in other devices, like kinect and so on. Should I just consider that my mic is not good enough, or am I making something wrong? So you have any suggestions? Ps. I am using the example on the MSDN documentation. – [Oakcool](#) May 3, 2013 at 1:00

---

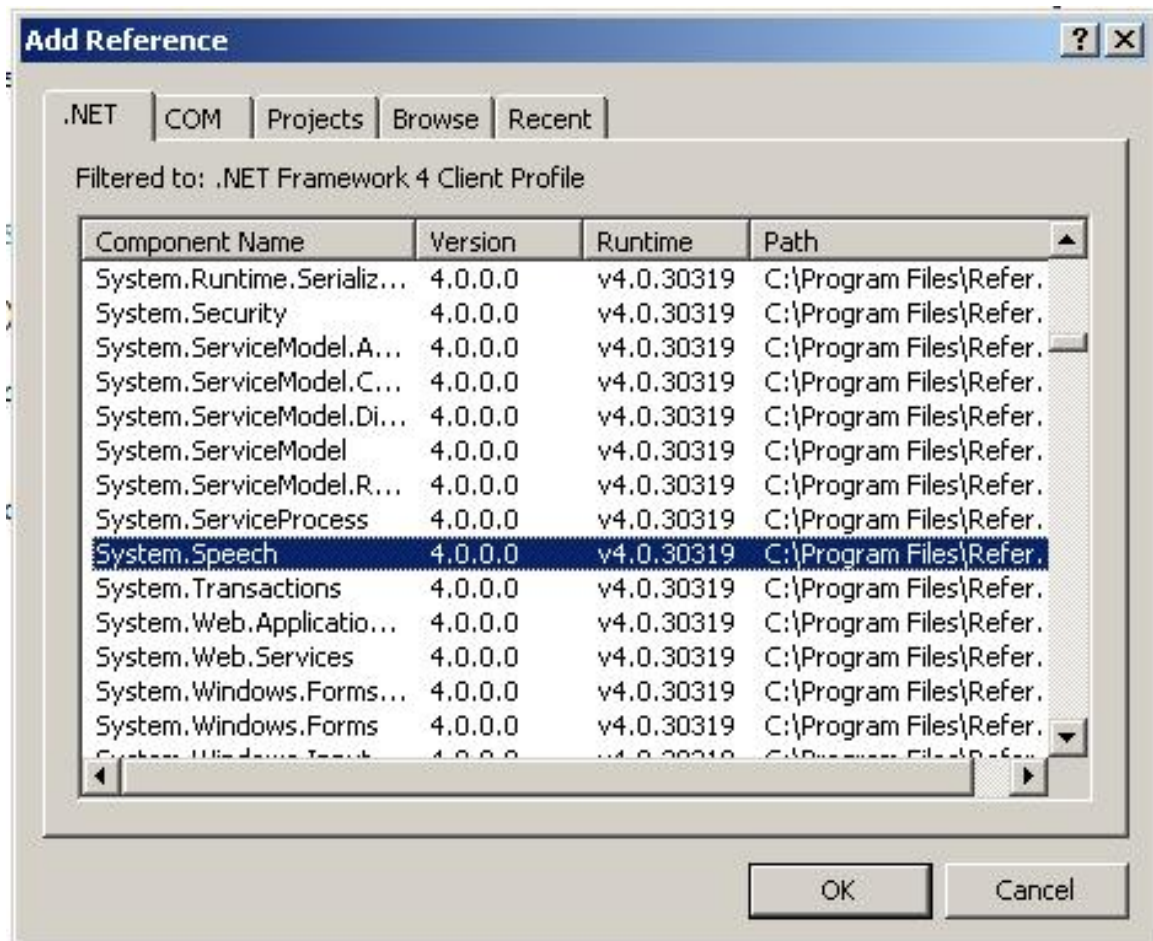
@Oakcool Are you using grammar-based recognition? That's what Xbox with Kinect is using. It will be much more accurate (but also more restrictive) than dictation. – [Philipp Schmid](#) May 3, 2013 at 16:46

---

@Philipp this is the example I used [msdn.microsoft.com/en-us/library/vstudio/...](https://msdn.microsoft.com/en-us/library/vstudio/...), I don't think its grammar based. What I

meant by using kinect, is that my accent did not matter when I was using kinect, on my XBox, not that I was programming against it. – [Oakcool](#) May 3, 2013 at 21:30

Before this add 'Speech' reference



Found that the code example posted by Kyralessa on Oct 22nd didn't work for me but a slightly revised version did. When adding strings into the Choices object use full text English words not numbers. Seems the MS speech recognition engine can't recognize numbers by themselves.

I have marked these modifications with some commenting added to the previous example.



```

public partial class Form1 : Form
{
    SpeechRecognizer rec = new SpeechRecognizer();

    public Form1()
    {
        InitializeComponent();
        rec.SpeechRecognized += rec_SpeechRecognized;
    }

    void rec_SpeechRecognized(object sender, SpeechRecogn
    {
        lblLetter.Text = e.Result.Text;
    }

    void Form1_Load(object sender, EventArgs e)
    {
        var c = new Choices();

        // Doesn't work must use English words to add to C
        // populate grammar.
        //
        //for (var i = 0; i <= 100; i++)
        //    c.Add(i.ToString());

        c.Add("one");
        c.Add("two");
        c.Add("three");
        c.Add("four");
        // etc...

        var gb = new GrammarBuilder(c);
        var g = new Grammar(gb);
        rec.LoadGrammar(g);
        rec.Enabled = true;
    }
}

```

Share Improve this answer

edited Apr 25, 2011 at 7:17

Follow



Community Bot

1 • 1



answered Nov 25, 2008 at 5:32



**Rob Segal**

7,615 ● 12 ● 45 ● 73

---

Not sure why it didn't work for you; the code I posted came directly from a program I wrote and used, and it worked for me. Perhaps it's related to the culture settings on your system? – [Ryan Lundy](#) Nov 19, 2009 at 14:25

---

Could be. I didn't look into this extremely in depth.  
– [Rob Segal](#) Dec 8, 2009 at 20:09

---



7



If the engine is what you're asking about then I've found (beware, I'm just listing, I haven't tried any of them):

[Lumenvox engine](#)

you also have the [SAPI SDK](#) from Microsoft itself, I've only tried it for text to speech but according to its definition:

The SDK also includes freely distributable text-to-speech (TTS) engines (in U.S. English and Simplified Chinese) **and speech recognition (SR) engines** (in U.S. English, Simplified Chinese, and Japanese).

Share Improve this answer

Follow

answered Oct 22, 2008 at 19:14



**Jorge Córdoba**

52.1k ● 11 ● 82 ● 130

---

1 The Lumenvox engine looks like it might do the trick! I'm going to have to play a bit with it to be certain. Also need to

discuss pricing with the managers. Thanks Jorge!

– [Richard C](#) Oct 22, 2008 at 19:20

---



3



Be warned that you're not going to get good results if you don't require training first. Speech recognition is a statistical application of phonetics, a field which is pretty frank about the fact that there's so much variation in the signal that it's almost a miracle anyone can understand what anyone else says. An off-the-shelf speech recognition engine will most likely tend towards a more general accent of English, but will fail miserably for anything even slightly different.

That's why training is so important. We can do well by overfitting with ease, especially if we reduce the problem space. But creating an extensible machine learning solution? Therein always lies the rub.

That being says, consider Sphinx-4. It's an off-the-shelf solution written in Java available at

<http://cmusphinx.sourceforge.net/sphinx4/>

Share Improve this answer

answered Oct 22, 2008 at 19:23

Follow



[Robert Elwell](#)

6,668 ● 1 ● 30 ● 32

---

2 +1 for the warning about not learning, am just wishing there was a .NET port for Sphinx-4. – [JWL](#) Jul 22, 2011 at 11:57

---



1



Check out the new Speech class libraries in .NET 3.5

<http://msdn.microsoft.com/en-us/library/system.speech.recognition.speechrecognizer.aspx>

general documentation for SR and TTS

<http://msdn.microsoft.com/en-us/library/system.speech.recognition.aspx>  
<http://msdn.microsoft.com/en-us/library/system.speech.synthesis.aspx>

Share Improve this answer

answered Oct 22, 2008 at 19:21

Follow



**stephbu**

5,082 ● 28 ● 42

---



0



Text to speech is available with the [Speech API](#). Personally, I'd probably require Vista and use the managed interfaces to [System.Speech.SpeechRecognition](#) and [System.Speech.Synthesis.TtsEngine](#), but a P/Invoke should be possible into the unmanaged APIs if you really need XP support.

Share Improve this answer

answered Oct 22, 2008 at 19:22

Follow



**Mark Brackett**

85.6k ● 17 ● 111 ● 155



0



Try [Microsoft Speech Server](#), which I think now is part of [Office Communication Server 2007](#). It contains a SR/TTS engines, C# API and tools that integrate with Visual Studio.

Share Improve this answer

answered Nov 4, 2008 at 21:41

Follow



**dbkk**

12.8k ● 13 ● 54 ● 60



0



This is the article from MSDN magazine that first discussed using the System.Speech APIs for Vista. Some of it is out of date because the API changed between beta (when the article was written) and the release of Vista, but this is still one of the best resources I've found and covers a good intro to the System.Speech namespace. See <http://msdn.microsoft.com/en-us/magazine/cc163663.aspx>



Share Improve this answer

answered Jun 9, 2010 at 18:29

Follow



[Michael Levy](#)

13.3k ● 15 ● 67 ● 101



0

[Dragon Naturally Speaking SDK](#) might be worth looking at. [This project](#) looked interesting.

Haven't got to play with either of them though.



Share Improve this answer

edited Apr 11, 2012 at 19:46



Follow



answered Oct 22, 2008 at 19:17



[itsmatt](#)

31.4k ● 11 ● 102 ● 165

Dead link and the "This Project" is about text to speech, not speech recognition – [Levitikon](#) Apr 11, 2012 at 19:19

@Levitikon Did you read the latter part of the question before you down-voted me? He was looking for both a speech recognition piece **and** text-to-speech. No surprise here that a 3+ year old link is dead. I updated it. – [itsmatt](#) Apr 11, 2012 at 19:45



0

Well, this question already has many good responses but I think it is valuable to update with some info from 2016 documentation the responses from Rob Segal and Philipp Schmid pointing to this nice code example:



<https://msdn.microsoft.com/en-us/library/office/system.speech.recognition.speechrecognitionengine.aspx>



It did not use the shared recognizer of Windows (The little Windows Mic that shows out up in the middle of the screen), it use a nice in app SpeechRecognitionEngine that not need any visual cue. The UI is completly at your control.

Share Improve this answer

answered Jul 20, 2016 at 0:44

Follow



Fidel Orozco

1,046 ● 12 ● 20

---