



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/709,126	12/10/2019	Ross Pirtle	7552/85663-Reissue-A	1509
23432	7590	08/23/2021		
COOPER & DUNHAM LLP 90 PARK AVENUE, 21 ST. FLOOR NEW YORK, NY 10016			EXAMINER ENGLISH, PETER C	
			ART UNIT	PAPER NUMBER
			3993	
			NOTIFICATION DATE	DELIVERY MODE
			08/23/2021	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

DUSPTO@cooperdunham.com
patcomm-in-grp@cooperdunham.com
pdocketing@cooperdunham.com

<i>Applicant-Initiated Interview Summary</i>	Application No. 16/709,126	Applicant(s) Pirtle, Ross		
	Examiner Peter C English	Art Unit 3993	AIA (First Inventor to File) Status Yes	Page 1 of 2

All Participants (applicant, applicants representative, PTO personnel)	Title	Type
Peter C English	Primary Examiner	Telephonic
Gay Spahn	SPE	
Glenn Dawson	Primary Examiner	
Paul Teng	Attorney of Record	

Date of Interview: 18 August 2021

Issues Discussed:

35 U.S.C. 112

See the attached interview agenda.

Discussed the examiner's findings of new matter (items 5-15 on pp. 2-8 of the prior Office action) and the corresponding rejections under 35 USC 251 (GROUND 1 on p 9 of the prior Office action) and 35 USC 112(a) (GROUND 2 on p. 10 of the prior Office action). Applicant's representative proposed amending lines 4-5 of claim 20 to eliminate the recitation of "one or more programs" and instead recite "a memory device storing executable instructions, wherein execution of the executable instructions..." The examiner agreed that this would constitute an improvement over the original claim language. However, the examiner explained the need in the applicant's response to explain how the original disclosure provides support for the specific algorithm covered by the claims.

Also discussed the rejection of claims 23, 24, 26 and 36 under 35 USC 112(a) for lack of written description and/or enablement (GROUND 4 on pp. 11-12 of the prior Office action). Applicant's representative argued there would be no need for a person of ordinary skill in the art to write specific computer programming for the claimed receipt, extraction and transmission of a secret code via voice message; rather, one would chose from an available library of software. Applicant's representative also indicated the intention to file a declaration under 37 CFR 1.132 attesting to these facts. The examiner noted that it would be helpful to point to disclosure in the original specification of such a selection from an available library of software. The examiner also explained the need to disclose the manner in which the claimed functions are performed, i.e., how the applicant's invention achieves the intend results, as opposed to requiring a person of ordinary skill in the art to devise their own solution.

Objections

See above explanation with respect to the examiner's findings of new matter.

Other

See above explanation with respect to the rejection under 35 USC 251.

<i>Applicant-Initiated Interview Summary</i>	Application No. 16/709,126	Applicant(s) Pirtle, Ross		
	Examiner Peter C English	Art Unit 3993	AIA (First Inventor to File) Status Yes	Page 2 of 2

/PETER C ENGLISH/ Primary Examiner, Art Unit 3993	Conferees: /GKD/ and /GAS/
Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04 Please further see: MPEP 713.04 Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b) 37 CFR § 1.2 Business to be transacted in writing	

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

Discuss the issues raised in the Office Action (OA) regarding the following terms:

- "... execution of the one or more programs causes the processor to ..." (OA, sect. 5-9, 11-12, 15);
- Enablement - voice message (OA, section 25)

English, Peter

From: Paul Teng <PTENG@COOPERDUNHAM.COM>
Sent: Monday, August 16, 2021 4:31 PM
To: English, Peter
Subject: RE: Reissue Application No. 16/709,126 [our Dkt. 7552/85663-Reissue-A] -- agenda for telephone interview scheduled for August 18, 10am

CAUTION: This email has originated from a source outside of USPTO. **PLEASE CONSIDER THE SOURCE** before responding, clicking on links, or opening attachments.

Examiner,

Understood. Thank you.

Paul Teng
COOPER & DUNHAM LLP
90 Park Avenue, 21st Floor
New York, NY 10016
Tel.: (212) 278-0400
Email: PTeng@cooperdunham.com
Mobile: (516) 424-5401

This message is intended solely for the use of the addressee and may contain confidential and/or attorney-client privileged information. If you are not the intended recipient, or an employee or agent responsible for delivering this message to the intended recipient, you are hereby notified that any review, distribution, or copying of this message is strictly prohibited. If you received this message in error, please notify us immediately. Thank you.

From: English, Peter <Peter.English@USPTO.GOV>
Sent: Monday, August 16, 2021 4:03 PM
To: Paul Teng <PTENG@COOPERDUNHAM.COM>
Subject: RE: Reissue Application No. 16/709,126 [our Dkt. 7552/85663-Reissue-A] -- agenda for telephone interview scheduled for August 18, 10am

Mr. Teng,

I've set up an audio conference for the interview.

Please follow these instructions when connecting:

Dial **571-273-0000**

When prompted, enter the:

Meeting Number followed by the pound key: **26671#**

Attendee Access Code followed by the pound key: **46671#**

The other USPTO participants will be Gay Spahn (Supervisory Reexamination Specialist) and Glenn Dawson (Reexamination Specialist).

The interview will be limited to 30 minutes per MPEP 713.01.

Peter English
USPTO
Art Unit 3993

From: Paul Teng <PTENG@COOPERDUNHAM.COM>
Sent: Monday, August 16, 2021 1:02 PM
To: English, Peter <Peter.English@USPTO.GOV>
Subject: Reissue Application No. 16/709,126 [our Dkt. 7552/85663-Reissue-A] -- agenda for telephone interview scheduled for August 18, 10am

CAUTION: This email has originated from a source outside of USPTO. **PLEASE CONSIDER THE SOURCE** before responding, clicking on links, or opening attachments.

Examiner,

The date and time (Weds., August 18, 10am) proposed in the message you left on my voicemail works fine for me.

Please find attached an agenda for the telephone interview.

Unless you tell me otherwise, I will call from you at 571-272-6671 on August 18, 10am.

Thank you.

Paul Teng
COOPER & DUNHAM LLP
90 Park Avenue, 21st Floor
New York, NY 10016
Tel.: (212) 278-0400
Email: PTeng@cooperdunham.com
Mobile: (516) 424-5401

This message is intended solely for the use of the addressee and may contain confidential and/or attorney-client privileged information. If you are not the intended recipient, or an employee or agent responsible for delivering this message to the intended recipient, you are hereby notified that any review, distribution, or copying of this message is strictly prohibited. If you received this message in error, please notify us immediately. Thank you.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor: Ross Pirtle

Applicant: RAILWARE, INC.

Reissue Application No.: 16/709,126 filed December 10, 2019 Art Unit: 3993

Original Patent No.: 9,403,545 granted on August 2, 2016 Examiner: Peter English

For: TOOLS FOR RAILWAY TRAFFIC CONTROL

Confirmation No.: 1509

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. § 1.132 OF ROSS PIRTLE

Sir:

I, Ross Pirtle, hereby declare as follows:

1. I am the inventor of the invention for which reissue of a patent is sought in the above-identified United States Reissue Patent Application. I have been active in the computer software or programming profession for over thirty years.
2. It is my understanding that each of claims 23, 24, 26 and 36 of the above-identified Reissue Patent Application has been rejected by the United States Patent and Trademark Office (USPTO) for the reason that the specification of the patent for which reissue is sought allegedly does not adequately describe components of, and an algorithm for, a voice interface part.
3. Based on my experience, a person skilled in the art looking to implement the claimed railway control apparatus would not have needed to engage in undue experimentation to implement an algorithm for a voice or speech interface because numerous ready-made voice or speech interface code were available at the time of filing via a DLL (Dynamic Link Library) or

library of routines.

4. For example, as shown in the attached document entitled "C# -Voice/Speech to text -Stack Overflow" (January 2011), the mindset of a programmer when developing or writing code for a software application more than ten years ago was to seek existing code via an API (application programming interface) or library when there is a need for converting voice/speech into text (and for converting text to speech).
5. Further, the attached document entitled ".NET – What is the difference between System.Speech.Recognition and Microsoft.Speech.Recognition?" (June 2010) shows that such DLLs were available and employed both for server code development as well as client code development.
6. In addition, the attached document entitled "Speech Input API for Android" (March 2010) shows that such approach was employed for non-Microsoft platforms as well.
7. Thus, persons skilled in the art, such as application developers, were aware of, and routinely resorted to, DLLs and libraries of routines, via an SDK (software development kit) or otherwise, for implementing the disclosed voice interface part regardless of the platform for which the application was being developed.

I hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under § 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: October 8, 2021

/Ross Pirtle/
Ross Pirtle

Voice/Speech to text [closed]

Asked 10 years, 8 months ago Active 5 years, 1 month ago Viewed 45k times

18

Closed. This question does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.

Want to improve this question? Update the question so it's [on-topic](#) for Stack Overflow.
Closed 5 years ago.

[Improve this question](#)

I need an API or library (preferably free) that will convert voice/speech through a microphone, into text (string).

Additionally, I will need an API or library that can do text-to-speech.

I'd like to use C# and .NET, but other languages will suffice.

Thanks.

c# .net speech-recognition voice-recognition speech-to-text

Share Follow

asked Jan 13 '11 at 6:40

 [Charles_Har](#)
255 1 4 6

6 Answers

Active Oldest Votes

You can use CMU Sphinx as it is pretty open and scalable solution and I think it can be used at both client and server side:

15

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

If you are looking for a Microsoft desktop solution then you can use SAPI:

<http://msdn.microsoft.com/en-us/magazine/cc163563.aspx>

On server side, you can use Microsoft Unified Communication, but do consider licencing as well:

<http://www.microsoft.com/uc/en/gb/default.aspx>

Update:

This thread has also some good reference:

[C# Speech Recognition - Is this what the user said?](#)

Share Follow

edited May 23 '17 at 11:56

 [Community Bot](#)
1 1

answered Jan 13 '11 at 7:28

 [ShahidAzim](#)
1,326 1 9 15

Just updated answer with one more link. – [ShahidAzim](#) Jan 13 '11 at 7:37

1 You don't need UCS if you only need speech recognition on Windows Server. You can download the free Microsoft Speech Platform - microsoft.com/downloads/en/ – [Michael Levy](#) Jan 14 '11 at 21:42

I didn't know about that, thanks for your post and it looks interesting. – [ShahidAzim](#) Jan 15 '11 at 16:51

Here is a complete example using C# and System Speech for converting from speech to text.

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)

[Sign up with Google](#)

[Sign up with GitHub](#)

[Sign up with Facebook](#)

[X](#)

9/25/21, 11:38 AM

c# - Voice/Speech to text - Slack Overflow

Step 1: Configuring the SpeechRecognitionEngine

```
_speechRecognitionEngine = new SpeechRecognitionEngine();
_speechRecognitionEngine.SetInputToDefaultAudioDevice();
_dictationGrammar = new DictationGrammar();
_speechRecognitionEngine.LoadGrammar(_dictationGrammar);
_speechRecognitionEngine.RecognizeAsync(RecognizeMode.Multiple);
```

At this point your object is ready to start transcribing audio from the microphone. You need to handle some events though, in order to actually get access to the results.

Step 2: Handling the SpeechRecognitionEngine Events

```
_speechRecognitionEngine.SpeechRecognized += new EventHandler(SpeechRecognized);
_speechRecognitionEngine.SpeechHypothesized += new EventHandler(SpeechHypothesizing);
```

```
_speechRecognitionEngine.SpeechRecognized += new EventHandler(SpeechRecognized);
_speechRecognitionEngine.SpeechHypothesized += new EventHandler(SpeechHypothesizing);
```

```
private void SpeechHypothesizing(object sender, SpeechHypothesizedEventArgs e) { ///real-time results from the engine string
realTimeResults = e.Result.Text; }
```

```
private void SpeechRecognized(object sender, SpeechRecognizedEventArgs e) { ///final answer from the engine string finalAnswer =
e.Result.Text; }
```

That's it. If you want to use a pre-recorded .wav file instead of a microphone, you would use

```
_speechRecognitionEngine.SetInputToWaveFile(pathToTargetWavFile);
```

instead of

```
_speechRecognitionEngine.SetInputToDefaultAudioDevice();
```

There are a bunch of different options in these classes and they are worth exploring in more detail.

<http://ellismis.com/2012/03/17/converting-or-transcribing-audio-to-text-using-c-and-net-system-speech/>

Share Follow

answered Mar 17 '12 at 12:13



bultionous

7,739 3 47 74

can we use SpeechRecognitionEngine for android applications using MONO Framework??? -- Always Tahir Jun 20 '12 at 7:47

Nope. Need some Visual Studio for that. -- bultionous Jun 21 '12 at 1:47

1 @bultionous incorrect. Mono is a Linux emulator for .NET, so yes, this will work with Mono. Also: Visual Studio is an IDE, not a programming language. -- Altkopher Apr 3 '17 at 16:25 ✓

See [Using c++ to call and use Windows Speech Recognition](#)

2 Which says:

Microsoft provides speech recognition engines for both client and server versions of Windows. Both can be programmed with C++ or with .NET languages. The traditional API for programming in C++ is known as SAPI. The .NET framework namespaces for client and server speech are System.Speech and Microsoft.Speech.

WABTEC EX1017

Lots of earlier questions have addressed this. See [Prototype based on speech recognition](#), [getting started with speech recognition and speech synthesis](#), and [SAPI and Windows 7 Problem](#) for examples.

Share Follow

edited May 23 '17 at 12:17

Community Bot
1 1

answered Jan 14 '11 at 21:29

Michael Levy
12.8k 15 60 95

I'd like to use C# and .NET, but other languages will suffice. Check this if you are open to C++ [Festival](#)

-1

Share Follow

answered Jan 13 '11 at 6:08

Mahesh
32.8k 17 82 112

There is a builtin DLL in every Windows OS for Text2Speech. You will find the according dll in c:\Programs\Shared Folders\Microsoft Shared\Speech\sapi.dll (sAPI - speech api) - I am not quite sure about the path - but in anyway you may search for sapi.dll.

-1

Afterwards you may use the following code snippet

```
SpVoice oVoice = new SpVoice();
oVoice.Voice = oVoice.GetVoices("", "").Item(0); // 0 indicating what kind of speaker
you want
oVoice.Volume = 50;
oVoice.Speak("hello world", SpeechVoiceSpeakFlags.SVSFDefault);
oVoice = null;
```

Share Follow

answered Jan 13 '11 at 7:01

Pilgerstorfer Franz
8,163 3 39 51

This is the path for Win7 C:\Windows\System32\Speech\Common – Serkan Hekimoglu Jan 13 '11 at 7:36

Speech to Text... not the other way around. – Rob Hay Sep 12 '11 at 7:32

1 @RobHay "Additionally, I will need an API or library that can do text-to-speech" so I think my answer is - at least - partially correct. – Pilgerstorfer Franz Jul 28 '12 at 12:12

For text to speech conversion you have to follow 3 steps:

-1

1.Add System.Speech reference.

2.Add Headers;

using System.Speech;

using System.Speech.Synthesis;

3.Add the following code where textBox1 is a Text Box default name.

```
SpeechSynthesizer speaker = new SpeechSynthesizer();
speaker.Rate = 1;
speaker.Volume = 100;
speaker.Speak(textBox1.Text);
```

Share Follow

answered Feb 6 '12 at 9:23

Rishi Jagati
606 1 6 38

works perfectly, this should be an accepted answer ; – ndoughmen May 16 '15 at 13:21

5 Except it is going the opposite direction from the question; instead of speech-to-text, this shows text-to-speech. – B. Clay Shannon Apr 12 '16 at 13:36

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

Sign up with Google

Sign up with GitHub

Sign up with Facebook

X

What is the difference between System.Speech.Recognition and Microsoft.Speech.Recognition?

Asked 11 years, 3 months ago · Active 6 years ago · Viewed 45k times



80

There are two similar namespaces and assemblies for speech recognition in .NET. I'm trying to understand the differences and when it is appropriate to use one or the other.



There is System.Speech.Recognition from the assembly System.Speech (in System.Speech.dll). System.Speech.dll is a core DLL in the .NET Framework class library 3.0 and later



42

There is also Microsoft.Speech.Recognition from the assembly Microsoft.Speech (in microsoft.speech.dll). Microsoft.Speech.dll is part of the UCMA 2.0 SDK

I find the docs confusing and I have the following questions:

System.Speech.Recognition says it is for "The Windows Desktop Speech Technology", does this mean it cannot be used on a server OS or cannot be used for high scale applications?

The UCMA 2.0 Speech SDK (<http://msdn.microsoft.com/en-us/library/dd266402%28v=office.12%29.aspx>) says that it requires Microsoft Office Communications Server 2007 R2 as a prerequisite. However, I've been told at conferences and meetings that if I do not require OCS features like presence and workflow I can use the UCMA 2.0 Speech API without OCS. Is this true?

If I'm building a simple recognition app for a server application (say I wanted to automatically transcribe voice mails) and I don't need features of OCS, what are the differences between the two APIs?

.net speech-recognition speech ucma2.0 ucs

Share Follow

asked Jun 4 '10 at 19:54



Michael Levy

12.8k 15 65 98

4 Answers



103

The short answer is that Microsoft.Speech.Recognition uses the Server version of SAPI, while System.Speech.Recognition uses the Desktop version of SAPI.



The APIs are mostly the same, but the underlying engines are different. Typically, the Server engine is designed to accept telephone-quality audio for command & control applications; the Desktop engine is designed to accept higher-quality audio for both command & control and dictation applications.



You can use System.Speech.Recognition on a server OS, but it's not designed to scale nearly as well as Microsoft.Speech.Recognition.

The differences are that the Server engine won't need training, and will work with lower-quality audio, but will have a lower recognition quality than the Desktop engine.

Share Follow

answered Jun 6 '10 at 2:39



Eric Brown

13.8k 7 29 68



53

I found [Eric's answer](#) really helpful, I just wanted to add some more details that I found.



System.Speech.Recognition can be used to program the desktop recognizers. SAPI and Desktop recognizers have shipped in the products:

- Windows XP: SAPI v5.1 and no recognizer
- Windows XP Tablet Edition: SAPI v5.1 and Recognizer v6.1
- Windows Vista: SAPI v5.3 and Recognizer v8.0
- Windows 7: SAPI v5.4 and Recognizer v8.0?

Servers come with SAPI, but no recognizer.

- Windows Server 2003: SAPI v5.1 and no recognizer
- Windows Server 2008 and 2008 R2: SAPI v5.3? and no recognizer

Desktop recognizers have also shipped in products like office.

- Microsoft Office 2003: Recognizer v6.1

Microsoft.Speech.Recognition can be used to program the server recognizers. Server recognizers have shipped in the products:

- Speech Server (various versions)
- Office Communications Server (OCS) (various versions)
- UCMA – which is a managed API for OCS that (I believe) included a redistributable recognizer
- Microsoft Server Speech Platform – recognizer v10.2

The complete SDK for the Microsoft Server Speech Platform 10.2 version is available at <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=1b1604d3-4665-4241-9a21-90a294a5c9a6>. The speech engine is a free download. Version 11 is now available at <http://www.microsoft.com/download/en/details.aspx?id=27226>.

For Microsoft Speech Platform SDK 11 info and downloads, see:

- Language packs - <http://www.microsoft.com/en-us/download/details.aspx?id=27224>
- Runtime engine - <http://www.microsoft.com/en-us/download/details.aspx?id=27225>
- SDK - <http://www.microsoft.com/en-us/download/details.aspx?id=27226>
- MSDN Documentation - <http://msdn.microsoft.com/en-us/library/dd266409>

Desktop recognizers are designed to run inproc or shared. Shared recognizers are useful on the desktop where voice commands are used to control any open applications. Server recognizers can only run inproc. Inproc recognizers are used when a single application uses the recognizer or when wav files or audio streams need to be recognized (shared recognizers can't process audio files, just audio from input devices).

Only Desktop speech recognizers include a dictation grammar (system provided grammar used for free text dictation). The class `System.Speech.Recognition.DictationGrammar` has no complement in the `Microsoft.Speech` namespace.

You can use the APIs to query determine your installed recognizers

- Desktop: `System.Speech.Recognition.SpeechRecognitionEngine.InstalledRecognizers()`
- Server: `Microsoft.Speech.Recognition.SpeechRecognitionEngine.InstalledRecognizers()`

I found that I can also see what recognizers are installed by looking at the registry keys:

- Desktop recognizers: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Recognizers\Tokens`
- Server recognizers: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech Server\v10.0\Recognizers\Tokens`

--- Update ---

As discussed in [Microsoft Speech Recognition - what reference do I have to add?](#), `Microsoft.Speech` is also the API used for the Kinect recognizer. This is documented in the MSDN article <http://msdn.microsoft.com/en-us/library/hh855387.aspx>

Share Follow

edited May 23 '17 at 12:17

Community Bot
1 1

answered Jun 8 '10 at 15:50

Michael Levy
12.8k 15 60 38

- 3 If I read the docs correctly, only Desktop speech recognizers include the dictation grammar (system provided grammar used for free text dictation). The class `System.Speech.Recognition.DictationGrammar` has no complement in the `Microsoft.Speech` namespace. – Michael Levy Jun 14 '10 at 22:49

According do MSDN API there is a `DictationGrammar` and a `WildcardGrammar` in C# (and I use it). But I don't know how to activate it throught XML (hacking parser?) see also: stackoverflow.com/questions/12181120/ – Jean-Philippe Encisse Sep 25 '12 at 20:51

I installed Italian language downloaded from microsoft.com/en-us/download/details.aspx?id=27224 but `System.Speech.Recognition.SpeechRecognitionEngine.InstalledRecognizers` shows en-US as only active language for speech recognition. I miss something? – P.O.W. Nov 13 '20 at 11:41

Here is the link for the Speech Library (MS Server Speech Platform):

7 [Microsoft Server Speech Platform 10.1 Released \(SR and ITS in 26 languages\)](#)

Share Follow

edited Feb 18 '14 at 9:54

answered Oct 1 '10 at 19:02

Nikolay Shmyrev

Switch Commerce

24k 5 38 85

168 2 4

- 2 10.2 was released recently too. microsoft.com/downloads/en/ - Michael Levy Oct 2 '10 at 17:22

Thanks for the info. I see that now includes a grammar validator. I was breaking my head trying to find errors in the ones I'm creating. Where should I keep watching for news about future releases? - Switch Commerce Oct 4 '10 at 18:19

"Where should I keep watching for news about future releases?" is a great question!! microsoft.com/speech/development.aspx is out of date. The speech blogs like blogs.msdn.com/s/speech and blogs.msdn.com/s/speech don't always have the latest updates. You might try gotospeech.net or the related sites like github.com. But as you can see, I haven't found a great source to keep up to date either. - Michael Levy Oct 5 '10 at 19:49

- 1 most recent at the time of writing is version 11 - microsoft.com/en-us/download/details.aspx?id=27225 - George Birbilis Sep 1 '15 at 16:05

btw, since grammar validator was mentioned above, see msdn.microsoft.com/en-us/library/9bb17840-7c25-425b-b0ff-1d3223.aspx for a list of grammar tools - George Birbilis Jan 4 '16 at 22:16

5 Seems Microsoft wrote an article that clears things up regarding the differences between Microsoft Speech Platform and Windows SAPI - <http://msdn.microsoft.com/en-us/library/9bb17840-7c25-425b-b0ff-1d3223.aspx>. A difference I found myself while converting Speech recognition code for Kinect from Microsoft.Speech to System.Speech (see <http://github.com/birbilis/Hotspotizer>) was that the former supports SGRS grammars with tag-format=semantics/1.0-literals, while the latter doesn't and you have to convert to semantics/1.0 by changing x to out="x", at tags

Share Follow

answered Sep 9 '15 at 20:55

George Birbilis

2,346 2 28 33

- 3 btw, you may find my SpeechLib code useful (speechlib.codeplex.com). You can remove the System.Speech reference from there and use Microsoft.Speech instead and set the appropriate conditional compilation symbol (see the sources) to use Microsoft.Speech in the code (mostly affects the using clauses, rest of the code is identical) - George Birbilis Jan 4 '16 at 22:04

- 1 the SpeechLib mentioned at the above comment has moved to github.com/geonicon/SpeechLib (since Codeplex is freezed to archive mode now) - George Birbilis May 17 '18 at 11:14

Android Developers Blog

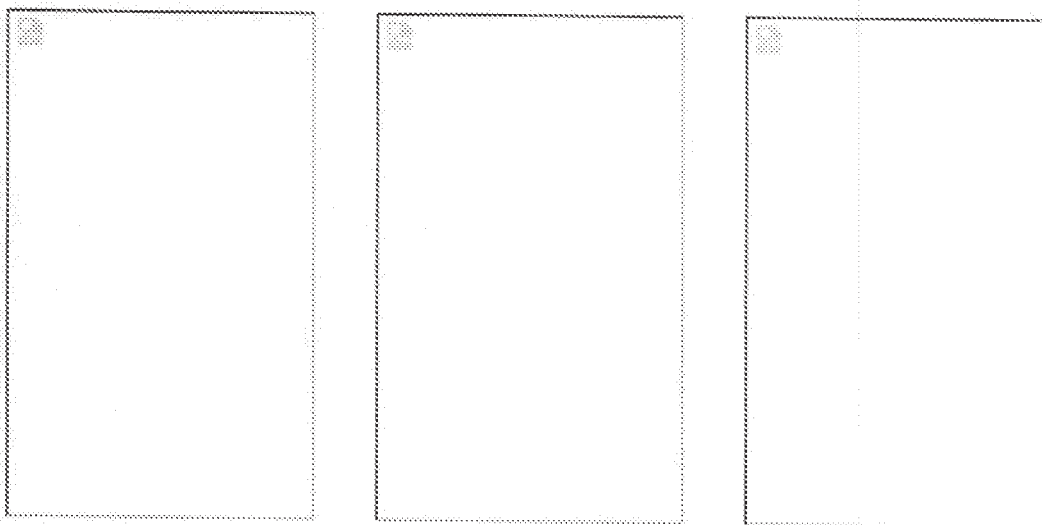
The latest Android and Google Play news for app and game developers.

Speech Input API for Android

05 March 2010

People love their mobile phones because they can stay in touch wherever they are. That means not just talking, but e-mailing, texting, microblogging, and so on. So, in addition to search by voice and voice shortcuts like "Navigate to", we included a voice-enabled keyboard in Android 2.1, which makes it even easier to stay connected. Now you can dictate your message instead of typing it. Just tap the new microphone button on the keyboard, and you can speak just about anywhere you would normally type.

We believe speech can fundamentally change the mobile experience. We would like to invite every Android application developer to consider integrating speech input capabilities via the Android SDK. One of my favorite apps in the Market that integrates speech input is Handcent SMS, because you can dictate a reply to any SMS with a quick tap on the SMS popup window.



Speech input integrated into Handcent SMS

The Android SDK makes it easy to integrate speech input directly into your own

≡ developers

recognition service on the device that's registered to receive a `RecognizerIntent`.

Google's Voice Search application, which is pre-installed on many Android devices, responds to a `RecognizerIntent` by displaying the "Speak now" dialog and streaming audio to Google's servers—the same servers used when a user taps the microphone button on the search widget or the voice-enabled keyboard. (You can check if Voice Search is installed in Settings → Applications → Manage applications.)

One important tip: for speech input to be as accurate as possible, it's helpful to have an idea of what words are likely to be spoken. While a message like "Mom, I'm writing you this message with my voice!" might be appropriate for an email or SMS message, you're probably more likely to say something like "weather in Mountain View" if you're using Google Search. You can make sure your users have the best experience possible by requesting the appropriate *language model*: "free_form" for dictation, or "web_search" for shorter, search-like phrases. We developed the "free form" model to improve dictation accuracy for the voice keyboard on the Nexus One, while the "web search" model is used when users want to search by voice.

Google's servers currently support English, Mandarin Chinese, and Japanese. The web search model is available in all three languages, while free-form has primarily been optimized for English. As we work hard to support more models in more languages, and to improve the accuracy of the speech recognition technology we use in our products, Android developers who integrate speech capabilities directly into their applications can reap the benefits as well.



Labels: Android 2.1 , IME , Speech Input , Text and Input



Global Google developer blogs



[Programa con Google \(Spanish LATAM\)](#)

[Google Developers Korea](#)

[Codigo \(Portuguese LATAM\)](#)

[Google Developers Japan](#)

[Developers Italia](#)

[Privacy](#) | [License](#) | [Brand guidelines](#)

Get news and tips by email

[SUBSCRIBE](#)