



Vestec Automatic Speech Recognition Engine
Standard Edition
Version 1.1.1

Asterisk Integration Guide



Vestec Automatic Speech Recognition Engine
Standard Edition
Version 1.1.1
Asterisk Integration Guide

Copyright© 2009 Voice Enabling Systems Technology, Inc. All rights reserved.

145 Columbia Street West, Suite 1, Waterloo, Ontario, Canada N2L 3L2

Information in this document is subject to change without notice and does not represent a commitment on the part of VESTEC, Inc. The software described in this document is provided under a license agreement or nondisclosure agreement. You may not copy, use, modify or distribute the software without the express written permission of Vestec, Inc.

Table of Contents

<u>About This Document</u>	3
<u>Audience</u>	3
<u>Organization</u>	3
<u>Conventions</u>	3
1 <u>Installation</u>	4
2 <u>Testing</u>	5
3 <u>Writing Dialplan for VASRE</u>	7
4 <u>Monitoring VASRE from Asterisk</u>	11

About This Document

Vestec's Automatic Speech Recognition Engine (VASRE) is a speaker-independent speech recognition engine that supports a distributed architecture of servers and clients. VASRE works for Windows, GNU/Linux, and Open Solaris platforms to process an audio file or stream from external sources, such as telephone systems. The grammar can be built simply, with a list of keywords or pronunciations to be recognized, or with a more sophisticated industry standard.

This guide explains how to install and use VASRE plug-in for Asterisk, a software PBX.

Audience

This guide is intended for speech application developers who are going to use VASRE for Asterisk.

Organization

This guide is organized as follows:

- Section 1 presents the installation procedure of the VASRE plug-in for Asterisk.
- Section 2 describes the testing procedure of the VASRE plug-in.
- Section 3 explains Dialplan applications used for VASRE plug-in.
- Section 4 lists CLI and AMI commands to monitor VASRE from Asterisk.

Conventions

Guides for VASRE use the following conventions:

- **Bold Arial** represents user utterances, recognized strings, and semantic results.
- `Courier New` represents file names, directory names, command line strings, and file contents.
- *Italic text* represents types, tokens, keywords, variables, and functions.
- Underlined text represents menu strings or texts in graphical user interface.
- *Italic Courier New* represents values replaced by you. For example, *YYYY-MM-DD* may represent a date in the year-month-day format.
- A paragraph starting with **N.B.** represents critical information or warning.
- For abbreviated terms, both singular and plural are spelled the same. For example, RM represents both resource manager and resource managers.

1 Installation

For GNU/Linux users, VASRE provides a software package dedicated to Asterisk integration. The main component of the package is a shared object file, which works as a plug-in (connector) between VASRE and Asterisk.

N.B. You should install Asterisk first before installing the VASRE plug-in.

N.B. At this point, the VASRE plug-in is available for GNU/Linux only. Windows users cannot use VASRE in conjunction with Asterisk.

You can either download the Asterisk source code from the Digium website and compile it, or get the precompiled binaries from the package manager of your GNU/Linux distribution. For more details, refer to the Digium website (<http://www.digium.com/en/supportcenter/>).

The package name of the VASRE plug-in for Asterisk is:

```
vasre-se-ast_astver-plugin-version-arch
```

where *astver*, *version*, and *arch* represent Asterisk version, VASRE version, and platform architecture, respectively. For example, the package name for VASRE Standard Edition 1.1.1 for Centos 5, i386, and Asterisk 1.4 would be:

```
vasre-se-ast_1.4-plugin-1.1.1~Centos_5-i386
```

The package is provided in two formats: `.deb` and `.rpm`. To install the `.deb` package, open a shell terminal and enter the following command:

```
dpkg -i vasre-se-ast_astver-plugin-version-arch.deb
```

To install the `.rpm` package, enter the following command:

```
rpm -i vasre-se-ast_astver-plugin-version-arch.rpm
```

Upon successful installation, you should have the following file and directories newly created:

- `/usr/lib/asterisk/modules/res_speech_vasre.so`
- `/opt/VestecASRE/Samples/Dialplan/`
- `/etc/VestecASRE/vasre-ast.conf`

If you need further help regarding the installation of the VASRE plug-in, visit our website at http://www.vestec.ca/support_overview.

2 Testing

To test Asterisk with VASRE, follow the steps below:

1. Configure Asterisk: Modify Asterisk's configuration files `iax.conf` and `sip.conf` under `/etc/asterisk/` to make Asterisk receive a call. For example, if you are going to use Asterisk for IP-telephone only, install any soft-phone and modify `sip.conf` as shown:

```
; ...
[general]
allowoverlap=no
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
[100]
type=friend
context=default
host=dynamic
; ...
```

2. Test Asterisk: Run Asterisk Command-Line Interface (CLI) using the following command:

```
/usr/sbin/asterisk -cvvv
```

If Asterisk is already running, the above command will fail. If this is the case, use the following to connect to running Asterisk:

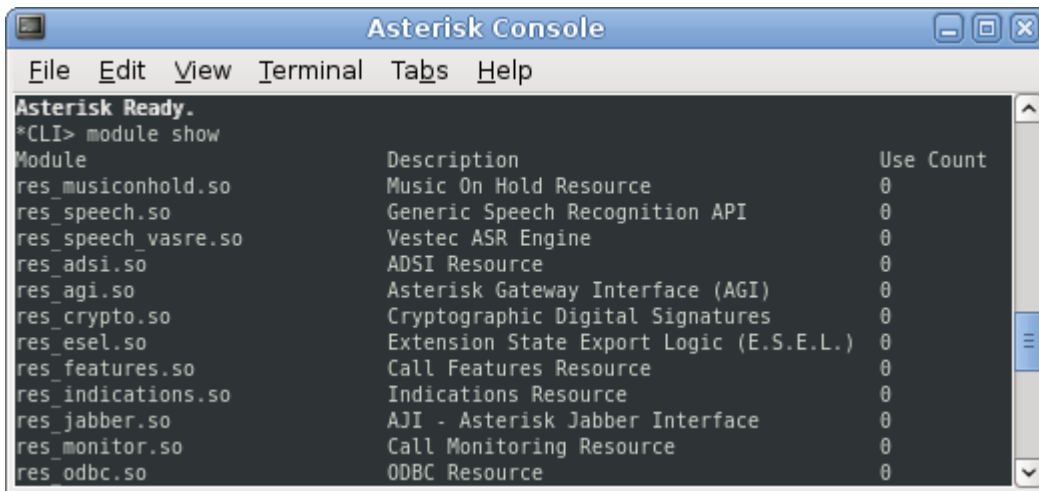
```
/usr/sbin/asterisk -rvvv
```

Modify `/etc/asterisk/extensions.conf` to set up Dialplan for test calls. Dialplan is a script describing the operation of Asterisk. Verify that Asterisk is responding to your call. See the Asterisk manual for further details.

3. Add the following two lines to Asterisk's `modules.conf` to make Asterisk load the shared object file of VASRE plug-in at start-up:

```
load => res_speech.so
load => res_speech_vasre.so
```

4. Apply the changes using `module reload` command from the Asterisk CLI. Double-check if `res_speech_vasre.so` was loaded using `module show` command as shown below:



```

Asterisk Ready.
*CLI> module show
Module                Description                Use Count
res_musiconhold.so    Music On Hold Resource     0
res_speech.so         Generic Speech Recognition API 0
res_speech_vasre.so   Vestec ASR Engine          0
res_adsi.so           ADSI Resource              0
res_agi.so            Asterisk Gateway Interface (AGI) 0
res_crypto.so         Cryptographic Digital Signatures 0
res_esel.so           Extension State Export Logic (E.S.E.L.) 0
res_features.so       Call Features Resource     0
res_indications.so    Indications Resource       0
res_jabber.so         AJI - Asterisk Jabber Interface 0
res_monitor.so        Call Monitoring Resource    0
res_odbc.so           ODBC Resource              0

```

5. Start RM and servers as described in Section 2.2 of Administration Guide. If the RM is running from a remote machine, you should modify `/etc/VestecASRE/vasre-ast.conf`. By default the file has one line starting with `-rm` keyword and localhost IP address, `127.0.0.1` separated by a white space. Replace the IP address part with the IP address of the RM. For example, if RM is running at 192.168.1.2, the line should be `-rm 191.168.1.2`. VASRE plug-in reads this file to communicate with the RM.
6. Modify `extensions.conf` as follows to transfer incoming calls to the VASRE sample Dialplan:
 - Include `vasre-demo.conf` under context `general`
 - Add `Goto` application under context `default` as shown below:

```

; ...
[general]
#include "/opt/VestecASRE/Samples/Dialplan/vasre-demo.conf"
; ...
[default]
exten => 500,1,Goto(vasre-demo-digit4,s,1)
; ...

```

7. Run `dialplan reload` command to apply this change.
8. Make a call to Asterisk and dial 500 to test the sample Dialplan.

The sample Dialplan `/opt/VestecASRE/Samples/Dialplan/vasre-demo.conf` simulates Asterisk application recognizing four-digit numbers.

3 Writing Dialplan for VASRE

The contents of the sample Dialplan show how you can open a VASRE port, set the parameters, control grammars, perform streamed audio processing, and get the result. The following text is the abbreviated version of the sample Dialplan:

```
[vasre-demo-digit4]
exten => s,1,Answer
exten => s,2,Wait(1)
exten => s,3,SpeechCreate(Vestec ASR Engine)
exten => s,4,GotoIf("${ERROR}" = "1"?5:6)
exten => s,5,Hangup
exten => s,6,Goto(start-vasre-demo-digit4,s,1)

[start-vasre-demo-digit4]
exten => s,1,Playback(/opt/VestecASRE/Samples/Dialplan/welcome)
exten => s,n,Set(THRESHOLD=10)
exten => s,n,Set(SPEECH_ENGINE(VadAmpThres)=2000)
exten => s,n,Set(SPEECH_ENGINE(CompMaxAmp)=0)
exten => s,n,Goto(vasre-digit4,s,1)

[vasre-digit4]
exten => s,1,SpeechLoadGrammar(digit4,/opt/VestecASRE/Samples/Dialplan/digit4.grm)
exten => s,2,SpeechActivateGrammar(digit4)
exten => s,3,SpeechStart
exten => s,4,SpeechBackground(/opt/VestecASRE/Samples/Dialplan/ask,8)
exten => s,5,GotoIf("${SPEECH(results)}" = "0"?6:8)
exten => s,6,Playback(/opt/VestecASRE/Samples/Dialplan/ask_again)
exten => s,7,Goto(4)
exten => s,8,Set(TEXT=${SPEECH_TEXT(0)})
exten => s,9,GotoIf("${SPEECH_SCORE(0)} > ${THRESHOLD} ]?10:7)
exten => s,10,Playback(/opt/VestecASRE/Samples/Dialplan/say_num)
; play back input
exten => s,19,SpeechDeactivateGrammar(digit4)
exten => s,20,Goto(vasre-demo-confirm,s,1)

[vasre-demo-confirm]
; confirm input
exten => s,13,Goto(vasre-demo-complete,s,1)

[vasre-demo-complete]
exten => s,1,Playback(/opt/VestecASRE/Samples/Dialplan/thankyou)
exten => s,2,SpeechDestroy
exten => s,3,Wait(1)
exten => s,4,Hangup
```

The incoming call passes through five contexts `vasre-demo-digit4`, `start-vasre-demo-digit4`, `vasre-digit4`, `vasre-demo-confirm`, and `vasre-demo-complete` in turn.

To initialize VASRE and open a port, the Dialplan should run `SpeechCreate` application first as in Priority 3 of `vasre-demo-digit4`. Make sure `Vestec ASR Engine` is used as the argument. If `res_speech_vasre.so` is missing, `/etc/VestecASRE/vasre-ast.conf` contains a wrong IP address of the RM, or the RM and recognition server are not ready, this step will fail and `${ERROR}` will be set to 1. Check the value of `${ERROR}` before progressing.

Use `Set` application to change VASRE parameters as shown in the third priority of `start-vasre-demo-digit4`. There are nine adjustable parameters:

- `MinSpeechLen`: The minimum speech length accepted by the server (0–1000; default: 300)
- `MaxSpeechLen`: The maximum speech length accepted by the server (1000–20000; default: 10000)
- `VadAmpThres`: The energy threshold for voice activation detection (200–30000; default: 1000)
- `VadZcThres`: The zero-cross threshold for voice activation detection (3–200; default: 10)
- `VadPtaThres`: The pitch threshold for voice activation detection (100–500; default: 400)
- `VadPtcThres`: The pitch count threshold for voice activation detection (1–10; default: 7)
- `VadHeadMargin`: The head margin of audio for voice activation detection (150–500; default: 300)
- `VedTailMargin`: The tail margin of audio for voice end detection (200–1200; default: 500)
- `CompMaxAmp`: The flag indicating if maximum amplitude of processed audio will be output (0 or 1; default: 0)

Refer to Section 3.2 of the Application Developer's Guide for the ranges and mode details for each parameter.

You may load multiple grammars and selectively activate one or more of them for speech recognition. Use the `SpeechLoadGrammar` application to add a grammar to the server as illustrated in Priority 1 of `vasre-digit4`. The first and second arguments of `SpeechLoadGrammar` are the tag and file path name of the grammar to be added. A grammar tag must start with an alphabetical letter and comprise alphanumeric letters, hyphens, underscores, and periods. The grammar path name must end with either `.grm` (text grammar) or `.gout` (binary grammar).

To delete the added grammar, use the `SpeechUnloadGrammar` application with the tag. Note that all the grammars added to the server will be automatically deleted when `SpeechDestroy` is executed. You should delete grammars only when you would like to maintain the vocabulary size of the added grammars at less than the license limit.

Use `SpeechActivateGrammar` or `SpeechDeactivateGrammar` to switch the activeness of loaded grammars as illustrated in Priority 2 of `vasre-digit4`.

To start the streamed audio processor, use the `SpeechStart` application as in Priority 3. You should use the `SpeechBackGround` application as the counterpart of `SpeechStart` to recognize the caller's speech while playing back a prerecorded prompt message as shown in Priority 4. Once speech is detected, `Dialplan` will stop playing the prompt and process the caller's speech. To adjust the sensitivity of speech detection, use `Set` application with the parameters listed above.

The following `Dialplan` functions are available to process the output of VASRE:

- `_${SPEECH(results)}`: Number of n-best alternatives
- `_${SPEECH_SCORE(result number)}`: Score of zero-indexed n-best alternative
- `_${SPEECH_TEXT(result number)}`: Text of zero-indexed n-best alternative
- `_${SPEECH_GRAMMAR(result number)}`: Matched grammar tag of zero-indexed n-best

alternative

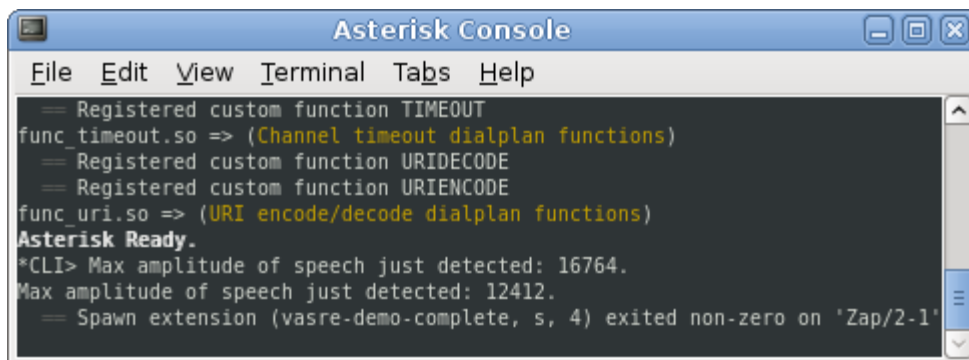
According to the value of `n-best` parameter specified when starting the server, zero or more `n-best` alternatives can be found by the server. `$(SPEECH(results))` returns this number. `$(SPEECH_SCORE(result number))`, `$(SPEECH_TEXT(result number))`, and `$(SPEECH_GRAMMAR(result number))` return the score, text, and matched grammar tag of an `n-best` alternative. `result number` represents zero-based index of the `n-best` alternative. For example, `result number` of zero represents the `n-best` alternative with the highest score, `result number` of one represents the one with the second highest score, and so on. If `result number` is not less than the number of `n-best` alternatives, an empty string will be returned.

`SPEECH_SCORE` returns a confidence score ranging from 0 to 1000 of the `n-best` alternative. Note that `SPEECH_TEXT` invokes `VA_GetTextOfNbestAlter` and returns the semantic result of the `n-best` alternative. If the semantic interpretation processor fails due to a syntax error in SI scripts or internal errors, `SPEECH_TEXT` returns the raw text recognized. See Application Developer's Guide for further details.

Before hanging up the call, invoke the `SpeechDestroy` application to close the port from the VASRE. You should have the pair of `SpeechCreate` and `SpeechDestroy` in Dialplan.

N.B. If your Asterisk system works for an analog telephony system and your interface card has no echo cancellation feature, you may suffer from the echoing of TTS voice. That is, the TTS voice played by Asterisk is falsely recognized as user speech and barged-in itself. If this happens, increase the value of `VadAmpThres` to reject the echo.

To help you to choose the proper threshold value, VASRE can output the maximum amplitude of the speech just detected to Asterisk CLI with the verbosity level of two or higher. In your Dialplan, set 1000 and 1 to `VadAmpThres` and `CompMaxAmp`, respectively. Restart Dialplan and leave it to detect the TTS echo as speech input. Then, the CLI will show you the maximum amplitude of the echo just detected. If the verbosity level of the CLI is two, you will see the messages as follows:



```

Asterisk Console
File Edit View Terminal Tabs Help
= Registered custom function TIMEOUT
func_timeout.so => (Channel timeout dialplan functions)
= Registered custom function URIDECODE
= Registered custom function URIENCODE
func_uri.so => (URI encode/decode dialplan functions)
Asterisk Ready.
*CLI> Max amplitude of speech just detected: 16764.
Max amplitude of speech just detected: 12412.
= Spawn extension (vasre-demo-complete, s, 4) exited non-zero on 'Zap/2-1'

```

If you change the amplitude threshold value to the maximum amplitude of the TTS echo observed, the echo will be effectively rejected. If you don't want to see the maximum amplitude of just-detected speech anymore, set `CompMaxAmp` value to zero.

All in all, you may use the following Dialplan applications to work with VASRE:

- `SpeechCreate(engine name)`: Open a port toward the server.
- `SpeechDestroy()`: Close the opened port. The application must be executed as the pair of `SpeechCreate`.

- `SpeechLoadGrammar (Grammar Tag | Path)`: Add grammar to the server.
- `SpeechUnloadGrammar (Grammar Tag)`: Delete grammar from the server.
- `SpeechActivateGrammar (Grammar Tag)`: Activate grammar.
- `SpeechDeactivateGrammar (Grammar Tag)`: Deactivate grammar.
- `SpeechStart ()`: Start the streamed audio processor.
- `SpeechBackground (Sound File | Timeout)`: Play prompt and process incoming audio.
- `SpeechProcessingSound (Sound File)`: Process an audio file. Only PCM with a 8kHz sampling rate will be acceptable.

4 Monitoring VASRE from Asterisk

The VASRE plug-in provides three commands to enable you to monitor VASRE from Asterisk CLI:

- `vasre show version`: Show VASRE and plug-in version information
- `vasre show license`: Show the number of licenses purchased and how many of them are in use
- `vasre show servers`: Show the status of servers

These commands are also available for Asterisk Manager Interface (AMI):

- `VasreShowVersion`: Show VASRE and plug-in version information
- `VasreShowLicense`: Show the number of licenses purchased and how many of them are in use
- `VasreShowServers`: Show the status of servers