



# Asterisk

## Watson SR and SS

### Usage Guide

---

Revision: 1

Created: September 28, 2018

Last updated: September 28, 2018

Author: Arsen Chaloyan

# Table of Contents

1 Overview.....	4
1.1 Applicable Versions.....	4
2 Generic Speech Recognition API .....	5
2.1 Overview.....	5
2.2 Configuration Steps .....	5
2.3 Usage Examples.....	5
3 Suite of UniMRCP Applications.....	7
3.1 Overview.....	7
3.2 Configuration Steps .....	7
3.3 Usage Examples.....	8
MRCPRecog.....	8
MRCPSynth.....	9
SynthAndRecog.....	9



# 1 Overview

This guide describes how to utilize the IBM Watson Speech services with Asterisk.



Note that the Asterisk and the UniMRCP server typically reside on different hosts in a LAN, although both might be installed on the same host.

Installation of the Asterisk and the UniMRCP server with the Watson SR and SS plugins is not covered in this document. Visit the corresponding web pages for more information.

<http://unimrcp.org/asterisk>  
<http://unimrcp.org/wsr>  
<http://unimrcp.org/wss>

## 1.1 Applicable Versions

Instructions provided in this guide are applicable to the following versions.



UniMRCP Modules for Asterisk 1.5.0 and above  
UniMRCP Watson SR Plugin 1.0.0 and above  
UniMRCP Watson SS Plugin 1.0.0 and above

# 2 Generic Speech Recognition API

## 2.1 Overview

The module *res\_speech\_unimrcp.so* provides an implementation of the Generic Speech Recognition API of Asterisk, based on the UniMRCP client library.

## 2.2 Configuration Steps

This section outlines major configuration steps required for use of the module *res\_speech\_unimrcp.so* with the UniMRCP server.

Check the name of the profile, referenced in the configuration file *res-speech-unimrcp.conf*, which is located in the configuration directory of Asterisk. Use *uni2* for MRCPv2, and *uni1* for MRCPv1.

```
[general]
unimrcp-profile = uni2 ; UniMRCP MRCPv2 Server
; unimrcp-profile = uni1 ; UniMRCP MRCPv1 Server
```

Open the configuration file *unimrcpclient.xml*, located in the configuration directory of UniMRCP and specify IP addresses of the client and the server. In the following example, the Asterisk/UniMRCP client is located on 10.0.0.1 and the UniMRCP server is on 10.0.0.2.

```
<properties>
  <!-- <ip type="auto"/> -->

  <ip>10.0.0.1</ip>
  <server-ip>10.0.0.2</server-ip>
</properties>
```

## 2.3 Usage Examples

Make use of a built-in speech grammar *transcribe* for recognition, having no speech contexts defined, by adding the following entries in the Asterisk configuration file *extensions.conf*.

```
[res-speech-unimrcp-1]
exten => s,1,Answer()
exten => s,2,SpeechCreate()
exten => s,3,SpeechActivateGrammar(builtin:speech/transcribe)
exten => s,4,SpeechBackground(beep, 20)
exten => s,5,Verbose(1, "Recognition result count: ${SPEECH(results)}")
```

```
exten => s,6,GotoIf(["${SPEECH(results)}" = "0"]?7:9)
exten => s,7,Playback(error)
exten => s,8,Goto(3)
exten => s,9,Verbose(1, "Recognition result: ${SPEECH_TEXT(0)}, confidence score:
${SPEECH_SCORE(0)}, grammar-uri: ${SPEECH_GRAMMAR(0)}")
exten => s,10,SpeechDestroy()
exten => s,11,Hangup()
```

In the Asterisk dialplan, associate the provided sample to an extension, for example, 701.

```
exten => 701,1,Goto(res-speech-unimrcp-1,s,1)
```

Place a test call and make sure recognition works as expected.

# 3 Suite of UniMRCP Applications

## 3.1 Overview

The module *app\_unimrcp.so* provides a suite of speech recognition and synthesis applications for Asterisk.

## 3.2 Configuration Steps

This section outlines major configuration steps required for use of the module *app\_unimrcp.so* with the UniMRCP server.

Open the configuration file *mrcp.conf*, located in the configuration directory of Asterisk, and add two profiles for MRCPv2 and MRCPv1 respectively. In the following example, the Asterisk/UniMRCP client is located on 10.0.0.1 and the UniMRCP server is on 10.0.0.2.

```
[uni2]
; MRCP settings
version = 2
;
; SIP settings
server-ip = 10.0.0.2
server-port = 8060
; SIP user agent
client-ip = 10.0.0.1
client-port = 25097
sip-transport = udp
;
; RTP factory
rtp-ip = 10.0.0.1
rtp-port-min = 28000
rtp-port-max = 29000
;
; Jitter buffer settings
playout-delay = 50
max-playout-delay = 200
; RTP settings
ptime = 20
codecs = PCMU PCMA L16/96/8000 telephone-event/101/8000
; RTCP settings
rtcp = 0
```

```
[uni1]
```

```

; MRCP settings
version = 1
;
; RTSP settings
server-ip = 10.0.0.2
server-port = 1554
resource-location = media
speechsynth = speechsynthesizer
speechrecog = speechrecognizer;
;
; RTP factory
rtp-ip = 10.0.0.1
rtp-port-min = 27000
rtp-port-max = 28000
;
; Jitter buffer settings
playout-delay = 50
max-playout-delay = 200
; RTP settings
ptime = 20
codecs = PCMU PCMA L16/96/8000 telephone-event/101/8000
; RTCP settings
rtcp = 0

```

### 3.3 Usage Examples

#### MRCPreCog

Make use of a built-in speech grammar *transcribe* for recognition, by adding the following entries in the Asterisk configuration file *extensions.conf*.

```

[app-unimrcp-1]
exten => s,1,Answer
exten => s,2,MRCPreCog("bultin:speech/transcribe", spl=en-US&f=beep&p=uni2)
exten => s,3,Goto(mrcpreCog-output,s,1)

```

Note the macro *mrcpreCog-output*, used in the example above, is defined in the file *mrcp\_sample\_apps.conf*.

In the Asterisk dialplan, associate the provided sample to an extension, for example, 801.

```

exten => 801,1,Goto(app-unimrcp-1,s,1)

```

Place a test call and make sure recognition works as expected.

## MRCPSynth

Use the application MRCPSynth to speech synthesis.

```
[app-unimrcp-2]
exten => s,1,Answer
exten => s,2,MRCPSynth(Welcome to Asterisk,l=en-US&p=uni2)
```

In the Asterisk dialplan, associate the provided sample to an extension, for example, 802.

```
exten => 802,1,Goto(app-unimrcp-2,s,1)
```

Place a test call and listen to the synthesized message.

## SynthAndRecog

Use the application SynthAndRecog for speech synthesis and recognition.

```
[app-unimrcp-3]
exten => s,1,Answer
exten => s,2, SynthAndRecog(Please say something,builtin:speech/transcribe, spl=en-US&p=uni2)
exten => s,3,Goto(mrcprecog-output,s,1)
```

In the Asterisk dialplan, associate the provided sample to an extension, for example, 803.

```
exten => 803,1,Goto(app-unimrcp-3,s,1)
```

Place a test call and listen to the synthesized prompt and say something. Make sure recognition works as expected.