mod_unimrcp

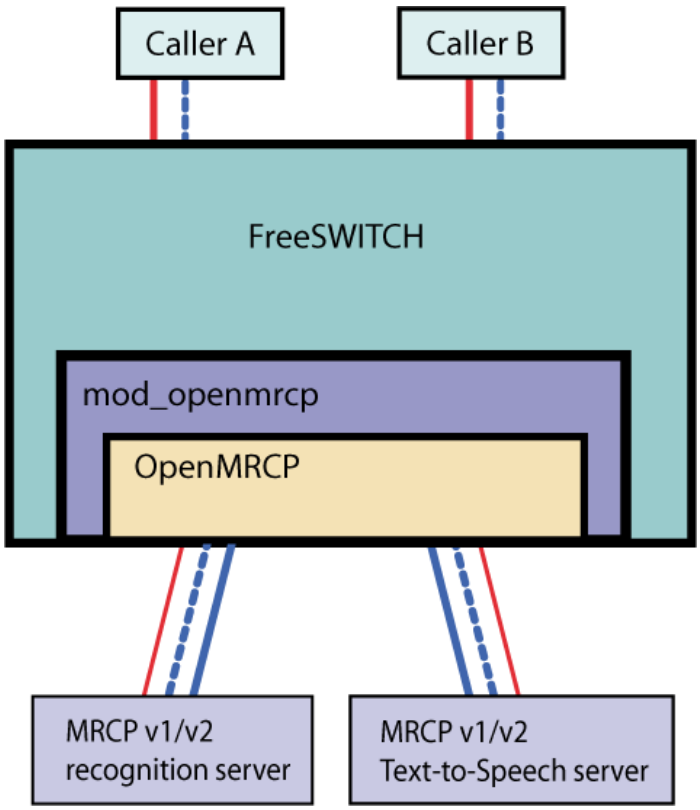
About

mod_unimrcp is the FreeSWITCH module that allows communication with Media Resource Control Protocol (MRCP) servers. MRCP allows client machines to control media resources on a network. MRCP version 1 uses the Real Time Streaming Protocol (RTSP) while version 2 uses the Session Initiation Protocol (SIP) to negotiate the MRCP connection. mod_unimrcp allows FreeSWITCH to act as such a client. Servers are supplied by numerous vendors such as Cepstral, Voxeo, Nuance, and many others.

Some of the media resources that can be controlled via MRCP:

- Automatic speech recognition (ASR)
- Text to speech (TTS)
- Speaker verification and identification (not currently supported by mod_unimrcp)

Click here to expand Table of Contents



Compatibility

mod_unimrcp has been tested with the following products

- UniMRCP Server
- Speech Technology Center, VoiceNavigator
- Nuance Speech Server 5.0/5.1
- Voxeo Prophecy 8.0
- Loquendo Suite 7.0
- Acapela TTS for Windows Server and Linux Server 6.400 with MRCPv2 add-on (As Acapela MRCPv2 add-on is based on UniMRCP Server: Reference)
- IVONA Telecom 1.6

Building

- Add asr_tts/mod_unimrcp to modules.conf
- Build FreeSWITCH with 'make install'

Configuration

- Edit conf/autoload_config/unimrcp.conf.xml
- Add MRCP profiles to conf/mrcp_profiles/
- `Add  <load module="mod_unimrcp"/> to autoload_configs/modules.conf.xml`

Example Configuration

unimrcp.conf.xml

```
<configuration name="unimrcp.conf" description="UniMRCP Client">
 <settings>
   <param name="default-tts-profile" value="voxeo-prophecy8.0-mrcp1"/>
   <param name="default-asr-profile" value="voxeo-prophecy8.0-mrcp1"/>
   <param name="log-level" value="DEBUG"/>
   <param name="max-connection-count" value="100"/>
   <param name="offer-new-connection" value="1"/>
 </settings>
 <profiles>
   <X-PRE-PROCESS cmd="include" data="../mrcp_profiles/*.xml"/>
 </profiles>
</configuration>
```

- **default-tts-profile** - what profile to use for client settings for contacting a TTS server when one isn't specified.
- **default-asr-profile** - what profile to use for client settings for contacting an ASR server when one isn't specified.
- **log-level** - log level for UniMRCP client library, valid values: EMERGENCY|ALERT|CRITICAL|ERROR|WARNING|INFO|DEBUG.
- **max-connection-count** - max number of MRCPv2 connections to manage.
- **offer-new-connection** - no idea... need to check UniMRCP documentation

mrcp_profiles

An MRCP profile allows you to define a configuration for a specific MRCP server. This allows you to integrate different types of MRCP servers with FreeSWITCH. Each profile defines the MRCP version to use, the client and server addresses and ports, codec preferences, and any default parameters to send to the server.

Default MRCP parameters for SPEAK and RECOGNIZE are set in synthparams and recogparams, respectively. See the appropriate RFC and your MRCP server documentation for the available MRCP parameters that may be used.

MRCPv1 example

MRCP version 1 is usually supported by MRCP servers. It uses RTSP over a TCP connection to send requests to the server. RTP is used to send audio between the client and server.

```
<include>
 <profile name="mrcpserver01" version="1">
   <param name="server-ip" value="10.10.5.1"/>
   <param name="server-port" value="554"/>
   <param name="resource-location" value=""/>
   <param name="speechsynth" value="synthesizer"/>
   <param name="speechrecog" value="recognizer"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
   <synthparams>
   </synthparams>
   <recogparams>
       <param name="start-input-timers" value="false"/>
   </recogparams>
 </profile>
</include>
```

- **server-ip** - IP address of MRCP server
- **server-port** - RTSP port. Possibly 554 or 1554. Check your MRCP server documentation.
- **resource-location** - "media" or blank
- **speechsynth** - "speechsynthesizer" or "synthesizer"
- **speechrecog** - "speechrecognizer" or "recognizer"
- **rtp-ext-ip** - External IP address for client RTP
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** - Start of RTP port range
- **rtp-port-max** - End of RTP port range
- **playout-delay** -
- **max-playout-delay** -
- **ptime** - ptime to negotiate with MRCP server
- **codecs** - codec negotiation preference. Server will probably support PCMU, PCMA, or L16

MRCPv2 example

MRCP version 2 is currently a draft standard, but is supported by some MRCP servers like Nuance Speech Server. MRCPv2 uses SIP to manage sessions and its own protocol to send speech requests. RTP is used to send audio between the client and server.

```
<include>
 <profile name="mrcpserver02" version="2">
   <param name="client-ip" value="10.10.5.2"/>
   <param name="client-port" value="5090"/>
   <param name="server-ip" value="10.5.5.152"/>
   <param name="server-port" value="5060"/>
   <param name="sip-transport" value="udp"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
   <synthparams>
   </synthparams>
   <recogparams>
       <param name="start-input-timers" value="false"/>
   </recogparams>
 </profile>
</include>
```

- **client-ext-ip** - External SIP IP address of MRCP client
- **client-ip** - SIP IP address of MRCP client
- **client-port** - SIP port of MRCP client (make sure it doesn't conflict with conf/sip_profiles)
- **server-ip** - SIP IP address of MRCP server
- **server-port** - SIP port of MRCP server
- **force-destination** -
- **sip-transport** - "udp" or "tcp"
- **ua-name** - User agent name
- **sdp-origin** -
- **rtp-ext-ip** - External IP address for client RTP
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** - Start of RTP port range
- **rtp-port-max** - End of RTP port range
- **playout-delay** -
- **max-playout-delay** -
- **ptime** - ptime to negotiate with MRCP server
- **codecs** - codec negotiation preference. Server will probably support PCMU, PCMA, or L16

Sample profiles

The FreeSWITCH trunk contains conf/mrcp_profiles for various MRCP servers. A few of those profiles are described below:

Speech Technology Center, VoiceNavigator

```
<include>
 <profile name="stc-vn-mrcp1" version="1">
   <param name="server-ip" value="10.5.5.152"/>
   <param name="server-port" value="8000"/>
   <param name="resource-location" value=""/>
   <param name="speechsynth" value="tts"/>
   <param name="speechrecog" value="asr"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="32768"/>
   <param name="rtp-port-max" value="33268"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
 </profile>
</include>
```

- **server-ip** - IP address of MRCP server
- **server-port** - RTSP port. 8000
- **resource-location** - "media" or blank
- **speechsynth** - "tts"
- **speechrecog** - "asr"
- **rtp-ext-ip** -
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** - "32768"
- **rtp-port-max** - "33268"
- **playout-delay** -
- **max-playout-delay** -
- **ptime** - ptime to negotiate with MRCP server
- **codecs** - codec preference

Voxeo Prophecy 8.0

```
<include>
 <profile name="voxeo-prophecy8.0-mrcp1" version="1">
   <param name="server-ip" value="99.185.85.31"/>
   <param name="server-port" value="554"/>
   <param name="resource-location" value=""/>
   <param name="speechsynth" value="synthesizer"/>
   <param name="speechrecog" value="recognizer"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
 </profile>
</include>
```

- **server-ip** - IP address of MRCP server
- **server-port** - RTSP port. 4900 or 554
- **resource-location** - "media" or blank
- **speechsynth** - "speechsynthesizer" or "synthesizer"
- **speechrecog** - "speechrecognizer" or "recognizer"
- **rtp-ext-ip** -
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** -
- **rtp-port-max** -
- **playout-delay** -
- **max-playout-delay** -
- **ptime** - ptime to negotiate with MRCP server
- **codecs** - codec preference

Nuance Speech Server 5.0

```
<include>
 <profile name="nuance5-mrcp2" version="2">
   <param name="client-ip" value="auto"/>
   <param name="client-port" value="5090"/>
   <param name="server-ip" value="10.5.5.152"/>
   <param name="server-port" value="5060"/>
   <param name="sip-transport" value="udp"/>
   <param name="rtp-ip" value="auto"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
 </profile>
</include>
```

- **client-ext-ip** -
- **client-ip** - IP address of MRCP client
- **client-port** - SIP port of MRCP client (make sure it doesn't conflict with conf/sip_profiles)
- **server-ip** - IP address of MRCP server
- **server-port** - SIP port of MRCP server
- **force-destination** -
- **sip-transport** - "udp" or "tcp"
- **ua-name** - User agent name
- **sdp-origin** -
- **rtp-ext-ip** -
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** -
- **rtp-port-max** -
- **playout-delay** -
- **max-playout-delay** -

- **ptime** - ptime to negotiate with MRCP server
- **codecs** - codec preference

Loquendo Suite 7.0

```
<include>
 <profile name="loquendo-7-mrcp2" version="2">
   <param name="client-ip" value="auto"/>
   <param name="client-port" value="5090"/>
   <param name="server-ip" value="127.0.0.1"/>
   <param name="server-port" value="5060"/>
   <param name="sip-transport" value="udp"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="PCMU PCMA L16/96/8000"/>
   <param name="jsgf-mime-type" value="application/jsgf"/>
 </profile>
 </include>
```

- **client-ip** - IP address of MRCP client
- **client-port** - SIP port of MRCP client (make sure it doesn't conflict with conf/sip_profiles)
- **server-ip** - IP address of MRCP server
- **server-port** - SIP port of MRCP server (this defaults to 5060 in the Loquendo config, which may conflict with FS)
- **sip-transport** - "udp" or "tcp"
- **rtp-ip** - IP address for client RTP
- **rtp-port-min** -
- **rtp-port-max** -
- **codecs** - codec preference

IVONA Telecom 1.6 MRCPv1

```
<include>
   <profile name="ivona16-mrcp1" version="1">
     <param name="server-ip" value="10.2.7.214"/>
     <param name="server-port" value="4900"/>
     <param name="resource-location" value="media"/>
     <param name="speechsynth" value="speechsynthesizer"/>
     <param name="speechrecog" value="speechrecognizer"/>
     <param name="rtp-ip" value="10.10.5.2"/>
     <param name="rtp-port-min" value="4000"/>
     <param name="rtp-port-max" value="5000"/>
     <param name="rtcp" value="1"/>
     <param name="rtcp-bye" value="2"/>
     <param name="rtcp-tx-interval" value="5000"/>
     <param name="rtcp-rx-resolution" value="1000"/>
     <param name="playout-delay" value="100"/>
     <param name="max-playout-delay" value="200"/>
     <param name="ptime" value="20"/>
     <param name="codecs" value="PCMA"/>
   </profile>
 </include>
```

Lumenvox

```
<include>
 <profile name="lumenvox" version="2">
  <param name="client-ip" value="FREESWITCH_IP"/>
  <param name="client-port" value="25097"/>
  <param name="server-ip" value="LUMENVOX_IP"/>
  <param name="server-port" value="5060"/>
  <param name="sip-transport" value="udp"/>
  <param name="rtp-ip" value="FREESWITCH_IP"/>
  <param name="rtp-port-min" value="28000"/>
  <param name="rtp-port-max" value="29000"/>
  <param name="codecs" value="PCMU PCMA L16/96/8000 telephone-event/101/8000"/>
  <synthparams>
  </synthparams>
  <recogparams>
      <param name="start-input-timers" value="false"/>
  </recogparams>
 </profile>
</include>
```

Never use any "auto" values for IP addresses in the configuration.

You can use the following extension for testing:

```
<extension name="unimrcp">
<condition field="destination_number" expression="^5$">
  <action application="answer"/>
  <action application="speak" data="unimrcp:lumenvox||FreeSWITCH is awesome"/>
  <action application="sleep" data="500"/>
</condition>
</extension>
```

UniMRCP Server

This is for the flite plugin of unimrcpserver (it assumes that unimrcpserver runs on same server).
The media engine/rtpfactory profile of the unimrcpserver should use the same L16/96/8000 codec.
You will need r1027 of unimcrp. The unimrcp flite plugin will use only voice "awb" and doesn't support SSML or prosody markers

```
<include>
 <profile name="unimrcpserver-mrcp2" version="2">
   <param name="server-ip" value="auto"/>
   <param name="server-port" value="8070"/>
   <param name="resource-location" value=""/>
   <param name="client-ip" value="10.10.5.2"/>
   <param name="client-port" value="5090"/>
   <param name="sip-transport" value="tcp"/>
   <param name="ua-name" value="Freeswitch"/>
   <param name="sdp-origin" value="Freeswitch"/>
   <param name="rtp-ip" value="10.10.5.2"/>
   <param name="rtp-port-min" value="4000"/>
   <param name="rtp-port-max" value="5000"/>
   <param name="codecs" value="L16/96/8000"/>
   <param name="speechsynth" value="flite"/>
 </profile>
 </include>
```

Hint: It's crucial that you provide the correct value for "speechsynth" to make it work! It MUST be the same as specified in unimrcpserver.xml! Otherwise you'll run into issues like not beeing able to return to dialplan: http://code.google.com/p/unimrcp/issues/detail?id=94

Using From FreeSWITCH

When specifying the TTS/ASR engine to use in FreeSWITCH, use

```
unimrcp:profile_name
```

Alternatively, you may use:

```
unimrcp
```

and the default profile specified in unimrcp.conf.xml will be selected.

TTS

mod_unimrcp supports plain text and Speech Synthesis Markup Language (SSML). TTS can be sent using either speak or playback (if prefixed with say:unimrcp:[optional voice]:<TTS text>).

Lua

```
session:set_tts_parms("unimrcp:nuance5-mrcp2", "Donna");
session:speak("{speech-language=en-US,prosody-rate=slow}Hello, from FreeSWITCH.");
```

Perl

```
$session->set_tts_parms("unimrcp:nuance5-mrcp2", "Donna");
$session->speak("{speech-language=en-US,prosody-rate=slow}Hello, from FreeSWITCH.");
```

With SSML added

```
$session->set_tts_parms("unimrcp:nuance5-mrcp2", "Donna");
$session->speak("<?xml >Hello, <emphasis level='strong'>John</emphasis> how are
you?</>");
```

JavaScript

```
session.speak("unimrcp:nuance5-mrcp2", "Donna",
"{speech-language=en-US,prosody-rate=slow}Hello, from FreeSWITCH.");
```

> The string after the : is the desired profile from your config.

C/C++

```
switch_ivr_play_file(session, fh,
"say:unimrcp:Donna:{speech-language=en-US,prosody-rate=slow}Hello, from FreeSWITCH.",
args);
```

```
switch_ivr_play_file(session, fh,
"say:unimrcp:Donna:{speech-language=en-US,prosody-rate=slow}Hello, from FreeSWITCH.",
args);
```

Dialplan XML

```
<extension name="unimrcp">
  <condition field="destination_number" expression="^5$">
    <action application="answer"/>
    <action application="set" data="tts_engine=unimrcp:unimrcpserver-mrcp2"/>
    <action application="set" data="tts_voice=awb"/>
    <action application="speak" data="This is our English text-to-speech system"/>
    <action application="sleep" data="500"/>
  </condition>
</extension>
```

ASR Parameters

MRCP parameters are passed directly to the MRCP server in the **RECOGNIZE** request.

!!! The following parameters are **mod_unimrcp**-specific and **are not passed to the MRCP server**:

- start-recognize: (true|false) If false, loading grammars will not automatically start recognition. Default is true.
- define-grammar: (true|false) If true, DEFINE-GRAMMAR request is sent prior to RECOGNIZE. Default is false.
- name: Can be used to set grammar name in mod_dptools: play_and_detect_speech.

ASR Grammars

ASR grammars may be inline, in a local file, referenced by a URL, built-in, or cached.

mod_unimrcp supports the following grammar formats for inline and local file grammars:

- Speech Recognition Grammar Specification (SRGS) - XML or ABNF notation
- Java Speech Grammar Format (JSGF)
- Nuance Grammar Specification Language (GSL)

Your MRCP server will most likely only support some of these formats - check your MRCP server documentation.

Inline Grammars

Inline grammars are prefixed with "inline:". Inline grammars are cached on the server with DEFINE-GRAMMAR. Subsequent pause/resume calls will reference the grammar by its name.

Grammars in local files

Grammars stored in local files will be loaded and cached on the server with DEFINE-GRAMMAR. All local files must end in the ".gram" extension. If the local files are in the freeswitch/grammars directory, they can be referenced in by file name without the extension. If the file is located in a different directory, the entire path must be specified. For example, /usr/local/freeswitch/grammars/yesno.gram can be referenced as "yesno" or as "/usr/local/freeswitch/grammars/yesno", but not as "yesno.gram" or "/usr/local/freeswitch/grammars/yesno.gram".

Built-in grammars

Built-in grammars are prefixed with "builtin:"

Cached grammars

Cached grammars are referenced by name prefixed with "session:"

Grammar URL

Only http:// and file:// grammar URLs are currently supported.

JavaScript

```
var asr = new SpeechDetect(session, "unimrcp");
```

C/C++

Load grammar

To detect speech the first time during a call:

```
switch_ivr_detect_speech(session, "unimrcp", "yesno", "yesno-name", "", NULL);
```

The channel will hold onto the ASR resource for the duration of the call, or until switch_ivr_stop_detect_speech() is called.

For subsequent detects with the same ASR resource:

```
switch_ivr_detect_speech_load_grammar(session, "yesno", "yesno-name");
```

ASR parameters

To set specific ASR parameters, add these params inside {} before the grammar, like in the example below.

```
switch_ivr_detect_speech(session, "unimrcp", "{variable1=val1,variable2=val2}yesno",
"yesno-name", "", NULL);
```

Start timers

By default, the MRCP server will start timers in the RECOGNIZE request. If you specify start-input-timers=false, then the timers will not start until START-INPUT-TIMERS is sent to the server. This allows the client to notify the server when the prompt has finished and the input timers should start.

FreeSWITCH does not call this function in any higher-level interface, so you must create (but don't open) the switch_asr_handle_t, ah, before calling switch_ivr_detect_speech() and pass that handle to the function. Then, you can use that handle to call switch_core_asr_start_timers().

```
switch_asr_handle_t *ah;
ah = switch_core_session_alloc(session, sizeof(switch_asr_handle_t));
switch_ivr_detect_speech(session, "unimrcp", "{start-input-timers=false}yesno",
"yesno-name", "", ah);
... play some prompts ...
... start of input or prompt has finished ...
switch_core_asr_start_timers(ah);
```

Pause/Resume

Once speech recognition has started, it can be paused and resumed. Pause is useful, because it allows you to stop speech recognition and still keep the resource.

To pause speech recognition:

```
switch_ivr_pause_detect_speech(session);
```

To resume speech recognition with the same grammar:

```
switch_ivr_resume_detect_speech(session);
```

To resume speech recognition with a different grammar:

```
switch_ivr_detect_speech_load_grammar(session, "places", "places-name");
```

> mod_unimrcp will stop any executing speech recognition if switch_ivr_detect_speech_load_grammar() is called. The new grammar is then loaded and speech recognition on the new grammar is started.

See Also

* UniMRCP home page

* RFC4463 (MRCP v1)

* Draft 24 (MRCP v2)
* SSML Specification
* SRGS Specification
* JSGF Specification

IETF Media Resource Control Protocol Version 2 (MRCPv2)

IETF Media Resource Control Protocol Version 1 (MRCPv1)

NLSML - Natural Language Semantics Markup Language for the Speech Interface Framework

SRGS - Speech Recognition Grammar Specification

Speech Processing for IP Networks - book on MRCP, website has good links

mod_dptools: detect_speech

mod_dptools: play_and_detect_speech