

*Powered by Universal Speech Solutions LLC*



# GoVivace SR Plugin

## Usage Guide

---

Revision: 2

Created: June 6, 2019

Last updated: March 21, 2020

Author: Arsen Chaloyan

# Table of Contents

1 Overview.....	4
1.1 Installation .....	4
1.2 Applicable Versions.....	4
2 Supported Features.....	5
2.1 MRCP Methods .....	5
2.2 MRCP Events .....	5
2.3 MRCP Header Fields .....	5
2.4 Grammars.....	6
2.5 Results.....	6
3 Configuration Format.....	7
3.1 Document.....	7
3.2 Pool of GoVivace Servers.....	8
3.3 GoVivace Server .....	8
3.4 Speech and DTMF Input Detector .....	9
3.5 Utterance Manager.....	11
3.6 RDR Manager .....	12
3.7 Monitoring Agent .....	13
3.8 Usage Change Handler .....	14
3.9 Usage Refresh Handler .....	14
3.10 License Server.....	15
4 Configuration Steps .....	17
4.1 Specifying Pool of GoVivace Servers .....	17
4.2 Specifying Recognition Language .....	17
4.3 Specifying Sampling Rate.....	17
4.4 Specifying Speech Input Parameters .....	18
4.5 Specifying DTMF Input Parameters.....	18
4.6 Specifying No-Input and Recognition Timeouts .....	19
4.7 Specifying Vendor-Specific Parameters .....	19
4.8 Maintaining Utterances.....	20
4.9 Maintaining Recognition Details Records .....	21
5 Recognition Grammars and Results.....	22
5.1 Using Built-in Speech Transcription .....	22
5.2 Using Built-in DTMF Grammars.....	23
5.3 Retrieving Results.....	23
6 Monitoring Usage Details .....	24
6.1 Log Usage .....	24

6.2	Update Usage .....	24
6.3	Dump Channels.....	25
7	Usage Examples.....	26
7.1	Gender Identification .....	26
7.2	DTMF Recognition.....	27
8	Sequence Diagram .....	29
9	References.....	30
9.1	GoVivace .....	30
9.2	Specifications.....	30

# 1 Overview

This guide describes how to configure and use the GoVivace Speech Recognition (SR) plugin to the UniMRCP server. The document is intended for users having a certain knowledge of GoVivace Speech APIs and UniMRCP.



## 1.1 Installation

For installation instructions, use one of the guides below.

- RPM Package Installation (Red Hat / Cent OS)
- Deb Package Installation (Debian / Ubuntu)

## 1.2 Applicable Versions

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



UniMRCP 1.6.0 and above

UniMRCP GoVivaceSR Plugin 1.0.0 and above

# 2 Supported Features

This is a brief check list of the features currently supported by the UniMRCP server running with the GoVivaceSR plugin.

## 2.1 MRCP Methods

- ✓ DEFINE-GRAMMAR
- ✓ RECOGNIZE
- ✓ START-INPUT-TIMERS
- ✓ STOP
- ✓ SET-PARAMS
- ✓ GET-PARAMS

## 2.2 MRCP Events

- ✓ RECOGNITION-COMPLETE
- ✓ START-OF-INPUT

## 2.3 MRCP Header Fields

- ✓ Input-Type
- ✓ No-Input-Timeout
- ✓ Recognition-Timeout
- ✓ Waveform-URI
- ✓ Media-Type
- ✓ Completion-Cause
- ✓ Confidence-Threshold
- ✓ Start-Input-Timers
- ✓ DTMF-Interdigit-Timeout
- ✓ DTMF-Term-Timeout
- ✓ DTMF-Term-Char
- ✓ Save-Waveform
- ✓ Speech-Language
- ✓ Cancel-If-Queue
- ✓ Sensitivity-Level

## 2.4 Grammars

- ✓ Built-in speech transcription grammar
- ✓ Built-in/embedded DTMF grammar
- ✓ SRGS XML (limited support)

## 2.5 Results

- ✓ NLSML

## 3 Configuration Format

The configuration file of the GoVivaceSR plugin is located in `/opt/unimrcp/conf/umsgovivacesr.xml`. The configuration file is written in XML.

### 3.1 Document

The root element of the XML document must be `<umsgovivacesr>`.

#### Attributes

Name	Unit	Description
<b>license-file</b>	File path	Specifies the license file. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used.

#### Parent

None.

#### Children

Name	Unit	Description
<b>&lt;server-pool&gt;</b>	String	Specifies a pool of GoVivace servers.
<b>&lt;speech-dtmf-input-detector&gt;</b>	String	Specifies parameters of the speech and DTMF input detector.
<b>&lt;utterance-manager&gt;</b>	String	Specifies parameters of the utterance manager.
<b>&lt;rdr-manager&gt;</b>	String	Specifies parameters of the Recognition Details Record (RDR) manager.
<b>&lt;monitoring-agent&gt;</b>	String	Specifies parameters of the monitoring manager.
<b>&lt;license-server&gt;</b>	String	Specifies parameters used to connect to the license server. The use of the license server is optional.

#### Example

This is an example of a bare document.

```
< umsgovivacesr license-file="umsgovivacesr_*.lic">
</ umsgovivacesr>
```

## 3.2 Pool of GoVivace Servers

This element specifies a pool of GoVivace servers.

### Attributes

None.

### Parent

```
<umsgovivacesr>
```

### Children

```
<server>
```

### Example

This is an example of a pool of GoVivace servers.

```
<server-pool>
  <server name="gender-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:7684" method="GenderId" secret-key="****"/>
  <server name="emotion-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:7687" method="EmotionId" secret-key="****"/>
  <server name="language-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:7686" method="LanguageId" secret-key="****"/>
  <server name="keyword-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:49149" method="telephony" secret-key="****"/>
  <server name="grxml-1" language="en-US" sampling-rate="8000"
    uri="ws://198.199.70.106:49162" method="answer" secret-key="****"/>
  <server name="grxml-2" language="en-US" sampling-rate="8000"
    uri="ws://198.199.70.106:49162" method="location" secret-key="****"/>
  <server name="grxml-3" language="en-US" sampling-rate="8000"
    uri="ws://198.199.70.106:49162" method="duration" secret-key="****"/>
</server-pool>
```

## 3.3 GoVivace Server

This element specifies parameters a GoVivace server.



## Attributes

Name	Unit	Description
<b>name</b>	String	Specifies the name of the server referenced in built-in grammars.
<b>language</b>	String	Specifies the language supported by the server.
<b>sampling-rate</b>	Integer	Specifies the sampling rate supported by the server.
<b>uri</b>	String	Specifies the Service URI of the server.
<b>method</b>	String	Specifies the method supported by the server.
<b>secret-key</b>	String	Specifies a secret key used for authentication to the server.

## Parent

<server-pool>

## Children

None.

## Example

This is an example of a GoVivace server.

```
<server name="gender-1" language="en-US" sampling-rate="8000"
uri="wss://services.govivace.com:7684" method="GenderId" secret-key="****"/>
```

## 3.4 Speech and DTMF Input Detector

This element specifies parameters of the speech and DTMF input detector.

## Attributes

Name	Unit	Description
<b>vad-mode</b>	Integer	Specifies an operating mode of VAD in the

		range of [0 ... 3]. Default is 1.
<b>speech-start-timeout</b>	Time interval [msec]	Specifies how long to wait in transition mode before triggering a start of speech input event.
<b>speech-complete-timeout</b>	Time interval [msec]	Specifies how long to wait in transition mode before triggering an end of speech input event.
<b>noinput-timeout</b>	Time interval [msec]	Specifies how long to wait before triggering a no-input event.
<b>input-timeout</b>	Time interval [msec]	Specifies how long to wait for input to complete.
<b>dtmf-interdigit-timeout</b>	Time interval [msec]	Specifies a DTMF inter-digit timeout.
<b>dtmf-term-timeout</b>	Time interval [msec]	Specifies a DTMF input termination timeout.
<b>dtmf-term-char</b>	Character	Specifies a DTMF input termination character.
<b>speech-leading-silence</b>	Time interval [msec]	Specifies desired silence interval preceding spoken input.
<b>speech-trailing-silence</b>	Time interval [msec]	Specifies desired silence interval following spoken input.
<b>speech-output-period</b>	Time interval [msec]	Specifies an interval used to send speech frames to the recognizer.

## Parent

<umsgovivacesr>

## Children

None.

## Example

The example below defines a typical speech and DTMF input detector having the default parameters set.

```
<speech-dtmf-input-detector
vad-mode="1"
speech-start-timeout="300"
speech-complete-timeout="1000"
noinput-timeout="5000"
```

```

input-timeout="10000"
dtmf-interdigit-timeout="5000"
dtmf-term-timeout="10000"
dtmf-term-char=""
speech-leading-silence="300"
speech-trailing-silence="300"
speech-output-period="200"
/>

```

## 3.5 Utterance Manager

This element specifies parameters of the utterance manager.

### Attributes

Name	Unit	Description
<b>save-waveforms</b>	Boolean	Specifies whether to save waveforms or not.
<b>purge-existing</b>	Boolean	Specifies whether to delete existing records on start-up.
<b>max-file-age</b>	Time interval [min]	Specifies a time interval in minutes after expiration of which a waveform is deleted. Set 0 for infinite.
<b>max-file-count</b>	Integer	Specifies the max number of waveforms to store. If reached, the oldest waveform is deleted. Set 0 for infinite.
<b>waveform-base-uri</b>	String	Specifies the base URI used to compose an absolute waveform URI.
<b>waveform-folder</b>	Dir path	Specifies a folder the waveforms should be stored in.
<b>file-prefix</b>	String	Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsgovivacesr-', if not specified.
<b>use-logging-tag</b>	Boolean	Specifies whether to use the MRCP header field Logging-Tag, if present, to compose the name of the file to be stored. Available since 1.2.0.

### Parent

<umsgovivacesr>

## Children

None.

## Example

The example below defines a typical utterance manager having the default parameters set.

```
<utterance-manager
  save-waveforms="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-base-uri="http://localhost/utterances/"
  waveform-folder=""
/>
```

## 3.6 RDR Manager

This element specifies parameters of the Recognition Details Record (RDR) manager.

### Attributes

Name	Unit	Description
<b>save-records</b>	Boolean	Specifies whether to save recognition details records or not.
<b>purge-existing</b>	Boolean	Specifies whether to delete existing records on start-up.
<b>max-file-age</b>	Time interval [min]	Specifies a time interval in minutes after expiration of which a record is deleted. Set 0 for infinite.
<b>max-file-count</b>	Integer	Specifies the max number of records to store. If reached, the oldest record is deleted. Set 0 for infinite.
<b>record-folder</b>	Dir path	Specifies a folder to store recognition details records in. Defaults to <code>\${UniMRCPIInstallDir}/var</code> .
<b>file-prefix</b>	String	Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsgovivacesr-', if not specified.

<b>use-logging-tag</b>	Boolean	Specifies whether to use the MRCP header field Logging-Tag, if present, to compose the name of the file to be stored. Available since 1.2.0.
------------------------	---------	--

### Parent

<umsgovivacesr>

### Children

None.

### Example

The example below defines a typical utterance manager having the default parameters set.

```
<rdr-manager
  save-records="false"
  purge-existing="false"
  max-file-age="60"
  max-file-count="100"
  waveform-folder=""
/>
```

## 3.7 Monitoring Agent

This element specifies parameters of the monitoring agent.

### Attributes

Name	Unit	Description
<b>refresh-period</b>	Time interval [sec]	Specifies a time interval in seconds used to periodically refresh usage details. See <usage-refresh-handler>.

### Parent

<umsgovivacesr>

### Children

<usage-change-handler>

<usage-refresh-handler>

## Example

The example below defines a monitoring agent with usage change and refresh handlers.

```
<monitoring-agent refresh-period="60">

  <usage-change-handler>
    <log-usage enable="true" priority="NOTICE"/>
  </usage-change-handler>

  <usage-refresh-handler>
    <dump-channels enable="true" status-file="umsgovivacesr-channels.status"/>
  </usage-refresh-handler >

</monitoring-agent>
```

## 3.8 Usage Change Handler

This element specifies an event handler called on every usage change.

### Attributes

None.

### Parent

```
<monitoring-agent>
```

### Children

```
<log-usage>
<update-usage>
<dump-channels>
```

## Example

This is an example of the usage change event handler.

```
<usage-change-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umsgovivacesr-usage.status"/>
  <dump-channels enable="false" status-file="umsgovivacesr-channels.status"/>
</usage-change-handler>
```

## 3.9 Usage Refresh Handler

This element specifies an event handler called periodically to update usage details.

## Attributes

None.

## Parent

<monitoring-agent>

## Children

<log-usage>

<update-usage>

<dump-channels>

## Example

This is an example of the usage change event handler.

```
<usage-refresh-handler>
  <log-usage enable="true" priority="NOTICE"/>
  <update-usage enable="false" status-file="umsgovivacesr-usage.status"/>
  <dump-channels enable="false" status-file="umsgovivacesr-channels.status"/>
</usage-refresh-handler>
```

## 3.10 License Server

This element specifies parameters used to connect to the license server.

### Attributes

Name	Unit	Description
<b>enable</b>	Boolean	Specifies whether the use of license server is enabled or not. If enabled, the license-file attribute is not honored.
<b>server-address</b>	String	Specifies the IP address or host name of the license server.
<b>certificate-file</b>	File path	Specifies the client certificate used to connect to the license server. File name may include patterns containing a '*' sign. If multiple files match the pattern, the most recent one gets used.
<b>ca-file</b>	File path	Specifies the certificate authority used to validate the license server.

<b>channel-count</b>	Integer	Specifies the number of channels to check out from the license server. If not specified or set to 0, either all available channels or a pool of channels will be checked based on the configuration of the license server.
<b>http-proxy-address</b>	String	Specifies the IP address or host name of the HTTP proxy server, if used. Available since 1.2.0.
<b>http-proxy-port</b>	Integer	Specifies the port number of the HTTP proxy server, if used. Available since 1.2.0.

### Parent

<umsgovivacesr>

### Children

None.

### Example

The example below defines a typical configuration which can be used to connect to a license server located, for example, at 10.0.0.1.

```
<license-server
  enable="true"
  server-address="10.0.0.1"
  certificate-file="unilic_client_*.crt"
  ca-file="unilic_ca.crt"
/>
```

For further reference to the license server, visit

<http://unimrcp.org/licserver>



## 4 Configuration Steps

This section outlines common configuration steps.

### 4.1 Specifying Pool of GoVivace Servers

One or more GoVivace servers can be specified in the configuration file *umsgovivacesr.xml*. Each instance of the GoVivace server may process a number of concurrent recognition requests for a given language and sampling rate. Each instance of the GoVivace server may support a distinct speech API such as

- Gender Identification
- Emotion Identification
- Language and Accent Identification
- Keyword Spotting
- GRXML Speech Recognition

In the following example, two GoVivace servers are used: one for gender identification and the other for emotion identification

```
<server-pool>
  <server name="gender-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:7684" method="GenderId" secret-key="****"/>
  <server name="emotion-1" language="en-US" sampling-rate="8000"
    uri="wss://services.govivace.com:7687" method="EmotionId" secret-key="****"/>
</ server-pool>
```

Note that installation and configuration of the GoVivace servers is not covered in this document.

### 4.2 Specifying Recognition Language

Recognition language can be specified by the client per MRCP session by means of the header field *Speech-Language* set in a *SET-PARAMS* or *RECOGNIZE* request.

### 4.3 Specifying Sampling Rate

In order to support audio data sampled at 16 kHz, the corresponding codecs needs to be specified in the configuration file *unimrcpsrserver.xml* under the XML element *<rtp-settings>* as follows.

```
<rtp-settings id="RTP-Settings-1">
  <codecs own-preference="false"> PCMU PCMA L16/96/8000 telephone-event/101/8000
    PCMU/97/16000 PCMA/98/16000 L16/99/16000 telephone-
    event/102/16000</codecs>
```

## 4.4 Specifying Speech Input Parameters

While the default parameters specified for the speech input detector are sufficient for the general use, various parameters can be adjusted to better suit a particular requirement.

- `speech-start-timeout`

This parameter is used to trigger a start of speech input. The shorter is the timeout, the sooner a *START-OF-INPUT* event is delivered to the client. However, a short timeout may also lead to a false positive.

- `speech-complete-timeout`

This parameter is used to trigger an end of speech input. The shorter is the timeout, the shorter is the response time. However, a short timeout may also lead to a false positive.

- `vad-mode`

This parameter is used to specify an operating mode of the Voice Activity Detector (VAD) within an integer range of [0 ... 3]. A higher mode is more aggressive and, as a result, is more restrictive in reporting speech. The parameter can be overridden per MRCP session by setting the header field *Sensitivity-Level* in a *SET-PARAMS* or *RECOGNIZE* request. The following table shows how the *Sensitivity-Level* is mapped to the *vad-mode*.

Sensitivity-Level	Vad-Mode
[0.00 ... 0.25)	0
[0.25 ... 0.50)	1
[0.50 ... 0.75)	2
[0.75 ... 1.00]	3

## 4.5 Specifying DTMF Input Parameters

While the default parameters specified for the DTMF input detector are sufficient for the general use, various parameters can be adjusted to better suit a particular requirement.

- `dtmf-interdigit-timeout`

This parameter is used to set an inter-digit timeout on DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Interdigit-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- `dtmf-term-timeout`

This parameter is used to set a termination timeout on DTMF input and is in effect when `dtmf-term-char` is set and there is a match for an input grammar. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- dtmf-term-char

This parameter is used to set a character terminating DTMF input. The parameter can be overridden per MRCP session by setting the header field *DTMF-Term-Char* in a *SET-PARAMS* or *RECOGNIZE* request.

## 4.6 Specifying No-Input and Recognition Timeouts

- noinput-timeout

This parameter is used to trigger a no-input event. The parameter can be overridden per MRCP session by setting the header field *No-Input-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

- input-timeout

This parameter is used to limit input (recognition) time. The parameter can be overridden per MRCP session by setting the header field *Recognition-Timeout* in a *SET-PARAMS* or *RECOGNIZE* request.

## 4.7 Specifying Vendor-Specific Parameters

The following parameters can optionally be specified by the MRCP client in *SET-PARAMS*, *DEFINE-GRAMMAR* and *RECOGNIZE* requests via the MRCP header field *Vendor-Specific-Parameters*.

Name	Unit	Description
<b>start-of-input</b>	String	Specifies the source of start of input event sent to the client (use "service-originated" for an event originated based on a first-received interim result and "internal" for an event determined by plugin). Available since 1.2.0.
<b>alternatives-below-threshold</b>	Boolean	Specifies whether to return speech recognition result alternatives with the confidence score below the confidence threshold. Available since 1.2.0.
<b>speech-start-timeout</b>	Time interval [msec]	Specifies how long to wait in transition mode before triggering a start of speech input event. Available since 1.2.0.

All the vendor-specific parameters can also be specified at the grammar-level via a built-in or SRGS XML grammar.

The following example demonstrates the use of a built-in grammar with the vendor-specific parameters *alternatives-below-threshold* and *speech-start-timeout* set to *true* and *100* correspondingly.

```
builtin:speech/transcribe?alternatives-below-threshold=true;speech-start-timeout=100
```

The following example demonstrates the use of an SRGS XML grammar with the vendor-specific parameters *alternatives-below-threshold* and *speech-start-timeout* set to *true* and *100* correspondingly.

```
<grammar mode="voice" root="transcribe" version="1.0" xml:lang="en-US"
xmlns="http://www.w3.org/2001/06/grammar">
  <meta name="scope" content="builtin"/>
  <meta name="alternatives-below-threshold" content="true"/>
  <meta name="speech-start-timeout" content="100"/>
  <rule id="transcribe">
    <one-of ><item>blank</item></one-of>
  </rule>
</grammar>
```

All the unknown vendor-specific parameters are transparently passed through and used as query parameters to compose service URI. For example, the following header field

```
Vendor-Specific-Parameters: contextId="arbitrary-string-id-1";traceId="arbitrary-string-id-2"
```

results in the service URI composed as follows

```
$uri?contextId=arbitrary-string-id-1&traceId=arbitrary-string-id-2
```

## 4.8 Maintaining Utterances

Saving of utterances is not required for regular operation and is disabled by default. However, enabling this functionality allows to save utterances sent to the GoVivace server and later listen to them offline.

The relevant settings can be specified via the element *utterance-manager*.

- *save-waveforms*

Utterances can optionally be recorded and stored if the configuration parameter *save-waveforms* is set to true. The parameter can be overridden per MRCP session by setting the header field *Save-Waveforms* in a *SET-PARAMS* or *RECOGNIZE* request.

- *purge-existing*

This parameter specifies whether to delete existing waveforms on start-up.

- *max-file-age*

This parameter specifies a time interval in minutes after expiration of which a waveform is deleted. If set to 0, there is no expiration time specified.

- *max-file-count*

This parameter specifies the maximum number of waveforms to store. If the specified number is reached, the oldest waveform is deleted. If set to 0, there is no limit specified.

- waveform-base-uri

This parameter specifies the base URI used to compose an absolute waveform URI returned in the header field *Waveform-Uri* in response to a RECOGNIZE request.

- waveform-folder

This parameter specifies a path to the directory used to store waveforms in. The directory defaults to *\${UniMRCPInstallDir}/var*.

## 4.9 Maintaining Recognition Details Records

Producing of recognition details records (RDR) is not required for regular operation and is disabled by default. However, enabling this functionality allows to store details of each recognition attempt in a separate file and analyze them later offline. The RDRs are stored in the JSON format.

The relevant settings can be specified via the element *rdr-manager*.

- save-records

This parameter specifies whether to save recognition details records or not.

- purge-existing

This parameter specifies whether to delete existing records on start-up.

- max-file-age

This parameter specifies a time interval in minutes after expiration of which a record is deleted. If set to 0, there is no expiration time specified.

- max-file-count

This parameter specifies the maximum number of records to store. If the specified number is reached, the oldest record is deleted. If set to 0, there is no limit specified.

- record-folder

This parameter specifies a path to the directory used to store records in. The directory defaults to *\${UniMRCPInstallDir}/var*.

# 5 Recognition Grammars and Results

## 5.1 Using Built-in Speech Transcription

For generic speech transcription, having no speech contexts defined, a pre-set identifier *transcribe* must be used by the MRCP client in a RECOGNIZE request as follows:

```
builtin:speech/transcribe
```

The name of the identifier *transcribe* can be changed from the configuration file *umsgovivacesr.xml*, since GoVivaceSR 1.1.0.

Speech contexts are defined in the configuration file *umsgovivacesr.xml* and available since GoVivaceSR 1.2.0. A speech context is assigned a unique string identifier and holds a list of phrases.

Below is a definition of a sample speech context *directory*:

```
<speech-context id="directory" speech-complete="true">
  <phrase>call Steve</phrase>
  <phrase>call John</phrase>
  <phrase>dial 5</phrase>
  <phrase>dial 6</phrase>
</speech-context>
```

Which can be referenced in a RECOGNIZE request as follows:

```
builtin:speech/directory
```

The prefixes *builtin:speech* and *builtin:grammar* can be used interchangeably as follows:

```
builtin:grammar/directory
```

Since GoVivaceSR 1.2.0, a speech context can be referenced by means metadata in SRGS XML grammar. For example, the following SRGS grammar references a built-in speech context *directory*.

```
<grammar mode="voice" root="directory" version="1.0"
  xml:lang="en-US"
  xmlns="http://www.w3.org/2001/06/grammar">
  <meta name="scope" content="builtin">
  <rule id="directory"><one-of/></rule>
```

```
</grammar>
```

Where the root rule name identifies a speech context.

## 5.2 Using Built-in DTMF Grammars

Pre-set built-in DTMF grammars can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:dtmf/$id
```

Where *\$id* is a unique string identifier of the built-in DTMF grammar.

Note that only a DTMF grammar identifier *digits* is currently supported.

Since GoVivaceSR 1.2.0, built-in DTMF digits can also be referenced by metadata in SRGS XML grammar. The following example is equivalent to the built-in grammar above.

```
<grammar mode="dtmf" root="digits" version="1.0"  
  xml:lang="en-US"  
  xmlns="http://www.w3.org/2001/06/grammar">  
  <meta name="scope" content="builtin"/>  
  <rule id="digits"><one-of/></rule>  
</grammar>
```

Where the root rule name identifies a built-in DTMF grammar.

## 5.3 Retrieving Results

Results received from the GoVivace server are transformed to the [NLSML](#) format.

# 6 Monitoring Usage Details

The number of in-use and total licensed channels can be monitored in several alternate ways. There is a set of actions which can take place on certain events. The behavior is configurable via the element *monitoring-agent*, which contains two event handlers: *usage-change-handler* and *usage-refresh-handler*.

While the *usage-change-handler* is invoked on every acquisition and release of a licensed channel, the *usage-refresh-handler* is invoked periodically on expiration of a timeout specified by the attribute *refresh-period*.

The following actions can be specified for either of the two handlers.

## 6.1 Log Usage

The action *log-usage* logs the following data in the order specified.

- The number of currently in-use channels.
- The maximum number of channels used concurrently.
- The total number of licensed channels.

The following is a sample log statement, indicating 0 in-use, 0 max-used and 2 total channels.

```
[NOTICE] GoVivaceSR Usage: 0/0/2
```

## 6.2 Update Usage

The action *update-usage* writes the following data to a status file *umsgovivacesr-usage.status*, located by default in the directory `${UniMRCPIInstallDir}/var/status`.

- The number of currently in-use channels.
- The maximum number of channels used concurrently.
- The total number of licensed channels.
- The current status of the license permit.
- The license server alarm. Set to *on*, if the license server is not available for more than one hour; otherwise, set to *off*. This parameter is maintained only if the license server is used.

The following is a sample content of the status file.

```
in-use channels: 0  
max used channels: 0
```



```
total channels: 2  
license permit: true  
licserver alarm: off
```

## 6.3 Dump Channels

The action *dump-channels* writes the identifiers of in-use channels to a status file *umsgovivacesr-channels.status*, located by default in the directory `${UniMRCPIInstallDir}/var/status`.

# 7 Usage Examples

## 7.1 Gender Identification

This examples demonstrates how to perform speech recognition by using a RECOGNIZE request.

C->S:

```
MRCP/2.0 336 RECOGNIZE 1
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 5000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Save-Waveform: true
Content-Length: 24

builtin:speech/GenderId
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 498 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 6e1a2e4e54ae11e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/utter-6e1a2e4e54ae11e7-1.wav>;size=20480;duration=1280
Content-Type: application/x-nlsml
```

Content-Length: 345

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/GenderId" confidence="0.95">
    <instance>
      <status>0</status>
      <message>Gender identification is successful</message>
      <gender>male</gender>
      <string-confidence>0.952294</string-confidence>
      <processing-time>1.052527</processing-time>
      <input-speech-duration>0.810000</input-speech-duration>
    </instance>
    <input mode="speech"></input>
  </interpretation>
</result>
```

## 7.2 DTMF Recognition

This examples demonstrates how to reference a built-in DTMF grammar in a RECOGNIZE request.

C->S:

```
MRCP/2.0 266 RECOGNIZE 1
Channel-Identifier: d26bef74091a174c@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
Confidence-Threshold: 0.7
Speech-Language: en-US
Dtmf-Term-Char: #
Content-Length: 19

builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: d26bef74091a174c@speechrecog
```

S->C:

```
MRCP/2.0 113 START-OF-INPUT 1 IN-PROGRESS
```

Channel-Identifier: d26bef74091a174c@speechrecog  
Input-Type: dtmf

S->C:

MRCP/2.0 382 RECOGNITION-COMPLETE 1 COMPLETE

Channel-Identifier: d26bef74091a174c@speechrecog

Completion-Cause: 000 success

Content-Type: application/x-nlsml

Content-Length: 197

```
<?xml version="1.0"?>
```

```
<result>
```

```
  <interpretation grammar="builtin:dtmf/digits" confidence="1.00">
```

```
    <input mode="dtmf">1 2 3 4</input>
```

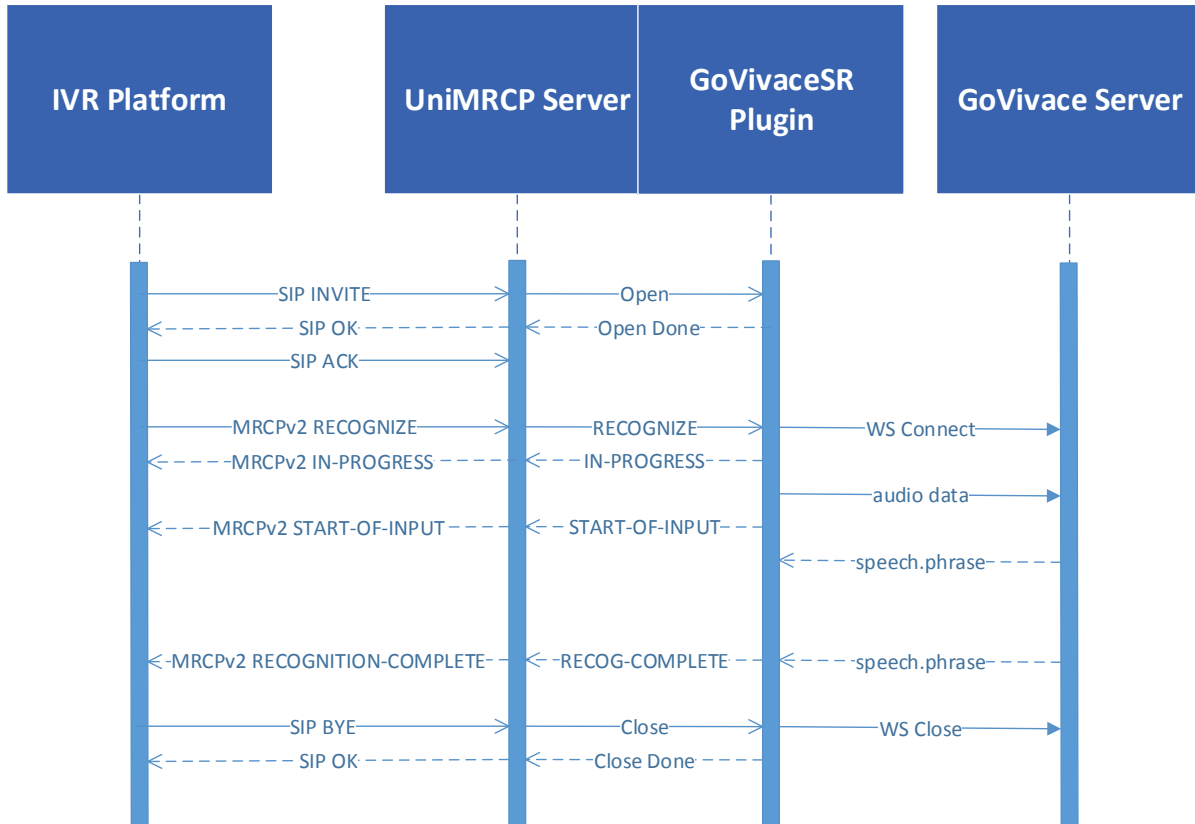
```
    <instance>1234</instance>
```

```
  </interpretation>
```

```
</result>
```

# 8 Sequence Diagram

The following sequence diagram outlines common interactions between all the main components involved in a typical recognition session performed over MRCPv2.



# 9 References

## 9.1 GoVivace

- [www.govivace.com](http://www.govivace.com)

## 9.2 Specifications

- [Speech Recognizer Resource](#)
- [NLSML Results](#)