*Powered by Universal Speech Solutions LLC*

UniMRCP

# Google SS Plugin

## Usage Guide

Revision: 10

Created: May 24, 2018

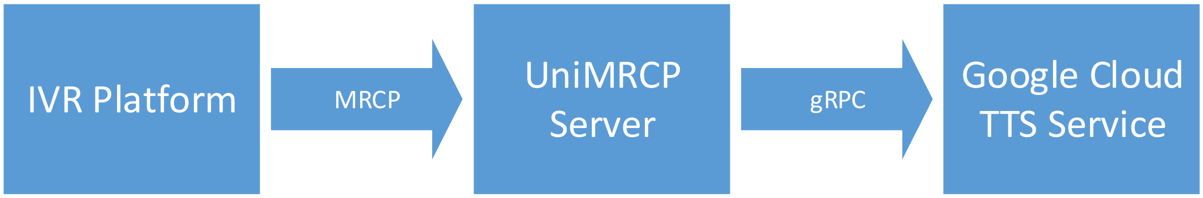
Last updated: March 8, 2021

Author: Arsen Chaloyan

# Table of Contents

# 1 Overview

This guide describes how to configure and use the Google Speech Synthesis (GSS) plugin to the UniMRCP server. The document is intended for users having a certain knowledge of Google Cloud Speech Platform 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.

> ℹ UniMRCP 1.5.0 and above
>
> UniMRCP GSS 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 GSS plugin.

## 2.1 MRCP Methods

- ✓ SPEAK
- ✓ STOP
- ✓ PAUSE
- ✓ RESUME
- ✓ BARGE-IN-OCCURRED
- ✓ SET-PARAMS
- ✓ GET-PARAMS

## 2.2 MRCP Events

- ✓ SPEECH-MARKER
- ✓ SPEAK-COMPLETE

## 2.3 MRCP Header Fields

- ✓ Kill-On-Barge-In
- ✓ Completion-Cause
- ✓ Voice-Gender
- ✓ Voice-Name
- ✓ Prosody-Rate
- ✓ Prosody-Volume
- ✓ Speech-Language
- ✓ Logging-Tag
- ✓ Cache-Control

## 2.4 Speech Data

- ✓ Plain text (text/plain)
- ✓ SSML (application/ssml+xml or application/synthesis+ssml)

# 3 Supported Voices

All the voices supported by Google Text-to-Speech API are listed in the following page:

https://cloud.google.com/text-to-speech/docs/voices

# 4 Configuration Format

The configuration file of the GSS plugin is located in */opt/unimrcp/conf/umsgss.xml*. The configuration file is written in XML.

## 4.1 Document

The root element of the XML document must be <*umsgss*>.

### Attributes

| Name | Unit | Description |
|---|---|---|
| **license-file** | 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. |
| **gapp-credentials-file** | File path | Specifies the Google Application Credentials file to use. 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 |
|---|---|---|
| **<synth-settings>** | String | Specifies synthesis parameters employed via gRPC. |
| **<waveform-manager>** | String | Specifies parameters of the waveform manager. Available since GSS 1.2.0. |
| **<sdr-manager>** | String | Specifies parameters of the Synthesis Details Record (SDR) manager. Available since GSS 1.2.0. |
| **<monitoring-agent>** | String | Specifies parameters of the monitoring manager. |
| **<license-server>** | 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.

```
        < umsgss license-file="umsgss_*.lic" gapp-credentials-file="*.json">
        </ umsgss>
```

## 4.2 Synthesis Settings

This element specifies synthesis parameters.

### Attributes

| Name | Unit | Description |
|---|---|---|
| **language** | String | Specifies the default language to use, if not set by the client. |
| **bypass-ssml** | Boolean | Specifies whether to transparently bypass or parse received content in order to determine voice parameters set in SSML. Available since GSS 1.1.0. |
| **normalize-ssml** | Boolean | Specifies whether to normalize SSML. The parameter is observed only when *bypass-ssml* is set to *false*. Available since GSS 1.4.0. |
| **voice-name** | String | Specifies the default voice name. Can be overridden by client. Available since GSS 1.1.0. |
| **voice-gender** | String | Specifies the default voice gender. Can be overridden by client. Available since GSS 1.1.0. |
| **effects-profile** | String | Specifies the audio effects profile identifier.<br>https://cloud.google.com/text-to-speech/docs/audio-profiles<br>Available since GSS 1.7.0/ |
| **http-proxy** | String | Specifies the URI of HTTP proxy, if used. Available since GSS 1.4.0. |
| **grpc-log-redirection** | Boolean | Specifies whether to enable gRPC log redirection. Available since GSS 1.6.0. |
| **grpc-log-verbosity** | String | Specifies gRPC logging verbosity. One of DEBUG, INFO, ERROR. See GRPC_VERBOSITY for more info. |

| | | |
|---|---|---|
| | | Available since GSS 1.6.0. |
| **grpc-log-trace** | String | Specifies a comma separated list of tracers producing gRPC logs. Use 'all' to turn all tracers on. See GRPC_TRACE for more info. Available since GSS 1.6.0. |
| **caching** | Boolean | Specifies whether to enable caching of synthesized waveforms. Available since GSS 1.8.0. |
| **prosody-rate** | Double | Specifies the default prosody rate (speaking_rate) in the range [0.25, 4.0]. Can be overridden by client. Available since GSS 1.9.0. |
| **prosody-volume** | Double | Specifies the default prosody volume (volume_gain_db) in the range [-96.0, 16.0]. Can be overridden by client. Available since GSS 1.9.0. |
| **prosody-pitch** | Double | Specifies the default prosody pitch in the range [-20.0, 20.0]. Available since GSS 1.9.0. |
| **deadline** | Time interval [msec] | Specifies the gRPC call deadline. Defaults to 0 (disabled). Available since GSS 1.10.0. |
| **reattempt** | Boolean | Specifies whether to reattempt the gRPC call if the original attempt fails. Disabled by default. Available since GSS 1.10.0. |
| **service-uri** | String | Specifies the service endpoint and defaults to texttospeech.googleapis.com:443. Available since GSS 1.11.0. |

### Parent

&lt;umsgss&gt;

### Children

None.

### Example

This is an example of synthesis parameters.

```
<synth-settings
  language="en-US"
  bypass-ssml="true"
  normalize-ssml="true"
  voice-name=""
  voice-gender=""
```

```
        />
```

## 4.3 Waveform Manager

This element specifies parameters of the waveform manager.

### Availability

>= GSS 1.2.0.

### Attributes

| Name | Unit | Description |
|---|---|---|
| **save-waveforms** | Boolean | Specifies whether to save waveforms or not. |
| **purge-existing** | Boolean | Specifies whether to delete existing records on start-up. |
| **max-file-age** | Time interval [min] | Specifies a time interval in minutes after expiration of which a waveform is deleted. Set 0 for infinite. |
| **max-file-count** | Integer | Specifies the max number of waveforms to store. If reached, the oldest waveform is deleted. Set 0 for infinite. |
| **waveform-folder** | Dir path | Specifies a folder the waveforms should be stored in. |
| **file-prefix** | String | Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsgss-', if not specified. |
| **use-logging-tag** | 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 GSS 1.6.0. |

### Parent

<umsgss>

### Children

None.

### Example

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

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

## 4.4 SDR Manager

This element specifies parameters of the Synthesis Details Record (SDR) manager.

### Availability

>= GSS 1.2.0.

### Attributes

| Name | Unit | Description |
|---|---|---|
| **save-records** | Boolean | Specifies whether to save recognition details records or not. |
| **purge-existing** | Boolean | Specifies whether to delete existing records on start-up. |
| **max-file-age** | Time interval [min] | Specifies a time interval in minutes after expiration of which a record is deleted. Set 0 for infinite. |
| **max-file-count** | Integer | Specifies the max number of records to store. If reached, the oldest record is deleted. Set 0 for infinite. |
| **record-folder** | Dir path | Specifies a folder to store recognition details records in. Defaults to ${UniMRCPInstallDir}/var. |
| **file-prefix** | String | Specifies a prefix used to compose the name of the file to be stored. Defaults to 'umsgss-', if not specified. |
| **use-logging-tag** | 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 GSS 1.6.0. |

### Parent

<umsgss>

### Children

None.

### Example

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

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

## 4.5 Monitoring Agent

This element specifies parameters of the monitoring agent.

### Attributes

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

### Parent

<umsgss>

### 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="umsgss-channels.status"/>
  </usage-refresh-handler >

</monitoring-agent>
```

## 4.6 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="umsgss-usage.status"/>
  <dump-channels enable="false" status-file="umsgss-channels.status"/>
</usage-change-handler>
```

## 4.7 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="umsgss-usage.status"/>
  <dump-channels enable="false" status-file="umsgss-channels.status"/>
</usage-refresh-handler>
```

## 4.8 License Server

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

### Attributes

| Name | Unit | Description |
|---|---|---|
| **enable** | Boolean | Specifies whether the use of license server is enabled or not. If enabled, the license-file attribute is not honored. |
| **server-address** | String | Specifies the IP address or host name of the license server. |
| **certificate-file** | 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. |
| **ca-file** | File path | Specifies the certificate authority used to validate the license server. |
| **channel-count** | 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. |
| **http-proxy-address** | String | Specifies the IP address or host name of the HTTP proxy server, if used. Available since GSS 1.6.0. |
| **http-proxy-port** | Integer | Specifies the port number of the HTTP proxy server, if used. Available since GSS 1.6.0. |

### Parent

<umsgss>

### 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

# 5 Configuration Steps

This section outlines common configuration steps.

## 5.1 Using Default Configuration

The default configuration should be sufficient for the general use.

## 5.2 Specifying Synthesis Language

Synthesis language can be specified by the client per MRCP session by means of the header field *Speech-Language* set in a *SET-PARAMS* or *SPEAK* request, or inline in the SSML data. Otherwise, the parameter *language* set in the configuration file *umsgss.xml* is used. The parameter defaults to *en-US*.

## 5.3 Specifying Sampling Rate

Sampling rate is determined based on the SDP negotiation. Refer to the configuration guide of the UniMRCP server on how to specify supported encodings and sampling rates to be used in communication between the client and server. Either 8 or 16 kHz can be used by Google Cloud Text-to-Speech API for synthesis.

## 5.4 Specifying Voice Parameters

**Global Settings**

The default voice name and gender can be specified from the configuration file *umsgss.xml* using the *voice-name* and *voice-gender* attributes of the *synth-settings* element. This functionality is available since GSS 1.1.0 release.

**MRCP Header Fields**

The voice name and gender can be specified by the MRCP client in *SET-PARAMS* and *SPEAK* requests.

- Voice-Name

  This is an optional parameter indicating the name of the voice to use for synthesis.

- Voice-Gender

  This is an optional parameter indicating the preferred gender of the voice to use for synthesis, which can be set to either *male* or *female* or *neutral*.

**SSML Content**

The voice name and gender can also be specified using the corresponding attributes of the *voice* element in SSML content. In order to parse and determine the parameters and pass them forward to Google Text-to-Speech API accordingly, the *bypass-ssml* attribute of the *synth-settings* element must be set to *false* in the configuration file *umsgss.xml*. This functionality is available since GSS 1.1.0 release.

Since GSS 1.1.0 release, if the *bypass-ssml* attribute is set to *false* and the *normalize-ssml* attribute is set to *true*, then the *voice* element, if present, is stripped off from the SSML content passed to the service in order to conform to the subset of SSML supported by Google Text-to-Speech API.

## 5.5 Specifying Prosody Parameters

The following prosody parameters can be specified by the MRCP client in *SET-PARAMS* and *SPEAK* requests.

- Prosody-Rate

  This is an optional parameter indicating the speaking rate, which can be set to one of the following labels: *x-slow***,** *slow***,** *medium***,** *fast***,** *x-fast***,** *default***.**

- Prosody-Volume

  This is an optional parameter indicating the speaking volume, which can be set to one of the following labels: *silent*, *x-soft*, *soft*, *medium*, *loud*, *x-loud*, *default*.

## 5.6 Specifying Speech Data

Speech data can be specified by the MRCP client in *SPEAK* requests using one of the following content types:

- plain/text
- application/ssml+xml (or application/synthesis+ssml)

## 5.7 Maintaining Waveforms

Collection of waveforms is not required for regular operation and is disabled by default. However, enabling this functionality allows to save synthesized speech received from the Google Cloud Speech service and later listen to them offline.

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

- save-waveforms

  Utterances can optionally be recorded and stored if the configuration parameter *save-waveforms* is set to true.

- 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-folder

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

## 5.8 Maintaining Synthesis Details Records

Collection of synthesis details records (SDR) is not required for regular operation and is disabled by default. However, enabling this functionality allows to store details of each synthesis attempt in a separate file and analyze them later offline. The SDRs ate stored in the JSON format.

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

- save-records

This parameter specifies whether to save synthesis 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*.

## 5.9 Using Cache

Since GSS 1.8.0, synthesized waveforms can be stored and re-used for consecutive speech synthesis requests, when applicable. In order to use this functionality, the attribute *caching* of the element *synth-settings* must be set to *true*. The attribute defaults to *false*.

The lifetime and size of cached records are controlled by the attributes *max-file-age* and *max-file-count* of the element *waveform-manager*.

The cached records are persistent and populated on initial loading, unless the attribute *purge-existing* of the element *waveform-manager* is set to *true*.

The following speech synthesis parameters are observed while searching for a cached record.

- language
- voice-name
- voice-gender
- sampling-rate
- prosody-rate
- prosody-volume
- content

The following cache control directives are observed while searching for a cached record.

- max-age
- min-fresh

The cache control directives can be specified by the client per individual speech synthesis request via the MRCP header field *Cache-Control*. By default, no cache control directives are applied.

# 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. Available since GSS 1.2.0.
- 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] GSS Usage: 0/0/2
```

## 6.2 Update Usage

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

- The number of currently in-use channels.
- The maximum number of channels used concurrently. Available since GSS 1.2.0.
- 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. Available since GSS 1.4.0.

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 *umsgss-channels.status*, located by default in the directory *${UniMRCPInstallDir}/var/status*.

# 7 Usage Examples

## 7.1 SSML

This examples demonstrates how to perform speech synthesis by using a SPEAK request with an SSML content.

C->S:

```
MRCP/2.0 309 SPEAK 1
Channel-Identifier: 4dde51f37d1a9546@speechsynth
Content-Type: application/ssml+xml
Voice-Age: 28
Content-Length: 163

<?xml version="1.0"?>
<speak version="1.0" xml:lang="en-US" xmlns="http://www.w3.org/2001/10/synthesis">
  <p>
    <s>Welcome to Uni MRCP.</s>
  </p>
</speak>
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 4dde51f37d1a9546@speechsynth
```

S->C:

```
MRCP/2.0 122 SPEAK-COMPLETE 1 COMPLETE
Channel-Identifier: 4dde51f37d1a9546@speechsynth
Completion-Cause: 000 normal
```

## 7.2 Plain Text

This example demonstrates how to perform speech synthesis by using a SPEAK request with a plain text content.

C->S:

```
MRCP/2.0 155 SPEAK 1
Channel-Identifier: 85667d0efbf95345@speechsynth
Content-Type: text/plain
Voice-Age: 28
Content-Length: 20

Welcome to Uni MRCP.
```

S->C:

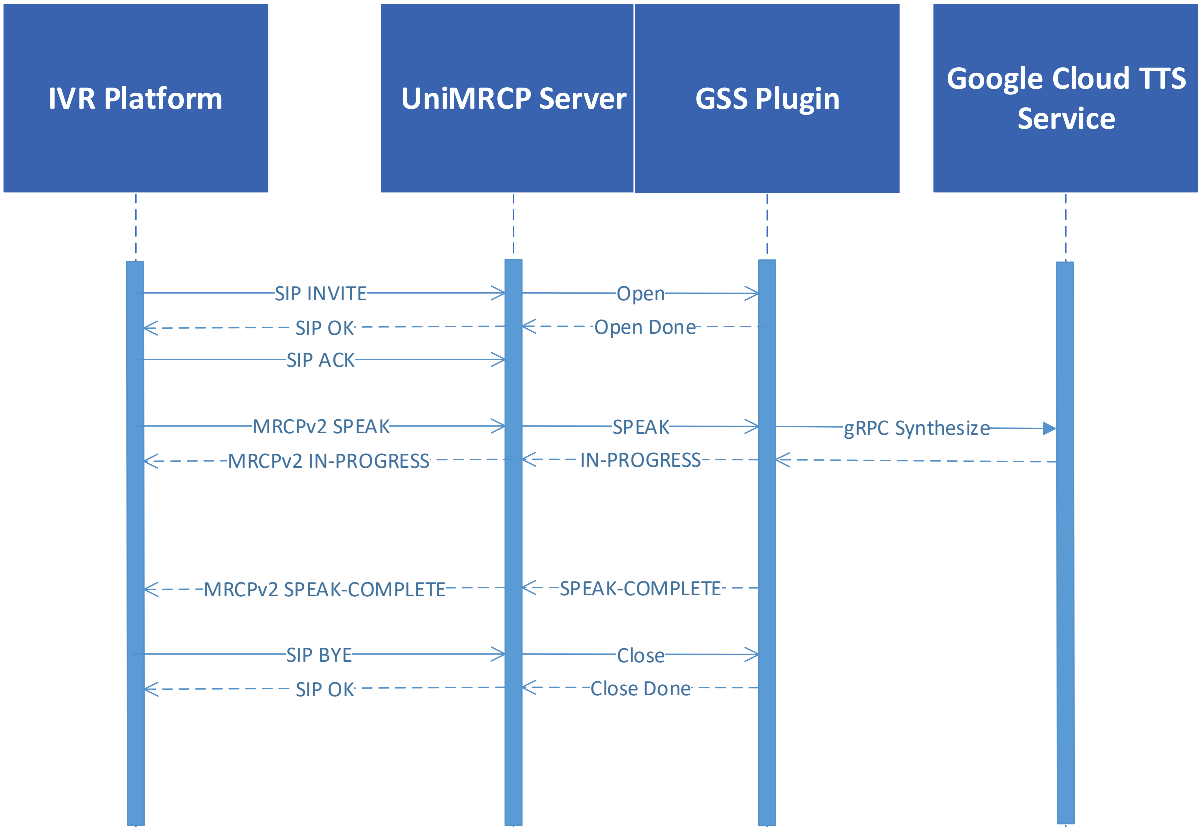
```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 85667d0efbf95345@speechsynth
```

S->C:

```
MRCP/2.0 122 SPEAK-COMPLETE 1 COMPLETE
Channel-Identifier: 85667d0efbf95345@speechsynth
Completion-Cause: 000 normal
```

# 8 Sequence Diagram

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



# 9 References

## 9.1 Google Cloud Platform

- Text-to-Speech API
- How-to Guides
- Best Practices

## 9.2 Specifications

- Speech Synthesizer Resource
- SSML