*Powered by Universal Speech Solutions LLC*

UniMRCP

# AWS Lex Plugin

## Usage Guide

Revision: 8

Created: October 15, 2018

Last updated: February 26, 2021

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 Streaming Recognition 8
3.3 Results 10
3.4 Utterance Manager 11
3.5 RDR Manager 12
3.6 Monitoring Agent 13
3.7 Usage Change Handler 14
3.8 Usage Refresh Handler 14
3.9 License Server 15

4 Configuration Steps 17

4.1 Using Default Configuration 17
4.2 Using with Polly 17
4.3 Specifying AWS Credentials 17
4.4 Specifying Lex Bot 19
4.5 Eliciting Intent 19
4.6 Triggering Intent 20
4.7 Eliciting Slot 20
4.8 Specifying Recognition Language 20
4.9 Specifying Sampling Rate 20
4.10 Specifying Speech Input Parameters 21
4.11 Specifying DTMF Input Parameters 21
4.12 Specifying No-Input and Recognition Timeouts 22
4.13 Maintaining Utterances 22
4.14 Maintaining Recognition Details Records 23

5 Recognition Grammars and Results 24

5.1 Using Built-in Speech Grammar 24
5.2 Using Built-in DTMF Grammars 24

- 5.3 Retrieving Results ..... 24

6 Monitoring Usage Details ..... 26

- 6.1 Log Usage ..... 26
- 6.2 Update Usage ..... 26
- 6.3 Dump Channels ..... 27

7 Usage Examples ..... 28

- 7.1 Hotel Room Booking ..... 28

8 Sequence Diagrams ..... 50

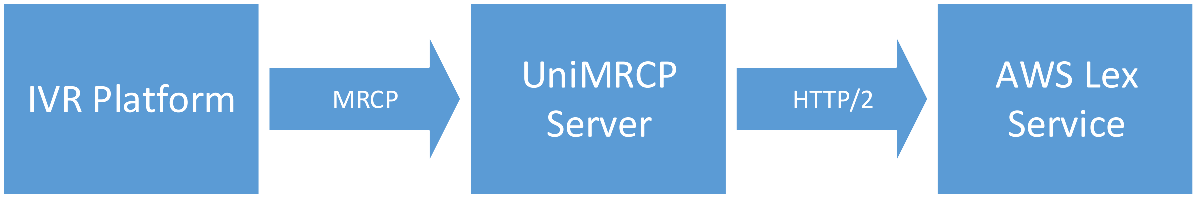
- 8.1 MRCPv1 ..... 50
- 8.2 MRCPv2 ..... 51

9 References ..... 52

- 9.1 AWS Lex ..... 52
- 9.2 Specifications ..... 52

# 1 Overview

This guide describes how to configure and use the Amazon Web Services (AWS) Lex plugin to the UniMRCP server for the use with Lex V2 API. The document is intended for users having a certain knowledge of AWS Lex 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 1.7.0 and above

UniMRCP Lex Plugin 2.0.0 and above

# 2 Supported Features

This is a brief check list of the features currently supported by the UniMRCP server running with the Lex 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
- ✓ Speech-Complete-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, event and DTMF grammars
- ✓ SRGS XML (limited support)

## 2.5 Results

- ✓ NLSML
- ✓ JSON

# 3 Configuration Format

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

## 3.1 Document

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

**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. |
| **credentials-file** | File path | Specifies the AWS credentials file to use. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used. |
| **credentials-provider** | String | Specifies a credentials provider. If not initialized or set to *custom*, the custom credentials provider is used to read credentials from *credentials-file*. Otherwise, if set to *default*, the AWS default credentials provider chain is used, and *credentials-file* is not observer.<br>Otherwise, if set to *sts*, the AWS STS profile credentials provider is used, and *credentials-file* is not observer. |
| **init-sdk** | Boolean | Specifies whether to initialize AWS SDK upon loading of the plugin. Must be set to true by default. Set it to false, if another plugin using the same AWS SDK is loaded prior to this plugin. |
| **shutdown-sdk** | Boolean | Specifies whether to shut down AWS SDK upon unloading of the plugin. Must be set to true by default. Set it to false, if another plugin using the same AWS SDK is unloaded next to this plugin. |
| **sdk-log-level** | Integer | Specifies a log level of AWS SDK. If not initialized or set to 0, the SDK logs are disabled. Acceptable values are from 0 (OFF) to 6 |

| | | |
|---|---|---|
| | | (TRACE). |

**Parent**

None.

**Children**

| Name | Unit | Description |
|---|---|---|
| **<streaming-recognition>** | String | Specifies recognition parameters of streaming recognition. |
| **<results>** | String | Specifies parameters of recognition results set in RECOGNITION-COMPLETE events. |
| **<speech-dtmf-input-detector>** | String | Specifies parameters of the speech and DTMF input detector. |
| **<utterance-manager>** | String | Specifies parameters of the utterance manager. |
| **<rdr-manager>** | String | Specifies parameters of the Recognition Details Record (RDR) manager. |
| **<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.

```
<umslex license-file="umslex_*.lic" credentials-file="*.json" credentials-provider="custom"
init-sdk="true" shutdown-sdk="true">
</umslex>
```

## 3.2 Streaming Recognition

This element specifies parameters of streaming recognition.

**Attributes**

| Name | Unit | Description |
|---|---|---|
| **language** | String | Specifies the default language to use, if not set by the client. |
| **skip-unsupported-grammars** | Boolean | Specifies whether to skip or raise an error while referencing a malformed or not supported grammar. |
| **transcription-grammar** | String | Specifies the name of the built-in speech transcription grammar. The grammar can be referenced as *builtin:speech/transcribe* or *builtin:grammar/transcribe*, where *transcribe* is the default value of this parameter. |
| **region** | String | Specifies the AWS region of the Lex Bot.<br>https://docs.aws.amazon.com/general/latest/gr/lex.html |
| **bot-name** | String | Specifies the name of the Lex bot identifier. Can be overridden by client via Vendor-Specific-Parameters. |
| **alias** | String | Specifies the alias identifier of the Lex bot. Can be overridden by client via Vendor-Specific-Parameters. |
| **proxy-host** | String | Specifies the host name of HTTP proxy, if used. |
| **proxy-port** | Integer | Specifies the port number of HTTP proxy, if used. |
| **proxy-username** | String | Specifies the username employed for HTTP proxy authentication, if used. |
| **proxy-password** | String | Specifies the password employed for HTTP proxy authentication, if used. |

### Parent

<umslex>

### Children

None.

### Example

This is an example of streaming recognition element.

```
<streaming-recognition
   language="en-US"
   skip-unsupported-grammars="true"
   transcription-grammar="transcribe"
   region="us-west-2"
```

```
      bot-name="BookTrip"
      alias="Dev"
    />
```

## 3.3 Results

This element specifies parameters of recognition results set in RECOGNITION-COMPLETE events.

### Attributes

| Name | Unit | Description |
|---|---|---|
| **format** | String | Specifies the format of results to be returned to the client (use "standard" for NLSML and "json" for JSON). |
| **indent** | Integer | Specifies the indent to use while composing the results. |
| **confidence-format** | String | Specifies the format of the confidence score to be returned (use "auto" for a format based on protocol version, "mrcpv2" for a float value in the range of 0..1, "mrcpv1" for an integer value in the range of 0..100) |
| **tag-format** | String | Specifies the format of the instance element to be returned. The parameter is observed only if the format is set to *standard*. Use one of:<br>• *semantics/json* for query result represented in JSON [default]<br>• *swi-semantics/json* for query result set in an inner <SWI_meaning> element represented in JSON |

### Parent

<umslex>

### Children

None.

### Example

This is an example of results element.

```
    <results
      format="standard"
      indent="0"
      confidence-format="auto"
```

```
          />
```

## 3.4 Utterance Manager

This element specifies parameters of the utterance manager.

### 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-base-uri** | String | Specifies the base URI used to compose an absolute waveform URI. |
| **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 'umslex-', 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. |

### Parent

<umslex>

### 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=""
/>
```

## 3.5 RDR Manager

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

### 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 'umslex-', 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. |

### Parent

<umslex>

**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=""
/>
```

## 3.6 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**

<umslex>

**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="umslex-channels.status"/>
  </usage-refresh-handler >

</monitoring-agent>
```

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

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

## 3.9 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. |

| | | |
|---|---|---|
| **http-proxy-port** | Integer | Specifies the port number of the HTTP proxy server, if used. |

### Parent

<umslex>

### 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](http://unimrcp.org/licserver)

# 4 Configuration Steps

This section outlines common configuration steps.

## 4.1 Using Default Configuration

The default configuration should be sufficient for the general use.

## 4.2 Using with Polly

This section must be skipped if the Lex plugin is used without the Polly plugin. However, in case both Polly and Lex plugins are loaded into the same instance of UniMRCP server, then the plugins need to be configured in a certain way to ensure the AWS SDK is initialized and shutdown only once.

In umspolly.xml, set *init-sdk* to *true* and *shutdown-sdk* to *false*.

```
<umspolly license-file="umspolly_*.lic" credentials-file="aws.credentials" init-sdk="true"
shutdown-sdk="false">
```

In umslex.xml, set *init-sdk* to *false* and *shutdown-sdk* to *true*.

```
<umslex license-file="umslex_*.lic" credentials-file="aws.credentials" init-sdk="false"
shutdown-sdk="true">
```

## 4.3 Specifying AWS Credentials

### IAM User

By default, the plugin uses credentials of an IAM user to consume AWS services. The procedure how to set up the credentials is documented in the installation guide. No further action is required if the IAM user is supposed to be used.

### Default Credentials Provider Chain

Since 1.3.0, the plugin can be configured to use the AWS default credentials provider chain, which in turn allows to derive an IAM role set on an instance UniMRCP server is running on. The behavior is controlled by the configuration attribute *credentials-provider*, which is supposed to be set to *default* to use the default credentials provider chain.

```
<umslex license-file="umslex_*.lic" credentials-file="aws.credentials" credentials-
provider="default" sdk="true" shutdown-sdk="true">
```

Note, if the attribute *credentials-provider* is not set to *custom*, the attribute *credentials-file* is not observed.

### STS Profile Credentials Provider

Since 1.4.0, the plugin can be configured to use the AWS STS profile credentials provider to assume a role. The behavior is controlled by the configuration attribute *credentials-provider*, which is supposed to be set to *sts* to use STSProfileCredentialsProvider.

```
<umslex license-file="umslex_*.lic" credentials-file="aws.credentials" credentials-provider="sts" sdk="true" shutdown-sdk="true">
```

Since 1.5.0, the plugin supports multiple AWS STS credentials profiles. The profiles can be loaded on start-up from configuration and also created on-demand by the MRCP client. Contact support for usage examples.

Since 1.6.0, different AWS credentials profiles can be configured per UniMRCP server profile. The profiles can also be created and referenced on-demand by the MRCP client via the header field Vendor-Specific-Parameters. The following parameters can be specified.

| Name | Unit | Description |
|---|---|---|
| **aws-credentials-file** | File path | Specifies the AWS credentials file to use. File name may include patterns containing '*' sign. If multiple files match the pattern, the most recent one gets used. |
| **aws-credentials-provider** | String | Specifies a credentials provider. Use one of:<br>• *custom* for credentials read from the specified file<br>• *default* for the AWS default credentials provider chain<br>• *sts* for the AWS STS profile credentials provider |
| **aws-credentials-profile** | String | Specifies a credentials profile to reference and/or create. |
| **aws-credentials-duration** | Integer | Specifies a lifetime of the credentials profile to create. |
| **aws-arn-role** | String | Specifies an ARN role. |
| **aws-region** | String | Specifies an AWS region. |

## 4.4 Specifying Lex Bot

The parameters of the default Lex bot *bot-name*, *alias* and *region* can be specified in the configuration file *umslex.xml*. For example:

```
<streaming-recognition
  language="en_US"
  region="us-west-2"
  bot-name="1ONTGGK2U4"
  alias="XYZALIASID"

/>
```

The parameters *bot-name*, *alias* and *region* can also be specified per individual MRCP SET-PARAMS or RECOGNIZE requests via the header field Vendor-Specific-Parameters. For example:

```
Vendor-Specific-Parameters: bot-name=1ONTGGK2U4;alias=XYZALIASID;aws-
region=us-west-2
```

The parameters can also be specified as query input attributes passed to the built-in speech grammar. For example:

```
builtin:speech/transcribe?bot-name=1ONTGGK2U4;alias=XYZALIASID;aws-region=us-
west-2
```

It is also possible to specify the parameters in an SRGS XML grammar by means of predefined metadata. For example:

```
<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="bot-name" content="1ONTGGK2U4"/>
   <meta name="alias" content="XYZALIASID"/>
   <meta name="aws-region" content="us-west-2"/>
   <rule id="main"> <one-of/ > </rule>
</grammar>
```

## 4.5 Eliciting Intent

The user application can start a dialog by specifying a welcome message and letting Lex eliciting the intent. The welcome message can be specified in the first MRCP RECOGNIZE request placed in an MRCP session. For example,

```
builtin:speech/transcribe?message=Welcome to travel agency;dialog-action=ElicitIntent
```

As a result, the RECOGNIZE request will complete as soon as the response from the Lex bot is received without any user interaction.

## 4.6 Triggering Intent

The user application can start a dialog by triggering an intent. The intent name can be specified in the first MRCP RECOGNIZE request placed in an MRCP session. For example,

```
builtin:speech/transcribe?intent-name=BookCar;dialog-action=Delegate
```

As a result, the RECOGNIZE request will complete as soon as the response from the Lex bot is received without any user interaction.

## 4.7 Eliciting Slot

Since 2.1.0, the user application can start a dialog by eliciting a particular slot in an intent. The slot and intent names can be specified in the first MRCP RECOGNIZE request placed in an MRCP session. For example,

```
builtin:speech/transcribe?intent-name=BookCar;slot-name=CarType;dialog-action=ElicitSlot
```

As a result, the RECOGNIZE request will complete as soon as the response from the Lex bot is received without any user interaction.

## 4.8 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. Otherwise, the parameter *language* set in the configuration file *umslex.xml* is used. The parameter defaults to *en_US*.

## 4.9 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.

Note: Lex V2 API currently supports 8kHz only.

## 4.10 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 |

## 4.11 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.

## 4.12 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.

## 4.13 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 Lex service 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*.

## 4.14 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 ate 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 Grammar

A pre-set built-in speech grammar can be referenced by the MRCP client in a RECOGNIZE request as follows:

```
builtin:speech/transcribe
```

## 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/digits
```

As a result, received DTMFs will be posted to the Lex bot for intent detection.

## 5.3 Retrieving Results

Results received from the Lex service are transformed to a certain data structure and sent to the MRCP client in a *RECOGNITION-COMPLETE* event. The way results are composed can be adjusted via the <*results*> element in the configuration file *umslex.xml*.

### NLSML Format

If the *format* attribute is set to *standard*, which is the default setting, then the header filed *Content-Type* is set to *application/x-nlsml* and the body of a RECOGNITION-COMPLETE event is set to an NSLML result composed as follows.

**<input>**

The <*input*> element in an NLSML result is set to the input transcript available in the structure *TranscriptEvent* received from Lex bot.

**<instance>**

By default, the <*instance*> element in an NLSML result contains data structure received within *TranscriptEvent*, *TextResponseEvent* and *IntentResultEvent* from the Lex bot. Since 1.3.0, this format can be adjusted via the *tag-format* attribute, which accepts the following values.

- semantics/json

  The data structure is represented in JSON.

- swi-semantics/json

The data structure is set in an inner <*SWI_meaning*> element being represented in JSON.

### JSON Format

If the *format* attribute is set to *json*, then the header field *Content-Type* is set to *application/json* and the body of a RECOGNITION-COMPLETE event is set to a JSON representation of data structure received within *TranscriptEvent*, *TextResponseEvent* and *IntentResultEvent* from the Lex bot.

# 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] Lex Usage: 0/0/2
```

## 6.2 Update Usage

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

# 7 Usage Examples

## 7.1 Hotel Room Booking

This example demonstrates an MRCP message exchange based on a conversation with the sample BookTrip Lex bot.
The results received from the bot are available in the NLSML instance element in the JSON format. The data is demonstrated in an unquoted form for readability.

**Interaction 0**

Event: Elicit an intent

C->S:

```
MRCP/2.0 325 RECOGNIZE 1
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: true
No-Input-Timeout: 30000
Speech-Complete-Timeout: 1500
Speech-Incomplete-Timeout: 15000
Content-Length: 58

builtin:speech/transcribe?message=Welcome to travel agency
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 1015 RECOGNITION-COMPLETE 1 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 829

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"",
"intentResult": {
        "interpretations":      [],
        "sessionState": {
                "dialogAction": {
                        "type": "ElicitIntent"
                },
                "sessionAttributes":    {
                        "x-amz-lex:channels:platform":  "UniMRCP"
                },
                "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
        },
        "sessionId":    "4c16b190426443d7",
        "eventId":      "RESPONSE-3"
},
"textResponse": {
        "messages":    [{
                        "content":      "Welcome to travel agency",
                        "contentType":  "PlainText"
                }],
        "eventId":      "RESPONSE-4"
}
}
</instance>
<input mode="speech"></input>
</interpretation>
</result>
```

### Interaction 1

Input: Book a room

C->S:

```
MRCP/2.0 314 RECOGNIZE 2
Channel-Identifier: 4c16b190426443d7@speechrecog
```

```
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
Speech-Complete-Timeout: 1500
Speech-Incomplete-Timeout: 15000
No-Input-Timeout: 30000
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 2 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 2054 RECOGNITION-COMPLETE 2 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 1867

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"book a room",
"intentResult": {
        "inputMode":    "Speech",
        "interpretations":      [{
                        "nluConfidence":        {
                                "score":        0.89
                        },
                        "intent":       {
                                "name": "BookHotel",
			"slots":	{
				"CheckInDate":	{
				},
				"Location":	{
				},
				"Nights":	{
				},
				"RoomType":	{
				}
			},
			"state":	"InProgress",
			"confirmationState":	"None"
		}
	}, {
		"intent":	{
			"name": "FallbackIntent",
			"slots":	{
			}
		}
	}, {
		"nluConfidence":	{
			"score":	0.5
		},
		"intent":	{
			"name": "BookCar",
			"slots":	{
				"CarType":	{
				},
				"DriverAge":	{
				},
				"PickUpCity":	{
				},
				"PickUpDate":	{
				},
				"ReturnDate":	{
				}
			}
		}
	}],
"sessionState": {
	"dialogAction": {
		"type": "ElicitSlot",
		"slotToElicit": "Location"
	},
	"intent":	{
		"name": "BookHotel",
		"slots":	{
			"CheckInDate":	{
			},
			"Location":	{
			},
                "Nights":     {
                },
                "RoomType":   {
                }
            },
            "state":      "InProgress",
            "confirmationState":   "None"
        },
        "sessionAttributes":   {
            "x-amz-lex:channels:platform":  "UniMRCP"
        },
        "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
    },
    "sessionId":   "4c16b190426443d7",
    "eventId":     "RESPONSE-6"
},
"textResponse": {
    "messages":   [{
            "content":     "What city will you be staying in?",
            "contentType":  "PlainText"
        }],
    "eventId":     "RESPONSE-7"
}
}
</instance>
<input mode="speech">book a room</input>
</interpretation>
</result>
```

### Interaction 2

Input: San Jose

C->S:

```
MRCP/2.0 314 RECOGNIZE 4
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
Speech-Complete-Timeout: 1500
Speech-Incomplete-Timeout: 15000
No-Input-Timeout: 30000
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 4 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 4 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 2484 RECOGNITION-COMPLETE 4 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 2297

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"san jose",
"intentResult": {
    "inputMode":    "Speech",
    "interpretations":    [{
            "nluConfidence":    {
                "score":    1
            },
            "intent":    {
                "name": "BookHotel",
                "slots":    {
                    "CheckInDate":  {
                    },
                    "Location":    {
                        "value":    {
                            "originalValue":    "san jose",
                            "interpretedValue":    "san jose",
                            "resolvedValues":    []

                        }
                    },
				"Nights":	{
				},
				"RoomType":	{
				}
			},
			"state":	"InProgress",
			"confirmationState":	"None"
		}
	}, {
		"intent":	{
			"name": "FallbackIntent",
			"slots":	{
			}
		}
	}, {
		"nluConfidence":	{
			"score":	0.42
		},
		"intent":	{
			"name": "BookCar",
			"slots":	{
				"CarType":	{
				},
				"DriverAge":	{
				},
				"PickUpCity":	{
					"value":	{
						"originalValue":	"jose",
						"interpretedValue":	"jose",
						"resolvedValues":	[]
					}
				},
				"PickUpDate":	{
				},
				"ReturnDate":	{
				}
			}
		}
	}],
"sessionState": {
	"dialogAction": {
		"type": "ElicitSlot",
		"slotToElicit": "CheckInDate"
	},
	"intent":	{
		"name": "BookHotel",
		"slots":	{
			"CheckInDate":	{
			},
			"Location":	{
				"value":	{
```

```
                        "originalValue":        "san jose",
                        "interpretedValue":     "san jose",
                        "resolvedValues":       []
                    }
                },
                "Nights":       {
                },
                "RoomType":     {
                }
            },
            "state":        "InProgress",
            "confirmationState":    "None"
        },
        "sessionAttributes":    {
            "x-amz-lex:channels:platform":  "UniMRCP"
        },
        "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
    },
    "sessionId":    "4c16b190426443d7",
    "eventId":      "RESPONSE-9"
},
"textResponse": {
    "messages":    [{
            "content":      "What day do you want to check in?",
            "contentType":  "PlainText"
        }],
    "eventId":      "RESPONSE-10"
}
}
</instance>
<input mode="speech">san jose</input>
</interpretation>
</result>
```

### Interaction 3

Input: March 1st

C->S:

```
MRCP/2.0 314 RECOGNIZE 6
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
No-Input-Timeout: 30000
Speech-Complete-Timeout: 1500
```

```
Speech-Incomplete-Timeout: 15000
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 6 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 6 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 2864 RECOGNITION-COMPLETE 6 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 2677

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"march first",
"intentResult": {
     "inputMode":    "Speech",
     "interpretations":     [{
               "nluConfidence":      {
                    "score":      1
               },
               "intent":      {
                    "name": "BookHotel",
                    "slots":      {
                         "CheckInDate":  {
                              "value":      {
                                   "originalValue":      "march first",
                                   "interpretedValue":    "2021-03-01",
                                   "resolvedValues":     ["2021-03-01"]
```

```
                        }
                    },
                    "Location":    {
                        "value":        {
                            "originalValue":        "san jose",
                            "interpretedValue":     "san jose",
                            "resolvedValues":       []
                        }
                    },
                    "Nights":      {
                    },
                    "RoomType":    {
                    }
                },
                "state":        "InProgress",
                "confirmationState":    "None"
            }
        }, {
            "intent":       {
                "name": "FallbackIntent",
                "slots":        {
                }
            }
        }, {
            "nluConfidence":        {
                "score":        0.42
            },
            "intent":       {
                "name": "BookCar",
                "slots":        {
                    "CarType":      {
                    },
                    "DriverAge":    {
                    },
                    "PickUpCity":   {
                    },
                    "PickUpDate":   {
                        "value":        {
                            "originalValue":        "march",
                            "interpretedValue":     "2021-03-01",
                            "resolvedValues":       ["2021-03-01"]
                        }
                    },
                    "ReturnDate":   {
                    }
                }
            }
        }],
    "sessionState": {
        "dialogAction": {
            "type": "ElicitSlot",
            "slotToElicit": "Nights"
        },
        "intent":     {
            "name": "BookHotel",
            "slots":      {
                "CheckInDate":  {
                    "value":      {
                        "originalValue":      "march first",
                        "interpretedValue":    "2021-03-01",
                        "resolvedValues":      ["2021-03-01"]
                    }
                },
                "Location":    {
                    "value":      {
                        "originalValue":      "san jose",
                        "interpretedValue":    "san jose",
                        "resolvedValues":      []
                    }
                },
                "Nights":     {
                },
                "RoomType":    {
                }
            },
            "state":      "InProgress",
            "confirmationState":    "None"
        },
        "sessionAttributes":   {
            "x-amz-lex:channels:platform":  "UniMRCP"
        },
        "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
    },
    "sessionId":    "4c16b190426443d7",
    "eventId":     "RESPONSE-14"},
"textResponse": {
    "messages":    [{
            "content":     "How many nights will you be staying?",
            "contentType":  "PlainText"
        }],
    "eventId":     "RESPONSE-15"
}
}
</instance>
<input mode="speech">march first</input>
</interpretation>
</result>
```

### Interaction 4

Input: 2 [nights]

C->S:

```
MRCP/2.0 314 RECOGNIZE 8
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
Speech-Complete-Timeout: 1500
Speech-Incomplete-Timeout: 15000
No-Input-Timeout: 30000
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 83 8 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 8 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 3162 RECOGNITION-COMPLETE 8 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 2975

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"two",
"intentResult": {
        "inputMode":    "Speech",
        "interpretations":      [{
                "nluConfidence":        {
                    "score":        1
                },
                "intent":       {
                    "name": "BookHotel",
                    "slots":        {
                        "CheckInDate":  {
                            "value":        {
                                "originalValue":        "march first",
                                "interpretedValue":     "2021-03-01",
                                "resolvedValues":       ["2021-03-01"]
                            }
                        },
                        "Location":     {
                            "value":        {
                                "originalValue":        "san jose",
                                "interpretedValue":     "san jose",
                                "resolvedValues":       []
                            }
                        },
                        "Nights":       {
                            "value":        {
                                "originalValue":        "two",
                                "interpretedValue":     "2",
                                "resolvedValues":       ["2"]
                            }
                        },
                        "RoomType":     {
                        }
                    },
                    "state":        "InProgress",
                    "confirmationState":    "None"
                }
            }, {
                "intent":       {
                    "name": "FallbackIntent",
                    "slots":        {
                    }
                }
            }, {
                "nluConfidence":        {
                    "score":        0.32
                },
                "intent":       {
                    "name": "BookCar",
                    "slots":        {
                        "CarType":      {
                        },
                        "DriverAge":    {
                    },
                    "PickUpCity":   {
                    },
                    "PickUpDate":   {
                        "value":        {
                            "originalValue":        "two",
                            "interpretedValue":     "2021-03-02",
                            "resolvedValues":       ["2021-03-02"]
                        }
                    },
                    "ReturnDate":   {
                    }
                }
            }
        }],
    "sessionState": {
        "dialogAction": {
            "type": "ElicitSlot",
            "slotToElicit": "RoomType"
        },
        "intent":       {
            "name": "BookHotel",
            "slots":        {
                "CheckInDate":  {
                    "value":        {
                        "originalValue":        "march first",
                        "interpretedValue":     "2021-03-01",
                        "resolvedValues":       ["2021-03-01"]
                    }
                },
                "Location":     {
                    "value":        {
                        "originalValue":        "san jose",
                        "interpretedValue":     "san jose",
                        "resolvedValues":       []
                    }
                },
                "Nights":       {
                    "value":        {
                        "originalValue":        "two",
                        "interpretedValue":     "2",
                        "resolvedValues":       ["2"]
                    }
                },
                "RoomType":     {
                }
            },
            "state":        "InProgress",
            "confirmationState":    "None"
        },
        "sessionAttributes":    {
                "x-amz-lex:channels:platform":  "UniMRCP"
            },
            "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
        },
        "sessionId":    "4c16b190426443d7",
        "eventId":      "RESPONSE-17"
    },
    "textResponse": {
        "messages":    [{
                "content":      "What type of room would you like, queen, king or deluxe?",
                "contentType":  "PlainText"
            }],
        "eventId":      "RESPONSE-18"
    }
    }
    </instance>
    <input mode="speech">two</input>
    </interpretation>
    </result>
```

### Interaction 5

Input: Queen

C->S:

```
MRCP/2.0 315 RECOGNIZE 10
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
Speech-Incomplete-Timeout: 15000
No-Input-Timeout: 30000
Speech-Complete-Timeout: 1500
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 84 10 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 116 START-OF-INPUT 10 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 3321 RECOGNITION-COMPLETE 10 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 3133

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
"transcript":"queen",
"intentResult": {
        "inputMode":    "Speech",
        "interpretations":      [{
                        "nluConfidence":        {
                                "score":        1
                        },
                        "intent":       {
                                "name": "BookHotel",
                                "slots":        {
                                        "CheckInDate":  {
                                                "value":        {
                                                        "originalValue":        "march first",
                                                        "interpretedValue":     "2021-03-01",
                                                        "resolvedValues":       ["2021-03-01"]
                                                }
                                        },
                                        "Location":     {
                                                "value":        {
                                                        "originalValue":        "san jose",
                                                        "interpretedValue":     "san jose",
                                                        "resolvedValues":       []
                                                }
                                        },
                                        "Nights":       {
                                                "value":        {
                                                        "originalValue":        "two",
                                                        "interpretedValue":     "2",
                        "resolvedValues":      ["2"]
                    }
                },
                "RoomType":     {
                    "value":       {
                        "originalValue":       "queen",
                        "interpretedValue":    "queen",
                        "resolvedValues":      ["queen"]
                    }
                }
            },
            "state":       "InProgress",
            "confirmationState":    "None"
        }
    }, {
        "intent":      {
            "name": "FallbackIntent",
            "slots":       {
            }
        }
    }, {
        "nluConfidence":       {
            "score":       0.31
        },
        "intent":      {
            "name": "BookCar",
            "slots":       {
                "CarType":     {
                },
                "DriverAge":    {
                },
                "PickUpCity":   {
                },
                "PickUpDate":   {
                },
                "ReturnDate":   {
                }
            }
        }
    }],
"sessionState": {
    "dialogAction": {
        "type": "ConfirmIntent"
    },
    "intent":      {
        "name": "BookHotel",
        "slots":       {
            "CheckInDate":  {
                "value":       {
                    "originalValue":       "march first",
                    "interpretedValue":    "2021-03-01",
                        "resolvedValues":      ["2021-03-01"]
                    }
                },
                "Location":    {
                    "value":       {
                        "originalValue":       "san jose",
                        "interpretedValue":    "san jose",
                        "resolvedValues":      []
                    }
                },
                "Nights":      {
                    "value":       {
                        "originalValue":       "two",
                        "interpretedValue":    "2",
                        "resolvedValues":      ["2"]
                    }
                },
                "RoomType":    {
                    "value":       {
                        "originalValue":       "queen",
                        "interpretedValue":    "queen",
                        "resolvedValues":      ["queen"]
                    }
                }
            },
            "state":       "InProgress",
            "confirmationState":    "None"
        },
        "sessionAttributes":    {
            "x-amz-lex:channels:platform":  "UniMRCP"
        },
        "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
    },
    "sessionId":    "4c16b190426443d7",
    "eventId":      "RESPONSE-20" },
"textResponse": {
    "messages":    [{
            "content":      "Okay, I have you down for a 2 night stay in san jose starting
2021-03-01.  Shall I book the reservation?",
            "contentType":  "PlainText"
        }],
    "eventId":      "RESPONSE-21"
}
}
</instance>
<input mode="speech">queen</input>
</interpretation>
</result>
```

### Interaction 6

Input: Yes

C->S:

```
MRCP/2.0 315 RECOGNIZE 12
Channel-Identifier: 4c16b190426443d7@speechrecog
Content-Type: text/uri-list
Cancel-If-Queue: false
Start-Input-Timers: false
No-Input-Timeout: 30000
Speech-Complete-Timeout: 1500
Speech-Incomplete-Timeout: 15000
Content-Length: 46

builtin:speech/transcribe
builtin:dtmf/digits
```

S->C:

```
MRCP/2.0 84 12 200 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
```

S->C:

```
MRCP/2.0 116 START-OF-INPUT 12 IN-PROGRESS
Channel-Identifier: 4c16b190426443d7@speechrecog
Input-Type: speech
```

S->C:

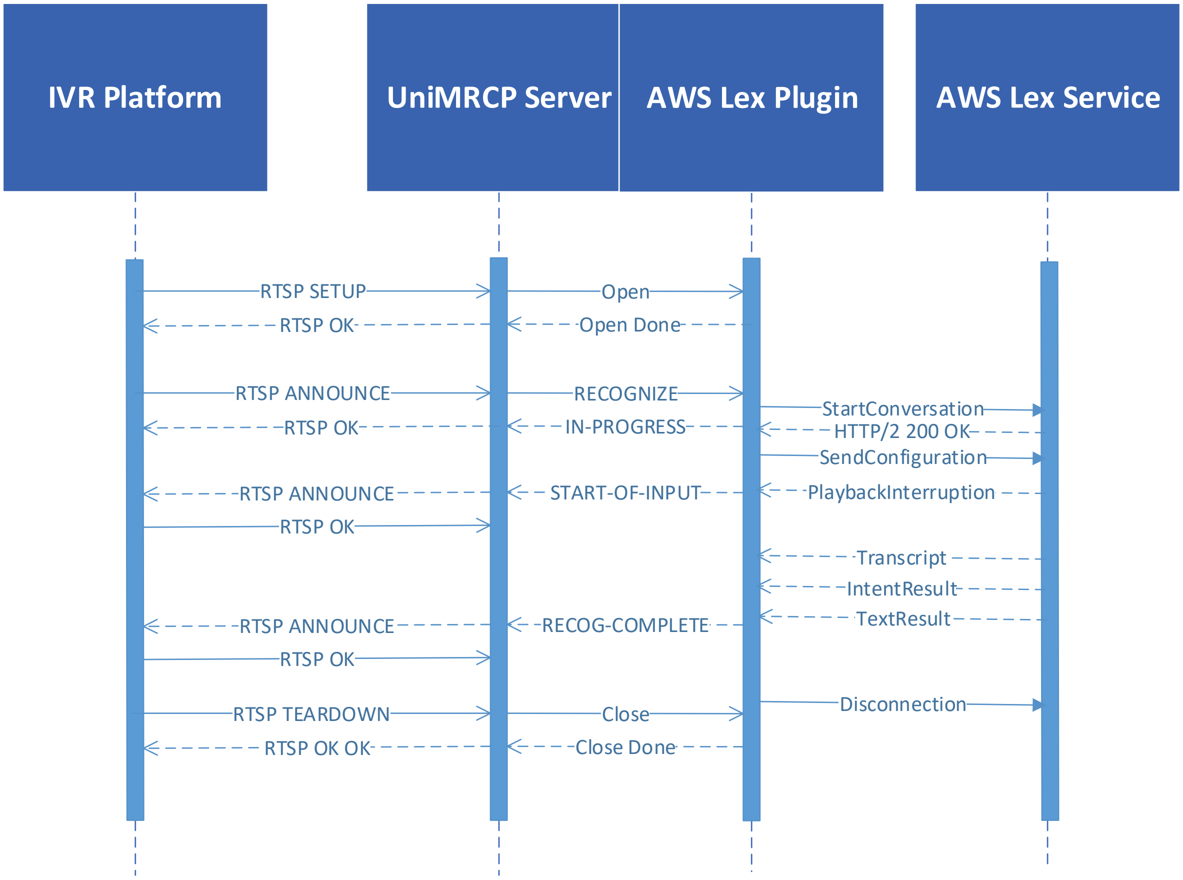
```
MRCP/2.0 3129 RECOGNITION-COMPLETE 12 COMPLETE
Channel-Identifier: 4c16b190426443d7@speechrecog
Completion-Cause: 000 success
Content-Type: application/x-nlsml
Content-Length: 2941

<?xml version="1.0"?>
<result>
<interpretation grammar="builtin:speech/transcribe" confidence="1">
<instance>
{
    "transcript":"yes",
    "intentResult": {
        "inputMode":    "Speech",
        "interpretations":      [{
                "nluConfidence":        {
                    "score":        1
                },
                "intent":       {
                    "name": "BookHotel",
                    "slots":        {
                        "CheckInDate":  {
                            "value":        {
                                "originalValue":        "march first",
                                "interpretedValue":     "2021-03-01",
                                "resolvedValues":       ["2021-03-01"]
                            }
                        },
                        "Location":     {
                            "value":        {
                                "originalValue":        "san jose",
                                "interpretedValue":     "san jose",
                                "resolvedValues":       []
                            }
                        },
                        "Nights":       {
                            "value":        {
                                "originalValue":        "two",
                                "interpretedValue":     "2",
                                "resolvedValues":       ["2"]
                            }
                        },
                        "RoomType":     {
                            "value":        {
                                "originalValue":        "queen",
                                "interpretedValue":     "queen",
                                "resolvedValues":       ["queen"]
                            }
                        }
                    },
                    "state":        "ReadyForFulfillment",
                    "confirmationState":    "Confirmed"
                }
            }, {
                "intent":       {
                    "name": "FallbackIntent",
                    "slots":        {
                    }
                }
            }, {
                "nluConfidence":        {
                    "score":        0.16
                },
                "intent":    {
                    "name": "BookCar",
                    "slots":    {
                        "CarType":    {
                        },
                        "DriverAge":   {
                        },
                        "PickUpCity":  {
                        },
                        "PickUpDate":  {
                        },
                        "ReturnDate":  {
                        }
                    }
                }
            }],
        "sessionState": {
            "dialogAction": {
                "type": "Close"
            },
            "intent":    {
                "name": "BookHotel",
                "slots":    {
                    "CheckInDate":  {
                        "value":    {
                            "originalValue":     "march first",
                            "interpretedValue":   "2021-03-01",
                            "resolvedValues":     ["2021-03-01"]
                        }
                    },
                    "Location":   {
                        "value":    {
                            "originalValue":     "san jose",
                            "interpretedValue":   "san jose",
                            "resolvedValues":     []
                        }
                    },
                    "Nights":    {
                        "value":    {
                            "originalValue":     "two",
                            "interpretedValue":   "2",
                            "resolvedValues":     ["2"]
                        }
                    },
                    "RoomType":   {
                        "value":    {
                            "originalValue":     "queen",
                            "interpretedValue":   "queen",
                            "resolvedValues":     ["queen"]
                        }
```

```
                                }
                        },
                        "state":        "ReadyForFulfillment",
                        "confirmationState":    "Confirmed"
                },
                "sessionAttributes":    {
                        "x-amz-lex:channels:platform":  "UniMRCP"
                },
                "originatingRequestId": "623d5403-f172-43f7-94c0-9f7d504850b9"
        },
        "sessionId":    "4c16b190426443d7",
        "eventId":      "RESPONSE-25"
},
"textResponse": {
        "eventId":      "RESPONSE-26"
}
}
</instance>
<input mode="speech">yes</input>
</interpretation></result>
```

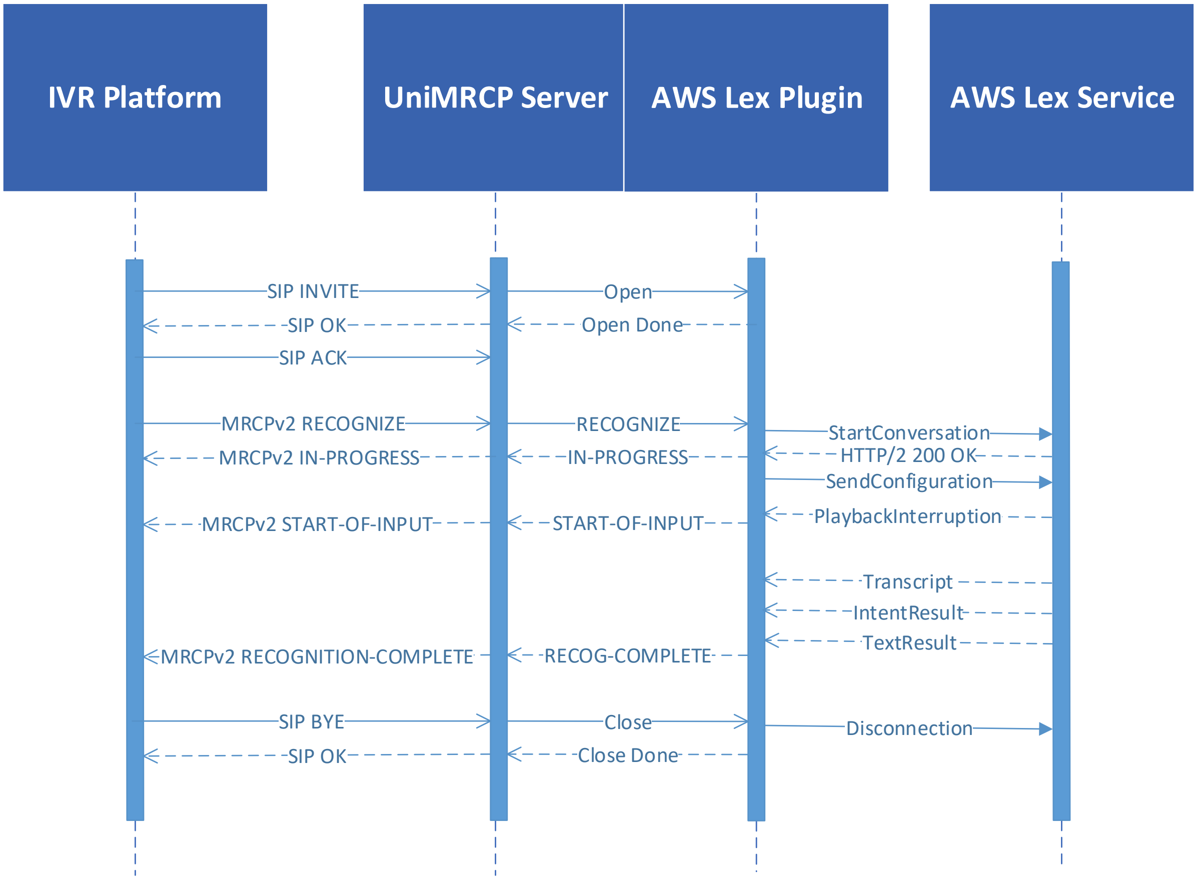
# 8 Sequence Diagrams

The following sequence diagrams outline common interactions between all the main components involved in a typical recognition session performed over MRCPv1 and MRCPv2 respectively.

## 8.1 MRCPv1



## 8.2 MRCPv2


IVR Platform
UniMRCP Server
AWS Lex Plugin
AWS Lex Service
SIP INVITE
Open
SIP OK
Open Done
SIP ACK
MRCPv2 RECOGNIZE
RECOGNIZE
StartConversation
MRCPv2 IN-PROGRESS
IN-PROGRESS
HTTP/2 200 OK
SendConfiguration
MRCPv2 START-OF-INPUT
START-OF-INPUT
PlaybackInterruption
Transcript
IntentResult
TextResult
MRCPv2 RECOGNITION-COMPLETE
RECOG-COMPLETE
SIP BYE
Close
Disconnection
SIP OK
Close Done

# 9 References

## 9.1 AWS Lex

- What is Amazon Lex
- How It Works
- API Reference

## 9.2 Specifications

- Speech Recognizer Resource
- NLSML Results