

*Powered by Universal Speech Solutions LLC*



# AWS Lex Plugin

## Usage Guide

---

Revision: 7

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 .....	15
3.9 License Server.....	15
4 Configuration Steps .....	18
4.1 Using Default Configuration.....	18
4.2 Using with Polly .....	18
4.3 Specifying AWS Credentials .....	18
4.4 Specifying Lex Bot .....	19
4.5 Specifying Recognition Language.....	20
4.6 Specifying Sampling Rate.....	20
4.7 Specifying Speech Input Parameters .....	20
4.8 Specifying DTMF Input Parameters.....	21
4.9 Specifying No-Input and Recognition Timeouts .....	21
4.10 Maintaining Utterances.....	21
4.11 Maintaining Recognition Details Records .....	22
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 Booking.....	28
8	Sequence Diagrams.....	37
8.1	MRCPv1 .....	37
8.2	MRCPv2 .....	38
9	References.....	39
9.1	AWS Lex .....	39
9.2	Specifications.....	39

# 1 Overview

This guide describes how to configure and use the Amazon Web Services (AWS) Lex V1 plugin to the UniMRCP server. 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 1.5.0 and above

UniMRCP Lex Plugin > 1.0.0 and < 2.0.0

## 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
<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.
<b>credentials-file</b>	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.
<b>credentials-provider</b>	String	Specifies a credentials provider. If not initialized or set to <i>custom</i> , the custom credentials provider is used to read credentials from <i>credentials-file</i> . Otherwise, if set to <i>default</i> , the AWS default credentials provider chain is used, and <i>credentials-file</i> is not observer. Available since 1.3.0.  Otherwise, if set to <i>sts</i> , the AWS STS profile credentials provider is used, and <i>credentials-file</i> is not observer. Available since 1.4.0.
<b>init-sdk</b>	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.
<b>shutdown-sdk</b>	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.
<b>sdk-log-level</b>	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). Available since 1.4.0.
--	--	---

## Parent

None.

## Children

Name	Unit	Description
<code>&lt;streaming-recognition&gt;</code>	String	Specifies recognition parameters of streaming recognition.
<code>&lt;results&gt;</code>	String	Specifies parameters of recognition results set in RECOGNITION-COMPLETE events.
<code>&lt;speech-dtmf-input-detector&gt;</code>	String	Specifies parameters of the speech and DTMF input detector.
<code>&lt;utterance-manager&gt;</code>	String	Specifies parameters of the utterance manager.
<code>&lt;rdr-manager&gt;</code>	String	Specifies parameters of the Recognition Details Record (RDR) manager.
<code>&lt;monitoring-agent&gt;</code>	String	Specifies parameters of the monitoring manager.
<code>&lt;license-server&gt;</code>	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.

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

## 3.2 Streaming Recognition

This element specifies parameters of streaming recognition.

### Attributes



Name	Unit	Description
<b>language</b>	String	Specifies the default language to use, if not set by the client.
<b>skip-unsupported-grammars</b>	Boolean	Specifies whether to skip or raise an error while referencing a malformed or not supported grammar.
<b>transcription-grammar</b>	String	Specifies the name of the built-in speech transcription grammar. The grammar can be referenced as <i>builtin:speech/transcribe</i> or <i>builtin:grammar/transcribe</i> , where <i>transcribe</i> is the default value of this parameter.
<b>thread-pool-size</b>	Integer	Specifies the size of the executor thread pool provided by AWS SDK.
<b>region</b>	String	Specifies the AWS region of the Lex Bot. <a href="https://docs.aws.amazon.com/general/latest/gr/rande.html#lex_region">https://docs.aws.amazon.com/general/latest/gr/rande.html#lex_region</a>
<b>bot-name</b>	String	Specifies the name of the Lex bot. Can be overridden by client via Vendor-Specific-Parameters.
<b>alias</b>	String	Specifies the alias of the Lex bot. Can be overridden by client via Vendor-Specific-Parameters.
<b>output-format</b>	String	Specifies the format of messages sent in a response from the Lex bot. If not specified, the format is not restricted. Can be overridden by client via Vendor-Specific-Parameters. Available since 1.1.0.
<b>timezone</b>	String	Specifies the timezone used to resolve dates. If not specified, the timezone associated to the region of the Lex bot is used. Can be overridden by client via Vendor-Specific-Parameters. Available since 1.1.0.
<b>proxy-host</b>	String	Specifies the host name of HTTP proxy, if used. Available since 1.4.0.
<b>proxy-port</b>	Integer	Specifies the port number of HTTP proxy, if used. Available since 1.4.0.
<b>proxy-username</b>	String	Specifies the username employed for HTTP proxy authentication, if used. Available since 1.4.0.
<b>proxy-password</b>	String	Specifies the password employed for HTTP proxy authentication, if used. Available since 1.4.0.

## 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
<b>format</b>	String	Specifies the format of results to be returned to the client (use "standard" for NLSML and "json" for JSON).
<b>indent</b>	Integer	Specifies the indent to use while composing the results.
<b>confidence-format</b>	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)
<b>tag-format</b>	String	Specifies the format of the instance element to be returned. The parameter is observed only if the format is set to <i>standard</i> . Use one of: <ul style="list-style-type: none"><li>• <i>semantics/xml</i> for query result represented in XML [default]</li><li>• <i>semantics/json</i> for query result represented in JSON</li><li>• <i>swi-semantics/xml</i> for query result set in an inner</li></ul>

		<p>&lt;SWI_meaning&gt; element represented in XML</p> <ul style="list-style-type: none"> <li>• <i>swi-semantic/json</i> for query result set in an inner &lt;SWI_meaning&gt; element represented in JSON</li> </ul> <p>Available since 1.3.0.</p>
--	--	---

### 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
<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 'umslex-', 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 1.5.0.

### 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
<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 'umslex-', 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 1.5.0.

### 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
<b>refresh-period</b>	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
<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 1.4.0.
<b>http-proxy-port</b>	Integer	Specifies the port number of the HTTP proxy server, if used. Available since 1.4.0.

### 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_*.cert"
  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 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.

```
<umspolly license-file="umspolly_*.lic" credentials-file="aws.credentials" init-sdk="true"
shutdown-sdk="false">
<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
<b>aws-credentials-file</b>	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. Available since 1.6.0.
<b>aws-credentials-provider</b>	String	Specifies a credentials provider. Use one of: <ul style="list-style-type: none"> <li><i>custom</i> for credentials read from the specified file</li> <li><i>default</i> for the AWS default credentials provider chain</li> <li><i>sts</i> for the AWS STS profile credentials provider</li> </ul> Available since 1.6.0.
<b>aws-credentials-profile</b>	String	Specifies a credentials profile to reference and/or create. Available since 1.6.0.
<b>aws-credentials-duration</b>	Integer	Specifies a lifetime of the credentials profile to create. Available since 1.6.0.
<b>aws-arn-role</b>	String	Specifies an ARN role. Available since 1.6.0.
<b>aws-region</b>	String	Specifies an AWS region. Available since 1.6.0.

## 4.4 Specifying Lex Bot

The parameters of the Lex bot *region*, *bot-name* and *alias* must be specified in the configuration file

*umslex.xml*. For example:

```
<streaming-recognition
  language="en-US"
  region="us-west-2"
  bot-name="BookTrip"
  alias="Dev"
/>
```

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

```
Vendor-Specific-Parameters: bot-name=BookTrip; alias=Dev
```

## 4.5 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*.

## 4.6 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.

The native sampling rate with the linear16 audio encoding is used to post audio data to the Lex service.

## 4.7 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

## 4.8 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.

## 4.9 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.

## 4.10 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*.

## 4.11 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 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/$id
```

As a result, captured DTMFs will be posted to the Lex service 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 NLSML result composed as follows.

#### `<input>`

The `<input>` element in an NLSML result is set to the input transcript available in the structure `PostContentResult` received in a response to the `PostContent` request.

#### `<instance>`

By default, the `<instance>` element in an NLSML result is set to an XML representation of the structure `PostContentResult` received in a response to the `PostContent` request. Since 1.3.0, this behavior can be adjusted via the *tag-format* attribute, which accepts the following values.

- `semantics/xml`

The default setting. The `PostContentResult` structure is represented in XML.



- semantics/json

The PostContentResult is represented in JSON.

- swi-semantics/xml

The PostContentResult structure is set in an inner `<SWI_meaning>` element being represented in XML.

- swi-semantics/json

The PostContentResult 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 the PostContentResult structure received in a response to the PostContent request.

## 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 *\${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. Available since 1.2.0.

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 *umslex-channels.status*, located by default in the directory *\${UniMRCPIInstallDir}/var/status*.

# 7 Usage Examples

## 7.1 Hotel Booking

This example demonstrates an MRCP message exchange based on a conversation with the sample BookTrip Lex bot.

### Interaction 1

Input: Book a hotel

C->S:

```
MRCP/2.0 361 RECOGNIZE 1
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 50000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Sensitivity-Level: 0.5
Save-Waveform: true
Content-Length: 25

builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 1 200 IN-PROGRESS
Channel-Identifier: 2a203b9adebe40e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 1 IN-PROGRESS
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Input-Type: speech
```

S->C:

MRCP/2.0 823 RECOGNITION-COMPLETE 1 COMPLETE  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Completion-Cause: 000 success  
Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-1.wav>;size=21440;duration=1340  
Content-Type: application/x-nlsml  
Content-Length: 538

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
    <instance>
      <intent>BookHotel</intent>
      <slots>
        <CheckInDate></CheckInDate>
        <Location></Location>
        <Nights></Nights>
        <RoomType></RoomType>
      </slots>
      <message>What city will you be staying in?</message>
      <dialogstate>ElicitSlot</dialogstate>
      <slottoelicit>Location</slottoelicit>
    </instance>
    <input mode="speech">book a hotel</input>
  </interpretation>
</result>
```

## Interaction 2

Input: Sunnyvale

C->S:

MRCP/2.0 361 RECOGNIZE 2  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Content-Id: request1@form-level  
Content-Type: text/uri-list  
Cancel-If-Queue: false  
No-Input-Timeout: 50000  
Recognition-Timeout: 10000  
Start-Input-Timers: true  
Confidence-Threshold: 0.87  
Sensitivity-Level: 0.5  
Save-Waveform: true  
Content-Length: 25

builtin:speech/transcribe

S->C:

MRCP/2.0 83 2 200 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog

S->C:

MRCP/2.0 115 START-OF-INPUT 2 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Input-Type: speech

S->C:

MRCP/2.0 832 RECOGNITION-COMPLETE 2 COMPLETE  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Completion-Cause: 000 success  
Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-2.wav>;size=18400;duration=1150  
Content-Type: application/x-nlsml  
Content-Length: 547

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
    <instance>
      <intent>BookHotel</intent>
      <slots>
        <CheckInDate></CheckInDate>
        <Location>sunnyvale</Location>
        <Nights></Nights>
        <RoomType></RoomType>
      </slots>
      <message>What day do you want to check in?</message>
      <dialogstate>ElicitSlot</dialogstate>
      <slottoelicit>CheckInDate</slottoelicit>
    </instance>
    <input mode="speech">sunnyvale</input>
  </interpretation>
</result>
```

### Interaction 3

Input: October 1<sup>st</sup>

C->S:

```
MRCP/2.0 361 RECOGNIZE 3
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 50000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Sensitivity-Level: 0.5
Save-Waveform: true
Content-Length: 25

builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 3 200 IN-PROGRESS
Channel-Identifier: 2a203b9adebe40e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 3 IN-PROGRESS
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Input-Type: speech
```

S->C:

```
MRCP/2.0 844 RECOGNITION-COMPLETE 3 COMPLETE
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Completion-Cause: 000 success
Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-
3.wav>;size=23520;duration=1470
Content-Type: application/x-nlsml
Content-Length: 559

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

```
<interpretation grammar="builtin:speech/transcribe" confidence="1">
  <instance>
    <intent>BookHotel</intent>
    <slots>
      <CheckInDate>2018-10-01</CheckInDate>
      <Location>sunnyvale</Location>
      <Nights></Nights>
      <RoomType></RoomType>
    </slots>
    <message>How many nights will you be staying?</message>
    <dialogstate>ElicitSlot</dialogstate>
    <slottoelicit>Nights</slottoelicit>
  </instance>
  <input mode="speech">october first</input>
</interpretation>
</result>
```

## Interaction 4

Input: 2 [nights]

C->S:

```
MRCP/2.0 361 RECOGNIZE 4
Channel-Identifier: 2a203b9adebe40e7@speechrecog
Content-Id: request1@form-level
Content-Type: text/uri-list
Cancel-If-Queue: false
No-Input-Timeout: 50000
Recognition-Timeout: 10000
Start-Input-Timers: true
Confidence-Threshold: 0.87
Sensitivity-Level: 0.5
Save-Waveform: true
Content-Length: 25

builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 4 200 IN-PROGRESS
Channel-Identifier: 2a203b9adebe40e7@speechrecog
```

S->C:



MRCP/2.0 115 START-OF-INPUT 4 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Input-Type: speech

S->C:

MRCP/2.0 856 RECOGNITION-COMPLETE 4 COMPLETE  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Completion-Cause: 000 success  
Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-4.wav>;size=13760;duration=860  
Content-Type: application/x-nlsml  
Content-Length: 572

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
    <instance>
      <intent>BookHotel</intent>
      <slots>
        <CheckInDate>2018-10-01</CheckInDate>
        <Location>sunnyvale</Location>
        <Nights>2</Nights>
        <RoomType></RoomType>
      </slots>
      <message>What type of room would you like, queen, king or deluxe?</message>
      <dialogstate>ElicitSlot</dialogstate>
      <slottoelicit>RoomType</slottoelicit>
    </instance>
    <input mode="speech">two</input>
  </interpretation>
</result>
```

## Interaction 5

Input: King

C->S:

MRCP/2.0 361 RECOGNIZE 5  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Content-Id: request1@form-level  
Content-Type: text/uri-list

Cancel-If-Queue: false  
No-Input-Timeout: 50000  
Recognition-Timeout: 10000  
Start-Input-Timers: true  
Confidence-Threshold: 0.87  
Sensitivity-Level: 0.5  
Save-Waveform: true  
Content-Length: 25

builtin:speech/transcribe

S->C:

MRCP/2.0 83 5 200 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog

S->C:

MRCP/2.0 115 START-OF-INPUT 5 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Input-Type: speech

S->C:

MRCP/2.0 869 RECOGNITION-COMPLETE 5 COMPLETE  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Completion-Cause: 000 success  
Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-5.wav>;size=12960;duration=810  
Content-Type: application/x-nlsml  
Content-Length: 585

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
    <instance>
      <intent>BookHotel</intent>
      <slots>
        <CheckInDate>2018-10-01</CheckInDate>
        <Location>sunnyvale</Location>
        <Nights>2</Nights>
        <RoomType>king</RoomType>
      </slots>
    </instance>
  </interpretation>
</result>
```

```
<message>Okay, I have you down for a 2 night stay in sunnyvale starting 2018-10-01.  
Shall I book the reservation?</message>  
<dialogstate>ConfirmIntent</dialogstate>  
</instance>  
<input mode="speech">king</input>  
</interpretation>  
</result>
```

## Interaction 6

Input: Yes

C->S:

```
MRCP/2.0 361 RECOGNIZE 6  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Content-Id: request1@form-level  
Content-Type: text/uri-list  
Cancel-If-Queue: false  
No-Input-Timeout: 50000  
Recognition-Timeout: 10000  
Start-Input-Timers: true  
Confidence-Threshold: 0.87  
Sensitivity-Level: 0.5  
Save-Waveform: true  
Content-Length: 25  
  
builtin:speech/transcribe
```

S->C:

```
MRCP/2.0 83 6 200 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog
```

S->C:

```
MRCP/2.0 115 START-OF-INPUT 6 IN-PROGRESS  
Channel-Identifier: 2a203b9adebe40e7@speechrecog  
Input-Type: speech
```

S->C:

MRCP/2.0 743 RECOGNITION-COMPLETE 6 COMPLETE

Channel-Identifier: 2a203b9adebe40e7@speechrecog

Completion-Cause: 000 success

Waveform-Uri: <http://localhost/utterances/umslex-2a203b9adebe40e7-6.wav>;size=14880;duration=930

Content-Type: application/x-nlsml

Content-Length: 459

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

```
<result>
```

```
  <interpretation grammar="builtin:speech/transcribe" confidence="1">
```

```
    <instance>
```

```
      <intent>BookHotel</intent>
```

```
      <slots>
```

```
        <CheckInDate>2018-10-01</CheckInDate>
```

```
        <Location>sunnyvale</Location>
```

```
        <Nights>2</Nights>
```

```
        <RoomType>king</RoomType>
```

```
      </slots>
```

```
      <dialogstate>ReadyForFulfillment</dialogstate>
```

```
    </instance>
```

```
    <input mode="speech">yes</input>
```

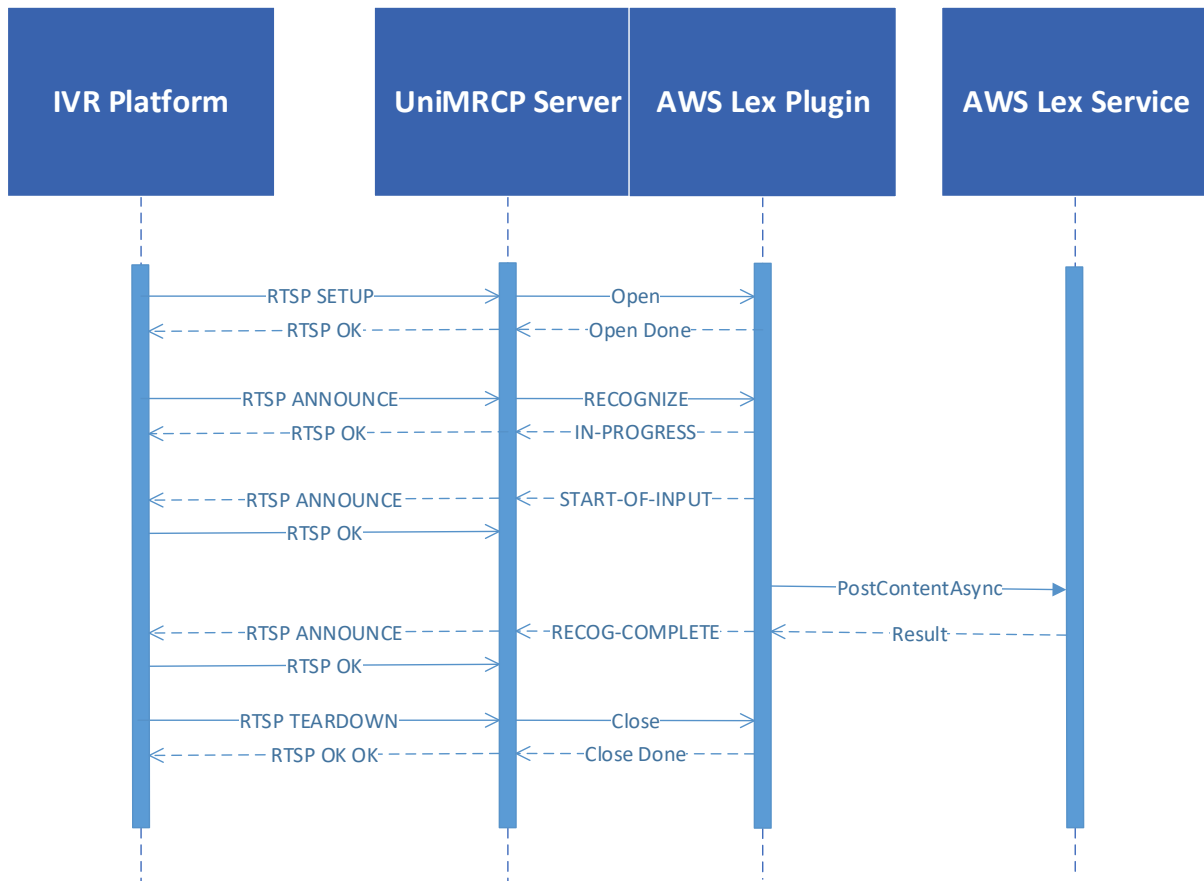
```
  </interpretation>
```

```
</result>
```

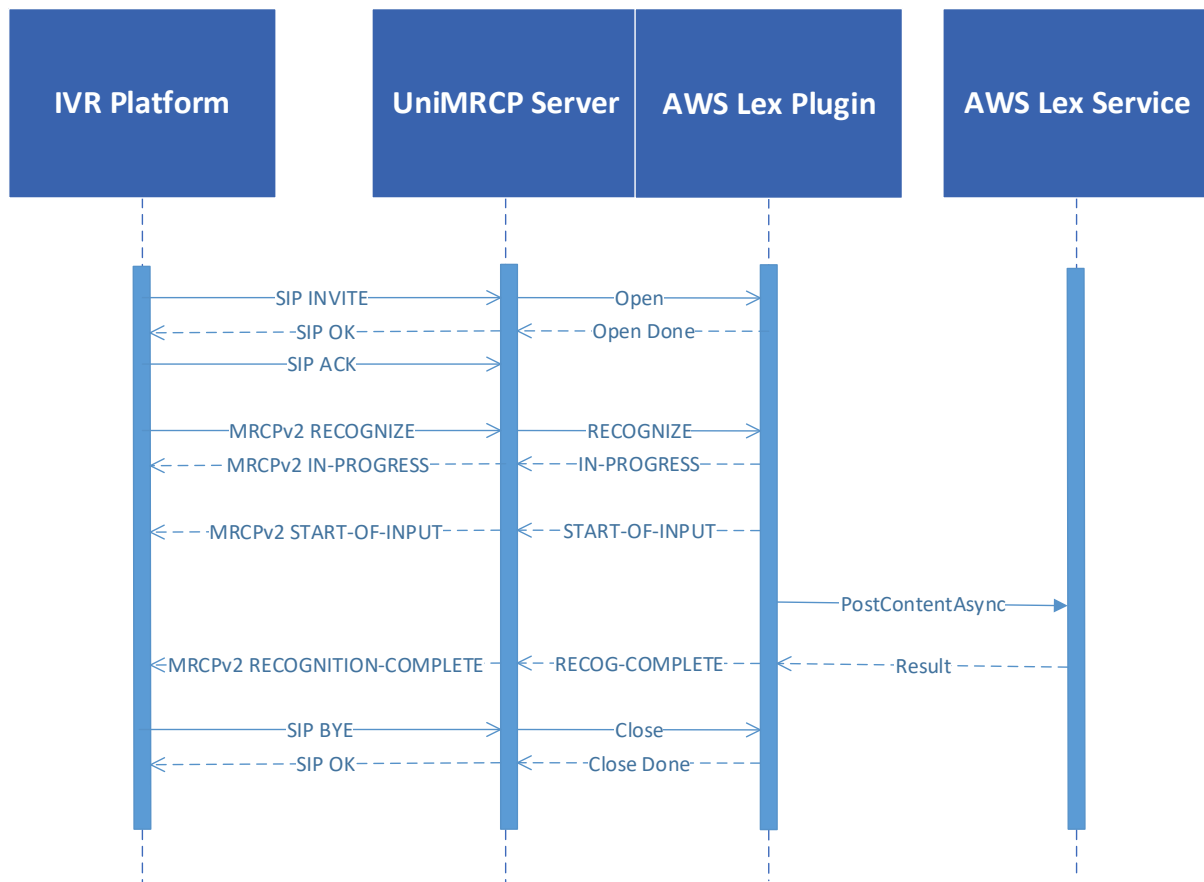
# 8 Sequence Diagrams

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



# 9 References

## 9.1 AWS Lex

- [What is Amazon Lex](#)
- [How It Works](#)
- [API Reference](#)

## 9.2 Specifications

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