Powered by Universal Speech Solutions LLC

UniMRCP

AWS Lex Plugin

Administrator Guide

Revision: 4

Distribution: Red Hat / Cent OS

Created: September 15, 2018

Last updated: March 16, 2021

Author: Arsen Chaloyan

Table of Contents

1 Overview

This guide describes how to obtain and install binary packages for the Amazon Web Services (AWS) Lex plugin to the UniMRCP server on Red Hat-based Linux distributions. The document is intended for system administrators and developers.

1.1 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 above

1.2 Supported Distributions

UniMRCP RPMs are currently available for x86_64 (64-bit) architecture only.

Operating System	Released	End of Support
Red Hat / Cent OS 7	September 2018	TBA
Red Hat / Cent OS 8	January 2021	TBA

Note: packages for other distributions can be made available upon request. For more information, contact services@unimrcp.org.

1.3 Authentication

UniMRCP binary packages are available to authenticated users only. In order to register a free account with UniMRCP, please visit the following page.

https://www.unimrcp.org/profile-registration

Note: a new account needs to be verified and activated prior further proceeding.

2 Installing RPMs Using YUM

Using the Yellowdog Updater, Modifier (yum), a command-line package management utility for Red Hat-based distributions, is recommended for installation of UniMRCP binary packages.

2.1 Repository Configuration

The content of a typical yum configuration file, to be placed in /etc/yum.repos.d/unimrcp.repo, is provided below.

```
[unimrcp]
name=UniMRCP Packages for Red Hat / Cent OS-$releasever $basearch
baseurl=https://username:password@unimrcp.org/repo/yum/main/rhel$releasever/$basearch/
enabled=1
sslverify=1
gpgcheck=1
gpgkey=https://unimrcp.org/keys/unimrcp-gpg-key.public

[unimrcp-noarch]
name=UniMRCP Packages for Red Hat / Cent OS-$releasever noarch
baseurl=https://username:password@unimrcp.org/repo/yum/main/rhel$releasever/noarch/
enabled=1
sslverify=1
gpgcheck=1
gpgkey=https://unimrcp.org/keys/unimrcp-gpg-key.public
```

The username and password fields included in the HTTPS URI must be replaced with the corresponding account credentials.

2.2 Repository Verification

In order to verify that yum can properly connect and access the UniMRCP repository, the following command can be used.

```
yum repolist unimrcp
yum repolist unimrcp-noarch
```

where *unimrcp* and *unimrcp-noarch* are names of the sections set in the yum configuration file above.

In order to retrieve a list of packages the UniMRCP repository provides, the following command can be used.

```
yum --disablerepo="*" --enablerepo="unimrcp" list available
yum --disablerepo="*" --enablerepo="unimrcp-noarch" list available
```

2.3 Lex Plugin Installation

Lex V2

In order to install the plugin for the Lex V2 API, including all the dependencies, use the following command.

```
yum install unimrcp-lex
```

Lex V1

In order to install the plugin for the Lex V1 API, including all the dependencies, use the following command.

```
yum install unimrcp-lexv1
```

Note: either the plugin for Lex V2 or V1 shall be installed.

In order to install the additional data files for the sample client application *umc*, the following command can be used.

```
yum install umc-addons
```

Note: this package is optional and provides additional data which can be used for validation of basic setup.

3 Installing RPMs Manually

UniMRCP RPM packages can be installed manually using the *rpm* utility. Note, however, that the system administrator should take care of package dependencies and install all the packages in appropriate order.

The RPM packages have the following naming convention:

```
$packagename-$universion-$packageversion.el$rhelversion.$arch.rpm
```

where

- *packagename* is the name of a package
- *universion* is the UniMRCP version
- *packageversion* is the RPM release version
- *rhelversion* is the Red Hat version
- *arch* is the architecture (x86_64, i686, …)

3.1 Package List

The following is a list of UniMRCP RPM packages required for the installation of the Lex plugin.

Package Name	Description
unimrcp-lex	AWS Lex plugin to the server supporting Lex V2 API
unimrcp-lexv1	AWS Lex plugin to the server supporting Lex V1 API
uniawssdk	UniMRCP edition of the AWS SDK CPP library.
uniawssdk-deps	UniMRCP edition of the dependencies of AWS SDK CPP library.
umc-addons	Sample en-US data files used with umc. [Optional]
unilicnodegen	Node information retrieval tool, required for license deployment.
unimrcp-server	Shared library and application of the server.
unimrcp-client	Shared libraries and sample applications of the client. [Optional]
unimrcp-demo-plugins	Set of demo plugins to the server. [Optional]

unimrcp-common	Data common for the client and the server.
uniapr	UniMRCP edition of the Apache Portable Runtime (APR) library.
uniapr-util	UniMRCP edition of the Apache Portable Runtime Utility (APR-Util) library.
unisofia-sip	UniMRCP edition of the Sofia SIP library.

3.2 Package Installation Order

Note that all the RPM packages provided by UniMRCP are signed by a GNU Privacy Guard (GPG) key. Before starting the installation, you may need to import the public key in order to allow the *rpm* utility to verify the packages.

```
rpm --import https://unimrcp.org/keys/unimrcp-gpg-key.public
```

Packages for the APR, APR-Util and Sofia-SIP libraries must be installed first.

```
rpm -ivh uniapr-$aprversion-$packageversion.el$rhelversion.$arch.rpm
rpm -ivh uniapr-util-$apuversion-$packageversion.el$rhelversion.$arch.rpm
rpm -ivh unisofia-sip-$sofiaversion-$packageversion.el$rhelversion.$arch.rpm
```

Then, a package containing common data for the client and the server, and a package for the server should follow.

```
rpm -ivh unimrcp-common-$uniersion-$packageversion.el$rhelversion.$arch.rpm
rpm -ivh unimrcp-server-$uniersion-$packageversion.el$rhelversion.$arch.rpm
```

Next, a package containing the utility tool *unilicnodegen*, required for license deployment.

```
rpm -ivh unilicnodegen-$toolversion-$packageversion.el$rhelversion.$arch.rpm
```

Next, package containing the AWS SDK library and the dependencies.

```
rpm -ivh uniawssdk-$awssdk-depsversion-$packageversion.el$rhelversion.$arch.rpm
rpm -ivh uniawssdk-$awssdkversion-$packageversion.el$rhelversion.$arch.rpm
```

Finally, a package containing the Lex plugin should follow.

```
rpm -ivh unimrcp-lex-$universion-$packageversion.el$rhelversion.noarch.rpm
```

4 Obtaining License

The Lex plugin to the UniMRCP server is a commercial product, which requires a license file to be installed.

4.1 License Type

The following license types are available:

- Trial
- Production
- Test and Development

4.2 Node Information

The license files are bound to a node the product is installed on. In order to obtain a license, the corresponding node information needs to be retrieved and submitted for generation of a license file.

Use the installed tool *unilicnodegen* to retrieve the node information.

```
/opt/unimrcp/bin/unilicnodegen
```

As a result, a text file *uninode.info* will be saved in the current directory. Submit the file *uninode.info* for license generation to services@unimrcp.org by mentioning the product name in the subject.

4.3 License Installation

The license file needs to be placed into the directory */opt/unimrcp/data*.

```
cp umslex_*.lic /opt/unimrcp/data
```

5 Obtaining Service Credentials

In order to utilize the AWS Lex API, corresponding service credentials need to be retrieved from the AWS console and further installed to the UniMRCP server.

5.1 Create IAM User

Sign up for an AWS account and create an IAM user.

```
https://docs.aws.amazon.com/lex/latest/dg/gs-account.html
```

5.2 Installation of Credentials

Create a text file *aws.credentials* in the directory */opt/unimrcp/data*.

```
nano /opt/unimrcp/data/aws.credentials
```

Place your AWS IAM user credentials in the following format.

```
{
  "aws_access_key_id": "••••••••••••",
  "aws_secret_access_key": "••••••••••••••••••••••••••••••••••••"
}
```

6 Configuring Server and Plugin

6.1 Plugin Factory Configuration

In order to load the Lex plugin into the UniMRCP server, open the file *unimrcpserver.xml*, located in the directory */opt/unimrcp/conf*, and add the following entry under the XML element <*plugin-factory*>. Disable other speech recognition plugins, if available. The remaining demo plugins might also be disabled, if not installed.

```
<!-- Factory of plugins (MRCP engines) -->
<plugin-factory>
   <engine id="Demo-Synth-1" name="demosynth" enable="true"/>
   <engine id="Demo-Recog-1" name="demorecog" enable="false"/>
   <engine id="Demo-Verifier-1" name="demoverifier" enable="true"/>
   <engine id="Recorder-1" name="mrcprecorder" enable="true"/>
   <engine id="Lex-1" name="umslex" enable="true"/>
</plugin-factory>
```

6.2 Logger Configuration

In order to enable log output from the plugin and set filtering rules, open the configuration file *logger.xml*, located in the directory */opt/unimrcp/conf*, and add the following entry under the element <*sources*>.

```
<source name="LEX-PLUGIN" priority="INFO" masking="NONE"/>
```

6.3 Lex Plugin Configuration

The configuration file of the plugin is located in */opt/unimrcp/conf/umslex.xml*. Default settings should be sufficient for general use.

Refer to the *Usage Guide* for more information.

7 Validating Setup

Validate your setup by using the sample UniMRCP client and server applications on the same host. The default configuration and data files should be sufficient for a basic test.

7.1 Setting up Sample Lex Bot

Follow the instructions to create a sample BookTrip Lex bot.

In order to identify the created Lex bot, the corresponding parameters must be specified in the configuration file of the plugin, located in */opt/unimrcp/conf/umslex.xml*.

Lex V2

```
<streaming-recognition
  language="en-US"
  region="us-west-2"
  bot-name="Your-Bot-Name-ID"
  alias="Your-Bot-Alias-ID"
/>
```

Lex V1

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

7.2 Launching Server

Launch the UniMRCP server application.

```
cd /opt/unimrcp/bin
./unimrcpserver
```

In the server log output, check whether the plugin is normally loaded.

```
[INFO]   Load Plugin [Lex-1] [/opt/unimrcp/plugin/umslex.so]
```

Next, check for the license information.

```
[NOTICE] UniMRCP Lex License

-product name:    umslex
-product version: 1.0.0
-license owner:  -
-license type:    trial
-issue date:      2018-09-15
-exp date:        2018-10-15
-channel count:   2
-feature set:     0
```

Next, check that the service account credentials are normally populated.

```
[NOTICE] Read AWS Credentials /opt/unimrcp/data/aws.credentials
```

7.3 Launching Client

Note: the optional package *umc-addons* must be installed for this test to work.

Launch the sample UniMRCP client application *umc*.

```
cd /opt/unimrcp/bin
./umc
```

Run a typical speech recognition scenario by issuing the command *run lex1* from the console of the *umc* client application.

```
run lex1
```

This command sends a RECOGNIZE request to the server and then starts streaming a sample audio input file *bookroom.pcm* to recognize.

Check for the NLSML results to be returned as expected. Below is a sample result returned by Lex V1.

```
<?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 room</input>
  </interpretation>
</result>
```

Visually inspect the log output for any possible warnings or errors.

Note that utterances are stored in the *var* directory, if the corresponding parameter is enabled in the configuration file *umslex.xml* and/or requested by the client.