

Powered by Universal Speech Solutions LLC



Azure Bot Service Plugin

Administrator Guide

Revision: 2

Distribution: Debian / Ubuntu

Created: May 1, 2020

Last updated: January 4, 2021

Author: Arsen Chaloyan

Table of Contents


1 Overview.....	3
1.1 Applicable Versions.....	3
1.2 Supported APIs.....	3
1.3 Supported Distributions.....	3
1.4 Authentication.....	3
2 Installing Deb Packages Using Apt-Get.....	5
2.1 Repository Configuration.....	5
2.2 GnuPG Key.....	5
2.3 Repository Update.....	5
2.4 Azure Bot Plugin Installation.....	5
3 Installing Deb Packages Manually.....	7
3.1 Package List.....	7
3.2 Package Installation Order.....	8
4 Obtaining License.....	9
4.1 License Type.....	9
4.2 Node Information.....	9
4.3 License Installation.....	9
5 Obtaining Service Credentials.....	10
5.1 Service Subscription.....	10
5.2 Installation of Credentials.....	10
6 Configuring Server and Plugin.....	11
6.1 Plugin Factory Configuration.....	11
6.2 Logger Configuration.....	11
6.3 Azure Bot Plugin Configuration.....	11
7 Validating Setup.....	12
7.1 Setting up Sample LUIS App.....	12
7.2 Launching Server.....	12
7.3 Launching Client.....	13

1 Overview

This guide describes how to obtain and install binary packages for the Microsoft Azure Bot Service (ABS) plugin to the UniMRCP server on Debian-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 1.7.0 and above UniMRCP ABS Plugin 1.0.0 and above

1.2 Supported APIs

The following Azure APIs are supported by the plugin.

API	Availability
LUIS	ABS 1.0.0 and above
Service Bot (Direct Line Speech)	ABS 1.1.0 and above

1.3 Supported Distributions

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

Operating System	32-bit	64-bit
Ubuntu 16.04 LTS (xenial)		✓
Ubuntu 18.04 LTS (bionic)		✓

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

1.4 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 Deb Packages Using Apt-Get

Using the APT package handling utility (`apt-get`) is recommended for installation of UniMRCP binary packages.

2.1 Repository Configuration

The content of a typical configuration file of the APT repository, to be placed in `/etc/apt/sources.list.d/unimrcp.list`, is provided below.

```
deb [arch=amd64] https://username:password@unimrcp.org/repo/apt/ distr main
```



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

The *distr* field must be replaced with the corresponding distribution code name such as *xenial*, *bionic*, etc.

2.2 GnuPG Key

For verification of binary packages, UniMRCP provides a public GnuPG key, which can be retrieved and installed as follows.

```
wget -O - https://unimrcp.org/keys/unimrcp-gpg-key.public | sudo apt-key add -
```

2.3 Repository Update

In order to check for updates and apply the changes in the APT configuration, use the following command.

```
sudo apt-get update
```

2.4 Azure Bot Plugin Installation

In order to install the Azure Bot plugin, including all the dependencies, use the following command.

```
sudo apt-get install unimrcp-azure-bot
```

As a result, *apt-get* will check and prompt to download all the required packages by installing them in the directory */opt/unimrcp*.

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

```
sudo apt-get install umc-addons
```

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

3 Installing Deb Packages Manually

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

The deb packages have the following naming convention:

```
$packagename_$(universion)-$(distr)_$(arch).deb
```

where

- *packagename* is the name of a package
- *universion* is the UniMRCP version
- *distr* is the distribution code name (trusty, xenial, ...)
- *arch* is the architecture (amd64, i386, all, ...)

3.1 Package List

The following is a list of UniMRCP deb packages required for the installation of the Azure SR plugin.

Package Name	Description
unimrcp-azure-bot	Azure Bot plugin to the server.
uniazuresdk	UniMRCP edition of the Azure Speech SDK 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

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

```
sudo dpkg --install uniapr_${aprversion}-${distr}_${arch}.deb
sudo dpkg --install uniapr-util_${apuverson}-${distr}_${arch}.deb
sudo dpkg --install unisofia-sip_${sofiaversion}-${distr}_${arch}.deb
```

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

```
sudo dpkg --install unimrcp-common_${universion}-${distr}_${arch}.deb
sudo dpkg --install unimrcp-server_${universion}-${distr}_${arch}.deb
```

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

```
sudo dpkg --install unilicnodegen_${stoolversion}-${distr}_${arch}.deb
```

Next, a package containing the AWS SDK library.

```
sudo dpkg --install uniazuresdk_${azuresdkversion}-${distr}_${arch}.deb
```

Finally, a package containing the Azure Bot plugin should follow.

```
sudo dpkg --install unimrcp-azure-bot_${universion}-${distr}_all.deb
```


4 Obtaining License

The Azure Bot 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.

```
sudo /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*.

```
sudo cp umsazurebot_*.lic /opt/unimrcp/data
```

5 Obtaining Service Credentials

In order to utilize either the LUIS API and/or the Bot Service API, corresponding service subscription key and region need to be retrieved from the Microsoft Azure portal and further installed to the UniMRCP server.

5.1 Service Subscription

Navigate to the Microsoft Azure dashboard and create a new resource.

1. Navigate to the Dashboard.
<https://portal.azure.com>
2. [Create a LUIS app for intent recognition](#)
and/or
[Create a bot with the Direct Line Speech channel](#)
3. Collect one of the two keys (1) and the region (2).

5.2 Installation of Credentials

Create a text file `azbot.subscription.key` in the directory `/opt/unimrcp/data`.

```
sudo nano /opt/unimrcp/data/azbot.subscription.key
```

Place the collected key and the region in the following JSON format.

```
{
  "subscription-key": "*****",
  "region": "westus",
  "service-endpoint": "",
  "api": ""
}
```

Note that if the credentials are supposed to be used with a LUIS app, then the `api` token must be set to `luis`. Otherwise, for a bot service, the `api` token shall remain empty or be set to `dsc` (Direct Service Connector).

6 Configuring Server and Plugin

6.1 Plugin Factory Configuration

In order to load the Azure SR 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 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="mrcpreorder" enable="true"/>
  <engine id="Azure-Bot-1" name="umsazurebot" 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="AZUREBOT-PLUGIN" priority="INFO" masking="NONE"/>
```

6.3 Azure Bot Plugin Configuration

The configuration file of the plugin is located in */opt/unimrcp/conf/umsazurebot.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 LUIS App

This step is required if a LUIS app is used and can be skipped for bot service.

Create a new or use an existing LUIS application.

In order to reference the LUIS application, the corresponding App ID must be specified in the configuration file of the plugin, located in `/opt/unimrcp/conf/umsazurebot.xml`.

```
<streaming-recognition
  language="en-US"
  appid="*****_****_****_****_*****"
/>
```

Note that the application identifier can also be specified per recognition request. See the Usage Guide for more information.

7.2 Launching Server

Launch the UniMRCP server application as a service

```
sudo systemctl start unimrcp
```

Open the current log file located in `/opt/unimrcp/log/unimrcpserver_current.log`, and check whether the plugin is normally loaded.

```
[INFO] Load Plugin [Azure-Bot-1] [/opt/unimrcp/plugin/umsazurebot.so]
```

Next, check for the license information.

```
[NOTICE] UniMRCP AZUREBOT License

-product name:  umsazurebot
-product version: 1.0.0
-license owner:  -
-license type:  trial
-issue date:    2020-04-25
-exp date:     2020-05-25
```

```
-channel count: 2
-feature set: 0
```

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 gdf1* from the console of the *umc* client application.

```
run gdf1
```

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. Note that deepening on the referenced LUIS mode, the content may differ.

```
<?xml version="1.0"?>
<result>
  <interpretation grammar="builtin:speech/transcribe" confidence="0.777">
    <instance>
      <object><string name="query">book a room</string>
      <object name="topScoringIntent">
        <string name="intent">RoomReservation.Reserve</string>
        <number name="score">0.77710616600000004</number>
      </object>
      <array name="entities"></array>
    </object>
    </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 *umsazurebot.xml* and/or requested by the client.