

Powered by Universal Speech Solutions LLC



AWS Polly Plugin

Administrator Guide

Revision: 3

Distribution: Debian / Ubuntu

Created: August 22, 2018

Last updated: March 15, 2021

Author: Arsen Chaloyan

Table of Contents


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

1 Overview

This guide describes how to obtain and install binary packages for the Amazon Web Services (AWS) Polly speech synthesis 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 1.5.0 and above
UniMRCP Polly Plugin 1.0.0 and above

1.2 Supported Distributions


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

Operating System	Released	End of Support
Ubuntu 16.04 LTS (xenial)	August 2018	March 2021
Ubuntu 18.04 LTS (bionic)	May 2019	TBA
Ubuntu 20.04 LTS (focal)	March 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 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

Supply login information by creating a file `/etc/apt/auth.conf.d/unimrcp.conf` containing the following entry.

```
machine unimrcp.org
login username
password password
```

Note: the *username* and *password* fields must be replaced with the corresponding account credentials.

Configure a repository by creating a file `/etc/apt/sources.list.d/unimrcp.list` containing the following entry.

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

Note: the *distr* field must be replaced with the corresponding distribution code name such as *xenial*, *bionic*, *focal*, etc. To determine the distribution code, use ``lsb_release -cs``.

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 Polly Plugin Installation

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

```
sudo apt-get install unimrcp-polly
```

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 Polly plugin.

Package Name	Description
unimrcp-polly	AWS Polly plugin to the server.
uniawssdk	UniMRCP edition of the 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

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

```
sudo dpkg --install uniapr_${saprversion}-${distr}_${sarch}.deb
sudo dpkg --install uniapr-util_${sapuversion}-${distr}_${sarch}.deb
sudo dpkg --install unisofia-sip_${sofiaversion}-${distr}_${sarch}.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_${suniversion}-${distr}_${sarch}.deb
sudo dpkg --install unimrcp-server_${suniversion}-${distr}_${sarch}.deb
```

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

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

Next, a package containing the AWS SDK library.

```
sudo dpkg --install uniawssdk_${sawssdkversion}-${distr}_${sarch}.deb
```

Finally, a package containing the Polly plugin should follow.

```
sudo dpkg --install unimrcp-polly_${suniversion}-${distr}_all.deb
```

4 Obtaining License

The Polly 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 umspolly_*.lic /opt/unimrcp/data
```


5 Obtaining Service Credentials

In order to utilize the AWS Polly 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/polly/latest/dg/setting-up.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 Polly 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 synthesizer 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="false"/>
  <engine id="Demo-Recog-1" name="demorecog" enable="true"/>
  <engine id="Demo-Verifier-1" name="demoverifier" enable="true"/>
  <engine id="Recorder-1" name="mrcpreorder" enable="true"/>
  <engine id="Polly-1" name="umspolly" 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="POLLY-PLUGIN" priority="INFO" masking="NONE"/>
```

6.3 Polly Plugin Configuration

The configuration file of the plugin is located in */opt/unimrcp/conf/umspolly.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 Launching Server

Launch the UniMRCP server application.

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

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

```
[INFO] Load Plugin [Polly-1] [/opt/unimrcp/plugin/umspolly.so]
```

Next, check for the license information.

```
[NOTICE] UniMRCP Polly License  
  
-product name:  umspolly  
-product version: 1.0.0  
-license owner:  -  
-license type:   trial  
-issue date:    2018-08-22  
-exp date:      2018-09-21  
-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
```

7.2 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 synthesis scenario by issuing the command *run bss1* from the console of the *umc* client application.

```
run bss1
```

This command sends a SPEAK request to the server and then records synthesized stream into a PCM file stored in the directory */opt/unimrcp/var*.

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