

*Powered by Universal Speech Solutions LLC*



# RPM Installation Manual

## Administrator Guide

---

Revision: 54

Created: February 7, 2015

Last updated: May 20, 2017

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 RPMs Using YUM.....	4
2.1	Repository Configuration .....	4
2.2	Repository Verification.....	4
2.3	UniMRCP Client Installation.....	5
2.4	UniMRCP Server Installation.....	5
3	Installing RPMs Manually .....	6
3.1	Package List.....	6
3.2	Package Dependency Graph .....	7
3.3	Package Installation Order.....	7

# 1 Overview

This guide describes how to obtain and install UniMRCP binary packages 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 1.4.0 and above

## 1.2 Supported Distributions

UniMRCP binary packages are currently available only for x86\_64 (64-bit) architecture.

Operating System	32-bit	64-bit
Red Hat / Cent OS 6		✓
Red Hat / Cent OS 7		✓

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

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
```

where `unimrcp` is a name of the section 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
```

## 2.3 UniMRCP Client Installation

In order to install the UniMRCP client binaries, including the dependencies, the following command can be used.

```
yum install unimrcp-client
```

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

Similarly, for installation of development kit(s), the UniMRCP client libraries and header files, the following command may follow.

```
yum install unimrcp-client-devel
```

## 2.4 UniMRCP Server Installation

In order to install the UniMRCP server binaries, including the dependencies, the following command can be used.

```
yum install unimrcp-server
```

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

Similarly, for installation of development kit(s), the UniMRCP server libraries and header files, the following command may follow.

```
yum install unimrcp-server-devel
```

In order to install a package containing a set of demo plugins to the UniMRCP server, the following command can be used.

```
yum install unimrcp-demo-plugins
```

## 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:

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

where

- *package* 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

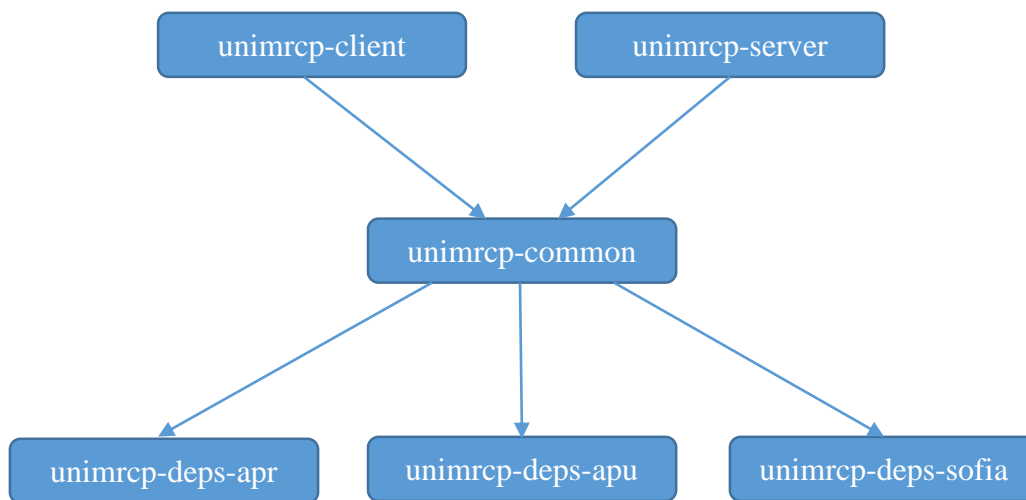
The following is a complete list of RPM packages UniMRCP provides.

Package Name	Description
<b>unimrcp-client</b>	Shared libraries and sample applications of the client.
<b>unimrcp-client-devel</b>	Development kit of the client.
<b>unimrcp-server</b>	Shared library and application of the server.
<b>unimrcp-server-devel</b>	Development kit of the server.
<b>unimrcp-demo-plugins</b>	Set of demo plugins to the server. [Optional]
<b>unimrcp-common</b>	Data common for the client and the server.
<b>unimrcp-common-devel</b>	Development kit of the common client and server data.
<b>uniapr</b>	UniMRCP edition of the Apache Portable Runtime (APR) library.
<b>uniapr-devel</b>	Development kit of the corresponding APR library.
<b>uniapr-util</b>	UniMRCP edition of the Apache Portable Runtime Utility (APR-Util) library.

<b>uniapr-util-devel</b>	Development kit of the corresponding APR-Util library.
<b>unisofia-sip</b>	UniMRCP edition of the Sofia SIP library.
<b>unisofia-sip-devel</b>	Development kit of the corresponding Sofia SIP library.

## 3.2 Package Dependency Graph

The following is a graph of package dependencies.



## 3.3 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$rhelversion.$sarch.rpm
rpm -ivh uniapr-util-$apuverson-$packageversion.el$rhelversion.$sarch.rpm
```

```
rpm -ivh unisofia-sip-$sofiaversion-$packageversion.el$rhelversion.$arch.rpm
```

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

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

Based on your requirements, either a client package, or a server package, or both of them can be installed next.

```
rpm -ivh unimrcp-client-$universion-$packageversion.el$rhelversion.$arch.rpm  
rpm -ivh unimrcp-server-$universion-$packageversion.el$rhelversion.$arch.rpm
```

Optionally, a package containing a set of demo plugins to the server may follow.

```
rpm -ivh unimrcp-demo-plugins-$universion-$packageversion.el$rhelversion.$arch.rpm
```

The same order should be considered for the installation of the corresponding development packages.