

SipXcom Voicemail to Email

This document provides a step by step guideline for setting up voicemail to email on SipXcom - this has been tested on SipXcom 4.6, 14.0x, and 14.10 releases. These steps have been validated with the following SMTP providers or email servers:

- [1and1.com](#)
- [gmail.com](#)
- Cablevision's [Optimum.net](#) service
- Outlook Exchange server with no authentication

This document is not meant to be exhaustive in terms of explaining the commands - there is a significant amount of Unix sendmail information on the Internet that provides further background.

Step 1 - Create EmailFormats.properties file on SipXcom

A fresh installation of SipXcom is not automatically configured for Voicemail to Email unified messaging. The voicemail to email content is triggered by the creation of the **EmailFormats.properties** file in the **/etc/sipxpbx/sipxivr** directory. **SSH** into the SipXcom **voice server** and create the **/etc/sipxpbx/sipxivr/EmailFormats.properties** file using your favorite Unix editor. Copy and paste the following lines into the file, save, and exit. Finally issue the **service sipxivr restart** command to get SipXcom to apply the changes.

⚠ Important - The **service sipxivr restart** command tells the sipxivr application to generate an email when voicemail is left for a user with unified messaging enabled.

/etc/sipxpbx/sipxivr/EmailFormats.properties

```
SenderName = Voicemail Notification Service
SenderMailto = support@lvttest.com
HtmlTitle = Voicemail Notification
PortalURL = {4}
Sender = {7} <{8}>
SubjectFull = {0,time,mm:ss} Voice Message from {2} ({3})
HtmlBodyFull = \
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"\n \
\"http://www.w3.org/TR/html4/strict.dtd">\n \
<p>See attached SipXcom Voicemail Message from {2} ({3}) left on {6,date,long} at {6,time,hh:mm:ss}. Duration
is {0,time,mm:ss} (mm:ss).</p>\n
```

The details regarding what each variable does can be found here if you wish to customize the voicemail message [Voicemail-Email Custom Notifications](#). The **EmailFormat.properties** file shown above will generate the following email when unified messaging is applied to a user extension.

From: Voicemail Notification Service [mailto:support@lvttest.com]
Sent: Friday, January 09, 2015 1:55 PM
To: Jane Smith
Subject: 00:03 Voice Message from 334 (LV Test)

See attached SipXcom Voicemail Message from 334 (LV Test) left on January 9, 2015 at 01:54:49. Duration is 00:03 (mm:ss).

Step 2 - Configure SipXcom User Extension for Unified Messaging

Go into SipXcom and configure the user extension for unified messaging per the attached diagram:

Make sure the email address is provisioned, voicemail notification is set to **E-mail notification**, E-mail format is set to **Full** (the other options were removed from the **EmailFormat.properties** template file found in the Sipfoundry wiki), and the Attach audio option is selected.

Step 3 - Get Authentication Credentials from SMTP Provider

Get authentication credentials from your SMTP provider - for this document **email@address.com** is the userid and **password** is the password. These authentication credentials will be provisioned in the **client-info** file for SMTP authentication.

⚠ Caution - be careful on the use of special symbols when creating email passwords for use by SMTP authentication. Some SMTP providers will respond with 'service unavailable' when the email password contains symbols not supported by their SMTP implementation.

Step 4 - Install the Cyrus SASL Library for SMTP Authentication - Optional

Unix applications such as **sendmail** use the Simple Authentication and Security Layer (SASL) from Cyrus for SMTP authentication. Some SMTP providers may use SASL for authentication - providers such as **Gmail** and **Optimum** will accept SASL when enabled but will authenticate SMTP messages without SASL. If SASL is required with an SMTP provider, issue the following commands on the SipXcom voice server.

- yum install cyrus-sasl-plain
- cd /etc/pki/tls/certs
- make sendmail.pem

Step 5 - Configure the client-info file for SMTP Authentication

The **client-info** file will be placed in a separate directory for ease of administration - the **sendmail.mc** file will point to the location of the client-info file (i.e. **/etc/mail/auth/client-info**) in Step 6. Issue the following commands:

- cd /etc/mail
- mkdir auth
- cd auth
- nano client-info

For **1and1.com** SMTP authentication, paste in the following replacing the **email@address.com** and **password** fields with the right credentials, save & close the file.

/etc/mail/auth/client-info for 1and1.com

```
AuthInfo:smtpland1.com "U:root" "I:email@address.com" "P:password" "M:PLAIN"
AuthInfo:smtpland1.com:587 "U:root" "I:email@address.com" "P:password" "M:PLAIN"
```

For **gmail.com** and **optimum.net** SMTP authentication, paste in the following replacing the **email@address.com** and **password** fields with the right credentials, save & close the file.

/etc/mail/auth/client-info for gmail.com

```
AuthInfo: "U:root" "I:email@address.com" "P:password"
```

Build the client-info database file to be used by sendmail for passing the authentication credentials by issuing the following command:

- `makemap hash client-info < client-info`

Step 6 - Modify the Sendmail Macro File for Authentication and Build the Sendmail Configuration File

The sendmail macro file ([sendmail.mc](#)) is found in the `/etc/mail/` directory. The file varies slightly depending on whether sendmail on the SipXcom server is talking to the [1and1.com](#) SMTP server, [gmail.com](#) SMTP server, Cablevision [optimum.net](#) SMTP server, or Microsoft Exchange SMTP server without authentication. Configuration files for [1and1](#), [gmail](#), and [optimum](#) are documented with and without SASL authentication using certificates.

 It is recommended that a backup copy of [sendfile.mc](#) file be created first e.g. `cp sendfile.mc sendfile.mcbak`

Select the appropriate provider and edit the `/etc/mail/sendmail.mc` file. The following statements should be inserted just above the first mailer instance in the default [sendmail.mc](#) file that is installed with a new SipXcom installation.

Where to insert Authentication Statements

```
dn1 FEATURE(masquerade_entire_domain)dn1
dn1 #
..... Insert Here .....
MAILER(smtp)dn1
MAILER(procmail)dn1
dn1 MAILER(cyrusv2)dn1
define(`confTO_DATAINIT', `lm')dn1 configured by SipXcom
define(`confTO_FILEOPEN', `lm')dn1 configured by SipXcom
```

For [1and1.com](#) without enabling SASL certificates, add the following statements **above** the first **"MAILER"** instance in the [sendmail.mc](#) file.

smtp.1and1.com without SASL

```
define(`SMART_HOST', `[smtp.1and1.com]')dn1
define(`RELAY_MAILER_ARGS', `TCP $h 587')dn1
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dn1
define(`confAUTH_OPTIONS', `A p')dn1
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dn1
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dn1
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dn1
```

For [1and1.com](#) with SASL authentication, add the following statements **above** the first **"MAILER"** instance in the [sendmail.mc](#) file. Enable the Cyrus mailer statement by removing the beginning `dn1` macro in the `'dn1 MAILER(cyrusv2) dn1'` statement (see code block).

smtp.1and1.com with SASL Authentication

```
define(`SMART_HOST', `[smtp.1and1.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
define(`CERT_DIR', `/etc/pki/tls/certs')
define(`confCACERT_PATH', `CERT_DIR')
define(`confCACERT', `CERT_DIR/ca-bundle.crt')
define(`confCRL', `CERT_DIR/ca-bundle.crt')
define(`confSERVER_CERT', `CERT_DIR/sendmail.pem')
define(`confSERVER_KEY', `CERT_DIR/sendmail.pem')
define(`confCLIENT_CERT', `CERT_DIR/sendmail.pem')
define(`confCLIENT_KEY', `CERT_DIR/sendmail.pem')
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
MAILER(cyrusv2)dnl
```

For [gmail.com](#) without enabling SASL certificates, add the following statements **above** the first "MAILER" instance in the [sendmail.mc](#) file.

smtp.gmail.com without SASL

```
define(`SMART_HOST', `[smtp.gmail.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dnl
```

For [gmail.com](#) with SASL authentication, add the following statements **above** the first "MAILER" instance in the [sendmail.mc](#) file. Enable the Cyrus mailer statement by removing the beginning `dnl` macro in the `dnl MAILER(cyrusv2) dnl` statement (see code block).

smtp.gmail.com with SASL Authentication

```
define(`SMART_HOST', `[smtp.gmail.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
define(`CERT_DIR', `/etc/pki/tls/certs')
define(`confCACERT_PATH', `CERT_DIR')
define(`confCACERT', `CERT_DIR/ca-bundle.crt')
define(`confCRL', `CERT_DIR/ca-bundle.crt')
define(`confSERVER_CERT', `CERT_DIR/sendmail.pem')
define(`confSERVER_KEY', `CERT_DIR/sendmail.pem')
define(`confCLIENT_CERT', `CERT_DIR/sendmail.pem')
define(`confCLIENT_KEY', `CERT_DIR/sendmail.pem')
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
MAILER(cyrusv2)dnl
```

For [optimum.net](#) without enabling SASL certificates, add the following statements **above** the first "MAILER" instance in the [sendmail.mc](#) file.

mail.optimum.net without SASL

```
define(`SMART_HOST', `[mail.optimum.net]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`SMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dnl
```

For [optimum.net](#) with SASL authentication, add the following statements **above** the first "**MAILER**" instance in the [sendmail.mc](#) file. Enable the Cyrus mailer statement by removing the beginning **dnl** macro in the `dnl MAILER(cyrusv2) dnl` statement (see code block).

mail.optimum.net with SASL Authentication

```
define(`SMART_HOST', `[mail.optimum.net]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`SMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
define(`CERT_DIR', `/etc/pki/tls/certs')
define(`confCACERT_PATH', `CERT_DIR')
define(`confCACERT', `CERT_DIR/ca-bundle.crt')
define(`confCRL', `CERT_DIR/ca-bundle.crt')
define(`confSERVER_CERT', `CERT_DIR/sendmail.pem')
define(`confSERVER_KEY', `CERT_DIR/sendmail.pem')
define(`confCLIENT_CERT', `CERT_DIR/sendmail.pem')
define(`confCLIENT_KEY', `CERT_DIR/sendmail.pem')
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash /etc/mail/auth/client-info')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
MAILER(cyrusv2)dnl
```

For **Microsoft Exchange** with no authentication or SASL certificates, add the following statements **above** the first "**MAILER**" instance in the [sendmail.mc](#) file.

Microsoft Exchange

```
define(`SMART_HOST', `[IP Address of Exchange server]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`SMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
```

Issue the `m4 sendmail.mc > sendmail.cf` command from the `/etc/mail` directory to build the sendmail configuration file from the macro file.

Finally issue the `service sendmail restart` command to pick up the new **sendmail** configuration file.

Error rendering macro 'contentbylabel'

parameters should not be empty