This Guide is intended to help users install and configure Open Meetings 2.X

The guide is aimed at Ubuntu 12.10 users, it has been written step by step with screenshots to aid in the successful build of OM from a fresh install.

SSL and Reverse proxy steps have been added but are optional.

N.B – When copying and pasting commands please check that symbols and character returns are correctly copied across.
Installing Ubuntu (Minimal Headless System)

Step 1: - Base System

Choose “Install Ubuntu Server”
Choose “United Kingdom”
Choose yes to automatically detect keyboard.

Enter a hostname.
Enter Username.

Enter a password for the new user.
Choose “no” to encrypt the Home Directory.

Choose yes to accept the detected time-zone.
Select “Guided – Use entire disk”

Accept the Disk selection to partition.
Choose “yes” to accept the changes to disk.

If you use a proxy server please enter that here, in most cases this is not needed and you can simply press enter to continue.
Select how you want to manage updates.

[!] Configuring tasksel

Applying updates on a frequent basis is an important part of keeping your system secure.

By default, updates need to be applied manually using package management tools. Alternatively, you can choose to have this system automatically download and install security updates, or you can choose to manage this system over the web as part of a group of systems using Canonical’s Landscape service.

How do you want to manage upgrades on this system?

- [x] No automatic updates
- Install security updates automatically
- Manage system with Landscape

---

[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

Choose software to install:

- [x] OpenSSH server
- [ ] DNS server
- [ ] LAMP server
- [ ] Mail server
- PostgresQL database
- [x] Print server
- [ ] Samba file server
- [ ] Tomcat Java server
- [ ] Virtual Machine host
- [ ] Manual package selection

[Continue]
Choose yes to install and configure the Grub Boot loader.

Press continue to reboot your system.
Step 2: - Setup SSH Environment

You should now be at the following screen, the next steps are easier done from a remote desktop using an SSH client such as putty. – But first we need to know our IP address, in most cases this was issued by your DHCP server (unless you specified manual network setup during install)
To find your IP address, first logon to your physical machine using root, then issue the following command:

```
ifconfig
```

This will show the following screen:

![ifconfig output]

You can see the IP Address in this case is 10.17.23.11 (Interface eth0)
You can now log off of the server.

![Putty client configuration]

From your desktop machine open your SSH client, in this case we will be using the putty client to connect to our new Server.

Enter the details and choose open.
The first log on you will receive this message; you can choose yes here and accept the key.

And finally this screen:

When using Ubuntu the root account has no password and by default is disabled, so first log on with the user that was created during the installation process (In this case omadmin), this account by default is part of the Admin group and therefore is also part of the sudoers group already, from here we can re-enable the root account.

To do this follow these steps

```
sudo passwd
```

Enter the omadmin password first

```
[sudo] password for omadmin: xxxxxx
```

Then enter the new root password twice

```
Enter new UNIX password:
Retype new UNIX password:
pwd: password updated successfully
```
We can now enter the following to change user to root

```
su -
```

Enter the password you set above and we are now ready to proceed.

### Step 3: Install Dependent software

First update the repos:

```
apt-get update
```

Now let’s create our work area

```
mkdir -p /usr/adm
```

We need to first install Libre Office:

```
apt-get install libreoffice -y
```

Libre Office installs OpenJDK so we need to install Oracle Java 6 JDK and update the alternatives – grab the relevant binary from oracle.

```
cd /usr/adm
```

Now issue the following to install it and correct the default java.

```
cd /usr/adm
chmod +x jdk-6u32-linux-x64.bin
./jdk-6u32-linux-x64.bin
mkdir -p /usr/lib/jvm
mv jdk1.6.0_32 /usr/lib/jvm/
update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.6.0_32/bin/javac 1
update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.6.0_32/bin/java 1
update-alternatives --install /usr/bin/javaws javaws /usr/lib/jvm/jdk1.6.0_32/bin/javaws 1
update-alternatives --config javac
update-alternatives --config java
update-alternatives --config javaws
```
Check java by issuing the following

```
java -version
```

```
java version "1.6.0_32"
Java(TM) SE Runtime Environment (build 1.6.0_32-b05)
Java HotSpot(TM) 64-Bit Server VM (build 20.7-b02, mixed mode)
```

```
ls -la /etc/alternatives/java*
```

and confirm the symbolic links point to the correct location.

The last stage of this step is to install the required dependencies for the OM install.

```
apt-get install imagemagick sox -y
apt-get install libgif-dev xpdf libfreetype6-dev libjpeg8 -y
apt-get install libjpeg8-dev libjpeg-dev libdirectfb-dev -y
apt-get install libbart-2.0-2 libt1-5 zip unZip bzip2 subversion git-core checkinstall -y
apt-get install yasm texi2html libfaac-dev libfaad-dev libmp3lame-dev libx11-dev -y
apt-get install libxfixes-dev libvorbis-dev -y
apt-get install libgsm1 libgsm1-dev -y
```

Step 4: Compile and Install SWFTools 2012-10-15-1307

```
cd /usr/adm
tar -zxvf swftools-2012-10-15-1307.tar.gz
cd swftools-2012-10-15-1307/
./configure
make
make install
```

Once that has completed you can now test it by issuing the following:

```
pdf2swf --version
```

Which should give you the following output:

```
pdf2swf - part of swftools 2012-10-15-1307
```
Step 5: Compile and Install ffmpeg 0.11.1

Let’s go back to our temporary working area

```
  cd /usr/adm
```

Download, compile and install ffmpeg by issuing these commands:

```
wget http://ffmpeg.org/releases/ffmpeg-0.11.1.tar.gz
tar -zxvf ffmpeg-0.11.1.tar.gz
cd ffmpeg-0.11.1
./configure --enable-libmp3lame --enable-libxvid --enable-libvorbis --enable-libgsm --enable-libfaac --enable-gpl --enable-nonfree
make
checkinstall
```

N.B - You may be asked a series of question towards the end of the install, press return for each to continue.

Once that has completed you can now test it by issuing the following:

```
ffmpeg -version
```

Which should give you the following output:

```
ffmpeg version 0.11.1
```

Step 6: - Create mysql DB for OM

Now we need to install MYSQL, issue this command (In this case username and password are openmeetings : ompassword)

```
apt-get install mysql-server -y
```

Enter the password as before “ompassword” and choose ok.
OpenMeetings 2.X Installation on Ubuntu 64bit 12.10

Stephen Cottham  22/01/2013

Now let’s create the needed DB’s for OM 2.x, issue the following commands:

```
mysql -u root -p
```

Enter password “**ompassword**”

Now issue these: (Assuming username **openmeetings** and password = **password**)

```
CREATE DATABASE openmeetings DEFAULT CHARACTER SET 'utf8';
GRANT ALL PRIVILEGES ON openmeetings.* TO 'openmeetings'@'localhost'
IDENTIFIED BY 'password' WITH GRANT OPTION;
quit
```

Successful DB creation shown above.
Step 7: Install JOD Converter

Let’s go back to our temporary working area

cd /usr/adm

Download, extract JOD by issuing these commands: *(We will move the JOD location after the installation of OM 2.x)*

```
wget http://jodconverter.googlecode.com/files/jodconverter-core-3.0-beta-4-dist.zip
unzip jodconverter-core-3.0-beta-4-dist.zip
```

Step 8: Install ANT 1.8.4 for compiling latest OM 2.x

Let’s go back to our temporary working area

cd /usr/adm

Download, extract ANT by issuing these commands:

```
wget http://mirror.catn.com/pub/apache//ant/binaries/apache-ant-1.8.4-bin.tar.gz
tar -zxvf apache-ant-1.8.4-bin.tar.gz
```

Once that has completed you can test it by issuing the following commands:

```
cd /usr/adm/apache-ant-1.8.4/bin
./ant -version
```

This should output the following:

*Apache Ant(TM) version 1.8.4 compiled on May 22 2012*
Step 9: Download and compile latest OM 2.1

Again back to our working area:

```
cd /usr/adm
```

Then check out the latest source code using the following:

```
svn checkout https://svn.apache.org/repos/asf/incubator/openmeetings/trunk/singlewebapp/
```

Once that has completed we can then build the source by issuing the following:

```
cd /usr/adm/singlewebapp
/usr/adm/apache-ant-1.8.4/bin/ant clean.all
/usr/adm/apache-ant-1.8.4/bin/ant -Ddb=mysql
```

This will take a little while depending on your system, once it has finished you should be left the following message:

**BUILD SUCCESSFUL**

Step 9a: Install pre-built OM 2.x (Alternative to Step 9)

Download the latest build from the following links:

2.0

[https://builds.apache.org/view/M-R/view/OpenMeetings/job/OpenMeetings%202.0/lastSuccessfulBuild/artifact/2.0/dist/](https://builds.apache.org/view/M-R/view/OpenMeetings/job/OpenMeetings%202.0/lastSuccessfulBuild/artifact/2.0/dist/)

Or

2.1

[https://builds.apache.org/job/openmeetings/lastSuccessfulBuild/artifact/singlewebapp/dist/](https://builds.apache.org/job/openmeetings/lastSuccessfulBuild/artifact/singlewebapp/dist/)

The file will be something like the following “apache-openmeetings-incubating-2.xxxxx.tar.gz:
(Where xxx is the date and build version)

We can do this using wget, so first we need to go back to our build area like so:

```
cd /usr/adm
mkdir -p singlewebapp/dist/red5
cd singlewebapp/dist/red5
```

Then grab the file and extract it:

```
wget https://builds.apache.org/job/openmeetings/lastSuccessfulBuild/artifact/singlewebapp\dist/apache-openmeetings-incubating-2.xxxxx.tar.gz
```
OpenMeetings 2.X Installation on Ubuntu 64bit 12.10

Stephen Cottham  22/01/2013

tar -zxvf apache-openmeetings-incubating-2.xxxx.tar.gz

Now download the mysql connector from here:

http://www.mysql.com/downloads/connector/j/

cd /usr/adm/singlewebapp/dist/red5/webapps/openmeetings/WEB-INF/lib


unzip mysql-connector-java-5.1.20.zip

cd mysql-connector-java-5.1.20

mv mysql-connector-java-5.1.20-bin.jar
/usr/adm/singlewebapp/dist/red5/webapps/openmeetings/WEB-INF/lib

Step 10: Install compiled\Pre-Built OM 2.x

Now we need to move the compiled source into the correct location, in this system we are using /usr/lib/red5, so issue the following commands to move the root folder over:

cd /usr/adm/singlewebapp/dist
mv red5/ /usr/lib/

Let’s move the JOD into place now


And set some permissions and ownerships

chown -R nobody /usr/lib/red5
chmod +x /usr/lib/red5/red5.sh
chmod +x /usr/lib/red5/red5-debug.sh

Set the start-up script for OM 2.x by issuing the following:

vi /etc/init.d/red5

and adding the following:

#!/bin/sh
### BEGIN INIT INFO
# Provides: red5
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Starts red5 server for Openmeetings.
### END INIT INFO
OpenMeetings 2.X Installation on Ubuntu 64bit 12.10

Stephen Cottham  22/01/2013

# For RedHat and cousins:
# chkconfig: 2345 85 85
# description: Red5 flash streaming server for OpenMeetings
# processname: red5
# Created By: Sohail Riaz (sohaileo@gmail.com)
# Modified by Alvaro Bustos
PROG=red5
RED5_HOME=/usr/lib/red5
DAEMON=$RED5_HOME/PROG.sh
PIDFILE=/var/run/$PROG.pid

[ -r /etc/sysconfig/red5 ] && . /etc/sysconfig/red5
RETV AL=0

case "$1" in
  start)
    cd $RED5_HOME
    start-stop-daemon --start -c nobody --pidfile $PIDFILE \  
    --chdir $RED5_HOME --background --make-pidfile \  
    --exec $DAEMON >/dev/null 2>/dev/null &
    RETVAL=$?
    if [ $RETV AL -eq 0 ]; then
      echo $! > $PIDFILE
    fi
    echo
  ;;
  stop)
    start-stop-daemon --stop --quiet --pidfile $PIDFILE \  
    --name java
    rm -f $PIDFILE
    echo
    [ $RETV AL -eq 0 ] && rm -f /var/lock/subsys/$PROG
  ;;
  restart|force-reload)
    $0 stop
    $0 start
  ;;
  status)
    # Debian and Ubuntu 10 status check
    ps aux | grep -f $PIDFILE >/dev/null 2>/dev/null && RETVAL=0 || RETVAL=3
    # Ubuntu 12 status check using improved "start-stop-daemon" status query
    # (use the above command, or comment out above command and uncomment the two
    # below commands.
    # start-stop-daemon --status --pidfile $PIDFILE
    # RETVAL=$?
    [ $RETV AL -eq 0 ] && echo "$PROG is running"
    [ $RETV AL -eq 1 ] && echo "$PROG is not running and the pid file exists"
    [ $RETV AL -eq 3 ] && echo "$PROG is not running"
    [ $RETV AL -eq 4 ] && echo "$PROG - unable to determine status"
  ;;
  checkports)
    netstat -anp | grep soffice
    netstat -anp | grep java
  ;;

echo "$Usage: $0 {start|stop|restart|force-reload|status|checkports}" RETVAL=1
esac
exit $RETV AL

Save the file and then set the permissions like below:

chmod +x /etc/init.d/red5
update-rc.d red5 defaults

Now we need to move the persistence files so we can connect to mysql, so issue the following:

Make backup copy

mv /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/
persistence.xml /usr/lib/red5/webapps/openmeetings/WEB-INF/
classes/META-INF/persistence.xml-ori

Rename mysql template to persistence.xml

mv /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/
mysql_persistence.xml /usr/lib/red5/webapps/openmeetings/WEB-INF/
classes/META-INF/persistence.xml

Edit the persistence file and add our mysql details, in this case we used "openmeetings" and "password" – so issue the following:

vi /usr/lib/red5/webapps/openmeetings/WEB-INF/classes/META-INF/persistence.xml

Then change the following

, Username=openmeetings
, Password=password"/>

At this stage we are ready to start up OM 2.x for the first time.

/etc/init.d/red5 start

Now open the browser and go to the following link. N.B remember to change the IP address to your OM2.x server, the one below 10.17.23.11 is just for this example.

http://10.17.23.11:5080/openmeetings/install
Choose the “Continue with STEP 1” link
The only section we need to fill out at this stage is the following:

Username: omadmin
Userpass: ompassword
Email: something@something.com
TimeZone: United Kingdom
Domain Name: somedomain

Now click on INSTALL at the bottom of the page, this will then create all the needed tables etc. - it can take a little while but be patient.

Once that has completed you can now enter the application by clicking on the “Enter the Application” link

You should see the following logon screen:

Enter these details to sign in.

Username: omadmin
Userpass: ompassword
Step 11: Add relevant paths to the configuration

Once logged in go to Administration > Configuration

![Apache OpenMeetings](image)

You will see on the left hand pane a list of keys and values, the ones we are interested in are:

- SWFTools Path: `/usr/local/bin/`
- ImageMagick Path: `/usr/bin/`
- FFMPEG Path: `/usr/local/bin`
- SoX Path: `/usr/bin`

Click on the left hand pane option and then enter the value as above, click on the save button to apply the changes; once you have done each key you should see the following:
JOD will find open office in this case so we do not need to set the path.
OpenMeetings 2.X Installation on Ubuntu 64bit 12.10

Step 12: Securing OpenMeetings using encryption (Optional)

12.1 - Generating CSR:

We can do this in a few ways, the first way I will show here is simply by generating a CSR and inserting these into OpenMeetings.

Create a new keystore and key, use the same password for both: (Taken from OM Website http://incubator.apache.org/openmeetings/RTMPSAndHTTPS.html)

```bash
keytool -keysize 2048 -genkey -alias red5 -keyalg RSA -keystore red5/conf/keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: <your hostname, e.g demo.openmeetings.de>
What is the name of your organizational unit?
[Unknown]: Dev
What is the name of your organization?
[Unknown]: OpenMeetings
What is the name of your City or Locality?
[Unknown]: Henderson
What is the name of your State or Province?
[Unknown]: Nevada
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=demo.openmeetings.de, OU=Dev, O=OpenMeetings, L=Henderson, ST=Nevada, C=US correct?
[no]: yes
Enter key password for <red5>
Generate a CSR:

keytool -certreq -keyalg RSA -alias red5 -file red5.csr -keystore red5/conf/keystore
```

Submit CSR to your CA of choice and receive a signed certificate
Import your chosen CA's root certificate into the keystore (may need to download it from their site - make sure to get the root CA and not the intermediate one)

```bash
keytool -import -alias root -keystore red5/conf/keystore -trustcacerts -file root.crt
```

(note: you may receive a warning that the certificate already exists in the system wide keystore - import anyway)

Import the intermediate certificate(s) you normally receive with the certificate:

```bash
keytool -import -alias intermed -keystore red5/conf/keystore -trustcacerts -file intermediate.crt
```

Import the certificate you received:

```bash
keytool -import -alias red5 -keystore red5/conf/keystore -trustcacerts -file demo.openmeetings.de.crt
```
12.2 – Using Existing certs such as wild card certificates instead of generating a new CSR.

First let’s go back to our work area:

cd /usr/adm/
mkdir certs
cd certs/

Using WinSCP or equivalent copy your wild card key and cert files: yourdomain.key.pem and yourdomain.cert.pem - *(These should be in PEM format)*

Now issue the following to convert the files to DER format

```
openssl pkcs8 -topk8 -nocrypt -in apache.key.pem -inform PEM -out key.der -outform DER
openssl x509 -in apache.cert.pem -inform PEM -out cert.der -outform DER
```

Now we need a couple of files to help us import the DER files into the keystore, so issue the following:

```
wget http://www.agentbob.info/agentbob/80/version/default/part/AttachmentData/data/ImportKey.java
wget http://www.agentbob.info/agentbob/81/version/default/part/AttachmentData/data/ImportKey.class
```

Then use these commands to import:

```
java ImportKey key.der cert.der
```

Finally move the keystore to the correct location

```
mv /root/keystore.ImportKey /usr/lib/red5/conf/keystore
```

N.B = Alias:importkey  Password:importkey (When using the java import key files, you can change the password afterwards)
Now that we have either a new Cert of the wild card cert inside our Keystore we need to make some changes to OM 2.x to use these certificates and thus encrypt communications using HTTPS and RTMPS.

To use RTMPS do the following:
First make some changes to the red5-core.xml file by issuing the following:

```bash
cd /usr/lib/red5/conf
vi red5-core.xml
```

now uncomment <!-- RTMPS --> section by removing the <!-- and the --> leaving this:

```xml
<bean id="rtmpsMinaIoHandler"
     class="org.red5.server.net.rtmps.RTMPSMinaIoHandler">
  <property name="handler" ref="rtmpHandler" />
  <property name="codecFactory" ref="rtmpCodecFactory" />
  <property name="rtmpConnManager" ref="rtmpMinaConnManager" />
  <property name="keyStorePassword" value="${rtmps.keystorepass}" />
  <property name="keystoreFile" value="conf/keystore" />
</bean>

<bean id="rtmpsTransport" class="org.red5.server.net.rtmp.RTMPMinaTransport" init-method="start" destroy-method="stop">
  <property name="ioHandler" ref="rtmpsMinaIoHandler" />
  <property name="connectors">
    <list>
      <bean class="java.net.InetSocketAddress">
        <constructor-arg index="0" type="java.lang.String" value="${rtmps.host}" />
        <constructor-arg index="1" type="int" value="${rtmps.port}" />
      </bean>
    </list>
  </property>
  <property name="ioThreads" value="${rtmp.io_threads}" />
  <property name="jmxPollInterval" value="1000" />
  <property name="tcpNoDelay" value="${rtmp.tcp_nodelay}" />
</bean>
```

Save this file and then do the following:

```bash
cd /usr/lib/red5/conf
vi red5.properties
```

set rtmps.port=5443
rtmps.keystorepass=password (password = password you set on your new keystore)

Now edit config.xml by doing the following:

```bash
cd /usr/lib/red5/webapps/openmeetings/
vi config.xml
```
Set these following values:

<rtmppsslport>5443</rtmppsslport>
<useSSL>yes</useSSL>
<proxyType>best</proxyType>

**To use HTTPS do the following:**

First make a backup of the original jee-container file by doing the following:

```
ed /usr/lib/red5/conf
mv jee-container.xml jee-container.xml.orig
```

Then rename the SSL jee template

```
mv jee-container-ssl.xml jee-container.xml
```

Now edit the config.xml

```
ed /usr/lib/red5/webapps/openmeetings/
vi config.xml
```

set

```
<protocol>https</protocol>
<red5httpport>443</red5httpport>
```

Lastly edit red5.properties by doing the following:

```
ed /usr/lib/red5/conf
vi red5.properties
```

set

```
https.port=443
http.port=443
```

Now restart OM using the following:

```
/etc/init.d/red5 restart
```

We can now connect using the following link:

https://yourdomain/openmeetings
Step 13: Installing Reverse Proxy using Apache Web Server (Optional)

Another way to secure the OpenMeetings service is to use Apache as a reverse proxy, to do this we need to do the following:

First install Apache2 and enabling relevant modules by running the following commands:

```
apt-get install apache2
a2enmod proxy
a2enmod proxy_http
a2enmod ssl
a2enmod headers
a2enmod rewrite
a2enmod cache
/etc/init.d/apache2 restart
```

We can now redirect port 80 (less secure) or port 443 (secure) to port 5080, to do this we need to create a virtual host, to do this do the following:

```
cd /etc/apache2/sites-enabled/
```

Now for SSL redirect (**using a Cert on Apache instead of keystore**) do the following

```
vi om.yourdomain.com-ssl
```

and add the following

```
<IfModule mod_ssl.c>
#NameVirtualHost *:443
ProxyRequests Off
<VirtualHost *:80>
    ServerAdmin hostmaster@domain.com
    ServerName om.yourdomain.com
    ProxyPreserveHost On
    RewriteEngine on
```

```
</VirtualHost>
```