Debian Post-Install Tasks

About

After the final "make" command to build FreeSWITCH™ from source, you might want to perform these tasks to set permissions and ownership of files and directories.

Starting FreeSWITCH

If you start FreeSWITCH for the first time without setting permissions and file ownership, the FS databases and log files will be created as user "root". Then when you change owner later FS will no longer have permission to write to these files and will fail.

The default and preferred init system on Debian 8 (and CentOS 7) is Systemd. If you didn't revert to System V init and you're compiling Freeswitch from source code, the systemd unit file will help you get close to a working FreeSWITCH system.

Set Owner and Permissions

Because you most likely built FreeSWITCH™ as root you must now create the user 'freeswitch' in group 'freeswitch' and change the FS files to owner = freeswitch and group = freeswitch and set the file mode appropriately. Execute the following commands as user root:

Set owner and permissions

```
# create user 'freeswitch'
# add it to group 'freeswitch'
# change owner and group of the freeswitch installation
cd /usr/local
groupadd freeswitch
adduser --quiet --system --home /usr/local/freeswitch --gecos "FreeSWITCH open source softswitch" --ingroup freeswitch freeswitch --disabled-password
chown -R freeswitch:freeswitch /usr/local/freeswitch/
chmod -R ug=rwX,o= /usr/local/freeswitch/
chmod -R u=rwx,g=rx /usr/local/freeswitch/bin/*
```

systemd

systemd is the service management system that replaces System V init. It is quite thorough and requires much simpler configuration scripts called **Unit Files**. systemd can start FreeSWITCH™ at boot time, monitor the application, restart it if it fails, and take other useful actions.

At the time of this writing (August 2015) the optimum systemd unit file is not certain, but the following have been proven to work on Debian 8.1 systems. If you have discovered a better approach, please update this section or submit a JIRA ticket for the DOCS project so that we can capture this knowledge for the community. If you prefer to use the old sysvinit system, then use the instructions on the Debian 7 page.

The systemd unit file you're going to use should be something similar to this. The pre-compiled packages come with their own systemd files and scripts to install them, although you may certainly use this as a guide if you have trouble; but be sure to observe the correct file locations as they differ between FreeSWITCH™ compiled installations and package installations.

When troubleshooting systemd difficulties look for messages in `/var/log/syslog` and `/var/log/daemon.log` and grep for freeswitch messages.

The unit file starts Freeswitch as root, so Freeswitch can set its priority, create some directories if needed, etc. Then it will drop privileges to continue running as user and group Freeswitch.

systemd unit file FreeSWITCH built from MASTER

```
; This file in installations built from Master can be found in
; /usr/src/freeswitch.git/debian
; or
; /usr/src/freeswitch/debian
[Unit]
Description=freeswitch
After=syslog.target network.target local-fs.target

[Service]
; service
Type=forking
PIDFile=/usr/local/freeswitch/run/freeswitch.pid
PermissionsStartOnly=true
; blank ExecStart= line flushes the list
ExecStart=
ExecStart=/usr/local/freeswitch/bin/freeswitch -u freeswitch -g freeswitch -ncwait -nonat -rp
TimeoutSec=45s
Restart=on-failure
; exec
WorkingDirectory=/usr/local/freeswitch/bin
User=root
Group=daemon
LimitCORE=infinity
LimitNOFILE=100000
LimitNPROC=60000
;LimitSTACK=240
LimitRTPRIO=infinity
LimitRTTIME=7000000
IOSchedulingClass=realtime
IOSchedulingPriority=2
CPUSchedulingPolicy=rr
CPUSchedulingPriority=89
UMask=0007

[Install]
WantedBy=multi-user.target
```

Now that the systemd unit file has been set up, you can **copy it** (do not link it) to the /etc/systemd/system directory where systemd looks for local customized files that won't be overwritten by systemd updates.

```
cp /usr/src/freeswitch/debian/freeswitch-systemd.freeswitch.service /etc/systemd/system/freeswitch.service
```

Next, notify systemd to look for changes in its configuration:

```
systemctl daemon-reload
```

Now you are ready to start FreeSWITCH for the first time. Because FS is owned by user freeswitch it should create its databases and log files under that user (instead of root) so that those files will still be accessible to it under future runs.

```
systemctl start freeswitch
```

To stop FreeSWITCH:

```
systemctl stop freeswitch
```

To configure FreeSWITCH to start automatically at boot time:

```
systemctl enable freeswitch
```

Scheduler Priority

if your filesystem supports extended attributes, execute

```
setcap 'cap_net_bind_service,cap_sys_nice=+ep' /usr/local/freeswitch/bin/freeswitch
```

to allow FreeSWITCH to increase its nice level and also to allow socket binding on low ports.

If setcap is not available, remove the -rp option from ExecStart and add these lines to the unit file above to give real-time priority to the process:

```
PermissionsStartOnly=true
```

```
ExecStartPost=/usr/bin/chrt -f -p 1 $MAINPID
```

Now you can start FreeSWITCH™ for the first time to test.

Is FreeSWITCH Running?

To determine if FreeSWITCH is actually running, use one of these commands:

```
  ps aux | grep freeswitch
```

```
  ps -e | grep freeswitch
```

Either of the above should display a line beginning with the pid (process id) of freeswitch if it is indeed running. Ignore the line that matches the grep command since it also contains the string "freeswitch".

```
  pidof freeswitch
```

pidof returns the process id of the named process. In this case, if FreeSWITCH is running you will see only its pid; if it prints nothing at all, then FreeSWITCH is not running.

DNS caching

By default Debian has no DNS caching and every lookup goes to server from /etc/resolv.conf. **Unbound** is light, secure and easy to use DNS caching server.

```
apt -y install unbound   # installation
systemctl start unbound  # start
systemctl enable unbound # auto start
```

```
prepend domain-name-servers 127.0.0.1;
```

line of

```
/etc/dhcp/dhclient.conf
```

should be **uncommented** to preserve changes of /etc/resolv.conf.

Proper entropy source

Most of Debian systems may not have sufficient entropy for proper cryptography operations. **Haveged** is a userspace entropy daemon which is not dependent upon the standard mechanisms for harvesting randomness for the system entropy pool.

```
apt -y install haveged   # installation
systemctl start haveged  # start
systemctl enable haveged # auto start
```

Automatic time synchronization

Chrony is fast, secure and easy to use time synchronization daemon.

```
apt -y install chrony   # installation
systemctl start chrony  # start
systemctl enable chrony # auto start
```