Examples of `/etc/init.d/freeswitch`

Unix-style `/etc/init.d/freeswitch`

The various unix-based systems often start FreeSWITCH from a script that lives in `/etc/init.d/freeswitch`.

**Linux**

**Debian/Ubuntu**

You should only need to use this if you are not installing from a pre-built package.

The existing start-up scripts can be found in the `src/freeswitch/debian/` directory, however these are tied in closely with the layout of the package builder and may not be suitable for your needs.

Below is a fully functional init.d script which you can use with any standard FreeSWITCH build (assuming you haven't changed --prefix).

**Adding users / fixing permissions**

Add the user and group to your system, and change the required permissions for the directory; replace with your installation location as required:

```
adduser --disabled-password --quiet --system --home /usr/local/freeswitch --gecos "FreeSWITCH Voice Platform" --ingroup daemon freeswitch
chown -R freeswitch:daemon /usr/local/freeswitch/
chmod -R o-rwx /usr/local/freeswitch/
```

**Installing the script**

Create the file `/etc/init.d/freeswitch` with the following code:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          freeswitch
# Required-Start:    $local_fs $remote_fs
# Required-Stop:     $local_fs $remote_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Description:       Freeswitch debian init script.
# Author:            Matthew Williams
#
### END INIT INFO
# Do NOT "set -e"
# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin
DESC="FreeSWITCH"
NAME=freeswitch
DAEMON=/usr/local/freeswitch/bin/$NAME
DAEMON_ARGS="--nc"
PIDFILE=/usr/local/freeswitch/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME

FS_USER=freeswitch
FS_GROUP=daemon

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Read configuration variable file if it is present
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

# Load the VERBOSE setting and other rcS variables
./lib/init/vars.sh
```

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
.
.
# Function that sets ulimit values for the daemon
#
# do_setlimits() {
ulimit -c unlimited
ulimit -d unlimited
ulimit -f unlimited
ulimit -i unlimited
ulimit -n 999999
ulimit -q unlimited
ulimit -u unlimited
ulimit -v unlimited
ulimit -x unlimited
ulimit -s 240
ulimit -l unlimited
return 0
}

# Function that starts the daemon/service
#
# do_start() {


# Set user to run as
if [ $FS_USER ]; then
DAEMON_ARGS="`echo $DAEMON_ARGS -u $FS_USER""
fi

# Set group to run as
if [ $FS_GROUP ]; then
DAEMON_ARGS="`echo $DAEMON_ARGS -g $FS_GROUP""
fi

# Return
#   0 if daemon has been started
#   1 if daemon was already running
#   2 if daemon could not be started
#   other if a failure occurred
start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test > /dev/null -- \
| | return 1
| do_setlimits
start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --background -- \
| $DAEMON_ARGS \n| | return 2
# Add code here, if necessary, that waits for the process to be ready
# to handle requests from services started subsequently which depend
# on this one. As a last resort, sleep for some time.

#
# Function that stops the daemon/service
#
# do_stop() {

# Return
#   0 if daemon has been stopped
#   1 if daemon was already stopped
#   2 if daemon could not be stopped
#   other if a failure occurred
start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE --name $NAME RETIVAL="$?"
| "$RETVAL" = 2 } && return 2
# Wait for children to finish too if this is a daemon that forks
# and if the daemon is only ever run from this initscript.
# If the above conditions are not satisfied then add some other code
# that waits for the process to drop all resources that could be
# needed by services started subsequently. As a last resort is to
# sleep for some time.
start-stop-daemon --stop --quiet --oknode --retry=0/30/KILL/5 --exec $DAEMON
[ "$?" = 2 ] && return 2
# Many daemons don't delete their pidfiles when they exit.
rm -f $PIDFILE
return "$RETVAL"
}

# Function that sends a SIGHUP to the daemon/service
# do_reload() {
# # If the daemon can reload its configuration without
# # restarting (for example, when it is sent a SIGHUP),
# # then implement that here.
# # start-stop-daemon --stop --signal 1 --quiet --pidfile $PIDFILE --name $NAME
# return 0
#}

case "$1" in
  start)
    [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
    do_start
    case "$?" in
      0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
      2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  stop)
    [ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
    do_stop
    case "$?" in
      0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
      2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  status)
    status_of_proc -p $PIDFILE $DAEMON $NAME && exit 0 || exit $?
    ;;
  reload|force-reload)
    # If do_reload() is not implemented then leave this commented out
    # and leave 'force-reload' as an alias for 'restart'.
    #
    log_daemon_msg "Reloading $DESC" "$NAME"
    do_reload
    log_end_msg $?
    ;;
  restart|force-reload)
    # If the 'reload' option is implemented then remove the
    # 'force-reload' alias
    #
    log_daemon_msg "Restarting $DESC" "$NAME"
    do_stop
    case "$?" in
      0|1)
        do_start
        case "$?" in
          0) log_end_msg 0 ;;
          1) log_end_msg 1 ;; # Old process is still running
          *) log_end_msg 1 ;; # Failed to start
        esac
        ;;
      *
      *)
        # Failed to stop
        log_end_msg 1
        ;;
    esac
    ;;
  *)
    # Failed to start
    log_end_msg 1
    ;;
 esac
Make the script executable and make it auto start on system boot:

```bash
chmod +x /etc/init.d/freeswitch
update-rc.d freeswitch defaults
```

### Run priority

You can modify the run priority in two places.

The first is using `DAEMON_ARGS`, using the arguments explained [FreeSwitch FAQ: How do I setup high priority to my freeswitch daemon](https://www.freeswitch.org/FreeSwitch_FAQ#Q:_How_do_I_setup_high_priority_to_my_freeswitch_daemon), for example:

```
DAEMON_ARGS="-nc -np" - normal (system) default priority
DAEMON_ARGS="-nc -lp" - low priority
DAEMON_ARGS="-nc -rp" - high real time priority
```

The second is to modify the `--nicelevel` of start-stop-daemon, for example:

```
... 
DAEMON_ARGS="-nc -np" - normal (system) default priority 
... 
start-stop-daemon --start --quiet --nicelevel -19 --pidfile $PIDFILE --exec $DAEMON --test > /dev/null 
-- 
   || return 1 
do_setlimits 
start-stop-daemon --start --quiet --nicelevel -19 --pidfile $PIDFILE --exec $DAEMON -- 
   $DAEMON_ARGS \ 
   || return 2 
...
```

### Fedora

*Note: Also tested on CentOS 5.5 with perfect results, just as always set chmod +x /etc/init.d/freeswitch*

This script runs FreeSWITCH as root (NOT RECOMMENDED)

Use the `-u USER` option to switch to a different user on startup. (see SECURITY NOTE in the script)

Be sure to run:

```bash
chkconfig freeswitch on
```

So that the script is added to all the rc#.d directories.

```
/etc/init.d/freeswitch
```

```bash
#!/bin/sh
#
# freeswitch: Starts the freeswitch Daemon
#
# chkconfig: 345 96 02
# processname: freeswitch
# description: Freeswitch fedora init script 
# config:
```
# Author: gled

# Source function library.
. /etc/init.d/functions
. /etc/sysconfig/network

PATH=/sbin:/usr/sbin:/bin:/usr/local/freeswitch/bin
DESC="FreeSwitch Voice Switching System"
NAME=freeswitch
DAEMON=/usr/local/freeswitch/bin/$NAME
DAEMON_ARGS="-nc"
PIDFILE=/usr/local/freeswitch/log/$NAME.pid

## SECURITY NOTE: To run as non-root, create a new user for FreeSWITCH and set these variables (FS_GROUP is optional).
##
#FS_USER=freeswitch
#FS_GROUP=freeswitch

do_setlimits() {
    ulimit -c unlimited
    ulimit -d unlimited
    ulimit -f unlimited
    ulimit -i unlimited
    ulimit -n 999999
    ulimit -q unlimited
    ulimit -u unlimited
    ulimit -v unlimited
    ulimit -x unlimited
    ulimit -s 244
    ulimit -l unlimited
    return 0
}

base=${0##*/}

do_start() {
    if [ -n "${FS_USER}" ]; then
        DAEMON_ARGS="${DAEMON_ARGS} -u ${FS_USER}"
        fi
    if [ -n "${FS_GROUP}" ]; then
        DAEMON_ARGS="${DAEMON_ARGS} -g ${FS_GROUP}"
        fi
    do_setlimits
    $DAEMON $DAEMON_ARGS
    RETVAL=$?
    if [ $RETVAL = 0 ]; then
        success "$base startup"
    else
        failure "$base startup"
    fi
    echo
    return $RETVAL
}

do_stop() {
    $DAEMON -stop
    RETVAL=$?
    if [ $RETVAL = 0 ] && success "$base shutdown" || failure "$base shutdown"
    rm -f $LOCKFILE
    echo
    return $RETVAL
}

# See how we were called.
case "$1" in
  start)
    do_start
  ;;
stop)
    do_stop
    ;;
restart)
    do_stop
    echo "Waiting for daemon to exit..."
    sleep 5
    do_start
    ;;
*)
    echo "$0 Usage: $0 {start|stop}"
    exit 2
    ;;
esac

exit $RETVAL

OpenSUSE

OpenSUSE /etc/init.d/freeswitch

#!/bin/sh
### BEGIN INIT INFO
# Provides: Freeswitch
# Required-Start: $network $firewall
# Required-Stop:
# Default-Start: 3 5
# Default-Stop: 0 1 2 6
# Short-Description: Freeswitch Back-to-Back User Agent
# Description: Start the Freeswitch Back-to-Back User Agent to provide VOIP/Multimedia gateway services
### END INIT INFO

FREESWITCH_BIN=/usr/local/freeswitch/bin/freeswitch
FREESWITCH_CONFIG=/usr/local/freeswitch/conf/freeswitch.xml

# Function that sets ulimit values for the daemon
# do_setlimits() {
# ulimit -c unlimited
ulimit -d unlimited
ulimit -f unlimited
ulimit -i unlimited
ulimit -n 999999
ulimit -q unlimited
ulimit -u unlimited
ulimit -v unlimited
ulimit -x unlimited
ulimit -s 240
ulimit -l unlimited
return 0
}

. /lib/lsb/init-functions
. /etc/rc.status
rc_reset

case "$1" in
    start)
echo -n "Starting Freeswitch "
do_setlimits
/sbin/startproc $FREESWITCH_BIN -c
rc_status -v
;;
stop)
  echo -n "Shutting down Freeswitch 
  /sbin/killproc $FREESWITCH_BIN
  rc_status -v
  ;;
try-restart|condrestart)
  if test "$1" = "condrestart"; then
    echo "$[attn] Use try-restart $[done](LSB)$[attn] rather than condrestart $[warn](RH)$[norm]"
  fi
  $0 status
  if test $? = 0; then
    $0 restart
  else
    rc_reset
  fi
rc_status
;;
restart)
  $0 stop
  $0 start
rc_status
;;
force-reload)
  echo -n "Reload service Freeswitch 
  /sbin/killproc -HUP $FREESWITCH_BIN
  touch /var/run/Freeswitch.pid
rc_status -v
;;
reload)
  echo -n "Reload service Freeswitch 
  /sbin/killproc -HUP $FREESWITCH_BIN
  touch /var/run/Freeswitch.pid
rc_status -v
;;
status)
  echo -n "Checking for service Freeswitch 
  /sbin/checkproc $FREESWITCH_BIN
rc_status -v
;;
probe)
  test /usr/local/freeswitch/conf/freeswitch.xml -nt /var/run/Freeswitch.pid && echo reload
  ;;
  *) echo "Usage: $0 {start|stop|status|try-restart|restart|force-reload|reload|probe}"
exit 1
esac
rc_exit

And run:

...set permissions

chmod +x /etc/init.d/freeswitch
chkconfig -a freeswitch
BusyBox Based Systems

This script was written for a CLFS embedded system, but it may be easily adapted for use on any BusyBox embedded systems by simply copy and pasting the code out of the functions.

BusyBox /etc/init.d/freeswitch

```bash
#!/bin/ash
#
# FreeSWITCH BusyBox-style startup Script
#
# For use with CLFS Embedded based systems
#
# Note: If you want to use this in any BusyBox based system, then remove
# the function definition below and also remove all traces of check_status
#
# Created by Jeffrey Leung <jleung@currie2004.ca>
#
# . /etc/rc.d/init.d/functions

PIDFILE=/usr/local/freeswitch/run/freeswitch.pid
if [ -e $PIDFILE ]
then
    PID=`cat /usr/local/freeswitch/run/freeswitch.pid`
fi

case "$1" in
    start)
        echo -n "Starting FreeSWITCH: "
        if [ -e $PIDFILE ]
            then
                false
                check_status
                exit 0
        fi
        ulimit -s 240
        /usr/local/freeswitch/bin/freeswitch -ncwait &> /dev/null
        check_status
    ;;
    stop)
        echo -n "Stopping FreeSWITCH: "
        if [ ! -e $PIDFILE ]
            then
                false
                check_status
                exit 0
        fi
        kill $PID

# A dirty hack to prevent this embedded system from messing up the filesystems
    TIMER=0
    LIMIT=300
    while [ -e $PIDFILE ]
        do
            let TIMER=TIMER+1
            sleep 1
    done
# A Timer to prevent the script from stalling if the FreeSWITCH pid did segfault
    if [ ![ $(($TIMER -gt $LIMIT)) && -e $PIDFILE ]]
        then
            kill -9 $PID
            rm -rf $PIDFILE
            check_status
            exit 0
```
fi
done
check_status
;;
reload)
echo "Reloading FreeSWITCH:"
if [ ! -e $PIDFILE ]
then
    false
    check_status
    exit 0
fi
kill -HUP $PID
check_status
;;
restart)
$0 stop
$0 start
;;
*
} echo "Usage: $0 {start|stop|reload|restart}"
exit 1
esac

BSD

 FreeBSD

/etc/rc.conf

freeswitch_enable="YES"
freeswitch_flags="-nonat -nc"
#!/bin/sh
#
# PROVIDE: freeswitch
# REQUIRE: LOGIN cleanvar
# KEYWORD: shutdown
#
# Add the following lines to /etc/rc.conf to enable freeswitch:
# freeswitch_enable: Set it to "YES" to enable freeswitch.
# Default is "NO".
# freeswitch_flags: Flags passed to freeswitch-script on startup.
# Default is "".
#
. /etc/rc.subr

name="freeswitch"
rcvar=${name}_enable

load_rc_config $name

: ${freeswitch_enable="NO"}
: ${freeswitch_pidfile="/usr/local/freeswitch/run/freeswitch.pid"}

start_cmd=${name}_start
stop_cmd=${name}_stop

pidfile=${freeswitch_pidfile}

freeswitch_start() {
    /usr/local/freeswitch/bin/./freeswitch ${freeswitch_flags}
    echo -n "Starting FreeSWITCH: "
}

freeswitch_stop() {
    /usr/local/freeswitch/bin/./freeswitch -stop
}

run_rc_command "$1"