

```
#!/bin/bash

# Run this from crontab periodically to start up
# Dire Wolf automatically.

# See User Guide for more discussion.
# For release 1.4 it is section 5.7 "Automatic Start Up After Reboot"
# but it could change in the future as more information is added.

# Versioning (this file, not direwolf version)
#-----
# v1.3 - KI6ZHD - added variable support for direwolf binary location
# v1.2 - KI6ZHD - support different versions of VNC
# v1.1 - KI6ZHD - expanded version to support running on text-only displays with
#             auto support; log placement change
# v1.0 - WB2OSZ - original version for Xwindow displays only

#How are you running Direwolf : within a GUI (Xwindows / VNC) or CLI mode
#
# AUTO mode is design to try starting direwolf with GUI support and then
# if no GUI environment is available, it reverts to CLI support with screen
#
# GUI mode is suited for users with the machine running LXDE/Gnome/KDE or VNC
# which auto-logs on (sitting at a login prompt won't work)
#
# CLI mode is suited for say a Raspberry Pi running the Jessie LITE version
# where it will run from the CLI w/o requiring Xwindows - uses screen

RUNMODE=AUTO

# Location of the direwolf binary. Depends on $PATH as shown.
# change this if you want to use some other specific location.
# e.g. DIREWOLF="/usr/local/bin/direwolf"

DIREWOLF="/usr/local/bin/direwolf"
# DIREWOLF="home/pi/direwolf/direwolf"

#Direwolf start up command :: two examples where example one is enabled
#
# 1. For normal operation as TNC, digipeater, IGate, etc.
# Print audio statistics each 100 seconds for troubleshooting.
# Change this command to however you wish to start Direwolf

DWCMD="$DIREWOLF"

#-----
#
```

```

# 2. Alternative for running with SDR receiver.
#   Piping one application into another makes it a little more complicated.
#   We need to use bash for the | to be recognized.

#DWCMD="bash -c 'rtl_fm -f 144.39M - | direwolf -c sdr.conf -r 24000 -D 1 -'"

#Where will logs go - needs to be writable by non-root users
LOGFILE=/var/tmp/dw-start.log

#-----
# Main functions of the script
#-----

#Status variables
SUCCESS=0

function CLI {
    SCREEN=`which screen`
    if [ $? -ne 0 ]; then
        echo -e "Error: screen is not installed but is required for CLI mode.
Aborting"
        exit 1
    fi

    echo "Direwolf in CLI mode start up"
    echo "Direwolf in CLI mode start up" >> $LOGFILE

    # Screen commands
    # -d m :: starts the command in detached mode
    # -S   :: name the session
    $SCREEN -d -m -S direwolf $DWCMD >> $LOGFILE
    SUCCESS=1

    $SCREEN -list direwolf
    $SCREEN -list direwolf >> $LOGFILE

    echo "-----"
    echo "-----" >> $LOGFILE
}

function GUI {
    # In this case
    # In my case, the Raspberry Pi is not connected to a monitor.
    # I access it remotely using VNC as described here:
    #
http://learn.adafruit.com/adafruit-raspberry-pi-lesson-7-remote-control-with-vnc
    #
    # If VNC server is running, use its display number.

```

```

# Otherwise default to :0 (the Xwindows on the HDMI display)
#
export DISPLAY=":0"

#Reviewing for RealVNC sessions (stock in Raspbian Pixel)
if [ -n "`ps -ef | grep vncserver-x11-serviced | grep -v grep`" ]; then
    sleep 0.1
    echo -e "\nRealVNC found - defaults to connecting to the :0 root window"
elif [ -n "`ps -ef | grep Xtightvnc | grep -v grep`" ]; then
    #Reviewing for TightVNC sessions
    echo -e "\nTightVNC found - defaults to connecting to the :1 root window"
    v=`ps -ef | grep Xtightvnc | grep -v grep`
    d=`echo "$v" | sed 's/.*tightvnc *\(:[0-9]\).*\/1/'`
    export DISPLAY="$d"
fi

echo "Direwolf in GUI mode start up"
echo "Direwolf in GUI mode start up" >> $LOGFILE
echo "DISPLAY=$DISPLAY"
echo "DISPLAY=$DISPLAY" >> $LOGFILE

#
# Auto adjust the startup for your particular environment:  gnome-terminal,
xterm, etc.
#

if [ -x /usr/bin/lxterminal ]; then
    /usr/bin/lxterminal -t "Dire Wolf" -e "$DWCMD" &
    SUCCESS=1
elif [ -x /usr/bin/xterm ]; then
    /usr/bin/xterm -bg white -fg black -e "$DWCMD" &
    SUCCESS=1
elif [ -x /usr/bin/x-terminal-emulator ]; then
    /usr/bin/x-terminal-emulator -e "$DWCMD" &
    SUCCESS=1
else
    echo "Did not find an X terminal emulator.  Reverting to CLI mode"
    SUCCESS=0
fi
echo "-----"
echo "-----" >> $LOGFILE
}

# -----
# Main Script start
# -----

# When running from cron, we have a very minimal environment
# including PATH=/usr/bin:/bin.
#

```

```
export PATH=/usr/local/bin:$PATH

#Log the start of the script run and re-run
date >> $LOGFILE

# First wait a little while in case we just rebooted
# and the desktop hasn't started up yet.
#
sleep 30

#
# Nothing to do if Direwolf is already running.
#

a=`ps ax | grep direwolf | grep -vi -e bash -e screen -e grep | awk '{print $1}'`
if [ -n "$a" ]
then
    #date >> /tmp/dw-start.log
    #echo "Direwolf already running." >> $LOGFILE
    exit
fi

# Main execution of the script

if [ $RUNMODE == "AUTO" ];then
    GUI
    if [ $SUCCESS -eq 0 ]; then
        CLI
    fi
    elif [ $RUNMODE == "GUI" ];then
        GUI
    elif [ $RUNMODE == "CLI" ];then
        CLI
    else
        echo -e "ERROR: illegal run mode given. Giving up"
        exit 1
    fi
```