

# How To Set Up NewsLine On Multiple Allstar Nodes

There are several HOWTOs out there that explain how to set up your Allstar node to broadcast Newsline. Those methods work well if you only have one node to deal with. But what if you have more than one?

My system currently consists of 3 repeaters, all linked via Allstar to a hub at my house. The reason for this is that I have Internet connectivity at my house but not my repeater sites (I use the AREDN Mesh Digital Network to tie it all together). I want to be able to tie my IRLP node into my system, as well as a SAME weather receiver.

This document will explain how to automate the process of broadcasting Amateur Radio NewsLine through your multi-node Allstar system. On my system, the latest news.mp3 from <https://www.arnewsline.org/> is automatically downloaded on Fridays afternoon, giving me lots of time to manually insert my IDs before broadcasting NewsLine on Sunday and Monday evenings.

In addition, all day Sunday and Monday, my system sends a tail message out to all repeaters informing about the 6 pm broadcast time. So the process works like this:

- Every Friday at 4:10 pm local time, a cronjob downloads the NewsLine news.mp3 file and moves it to its own directory (/etc/asterisk/audio/newsfiles/). Downloading it on Friday allows for plenty of time to edit it to insert my own IDs into it.
- Starting on Sunday, 0000 hours local time, another cronjob starts the Tail Message script to announce the broadcast at 1800 hours and stop the script 10 minutes before start time. This tail message runs 2 minutes after the last activity on any repeater in the system but cannot run more frequently than every 15 minutes. It sends commands to each of the other nodes (via passwordless ssh login) to play the appropriate tail message directly on that node. Self-terminate at 1749 hours local time
- At 1750 local time, another cronjob starts the script that send commands to each node to announce the 10 and 5 minute warning at 1750 and 1755 hours local time respectively and then broadcasts NewsLine.
- At 1752 hours local time, a cronjob runs the script that converts the news.mp3 file to news.ul, the only audio format Allstar understands and send a copy to each of the nodes in the system. Delete the now unneeded news.mp3 and news.wav files (the .wav version is created as part of the conversion from .mp3 to .ul). Running it
- Right before NewsLine plays, send the appropriate command to instruct each of the remote Allstar nodes to send DTMF to their connected RC210 Repeater Controller, which in turn runs a Macro to disable the repeater's receiver, Timeout Timer and IDs for 30 minutes (as the NewsLine audio file contains the necessary IDs, there is no need for the controller to do so). Disabling the Timeout Timer ensures the transmitter stays active for the entire broadcast.
- After NewsLine completes, send the appropriate command to instruct each of the remote Allstar nodes to send DTMF to their connected RC210 Repeater Controller, which in turn runs a Macro to re-enable the repeater's receiver, Timeout Timer IDs

As I broadcast Newsline on both Sunday and Monday evenings, I have crontabs set to duplicate everything except the download and file convert/copy on Mondays.

## How To Automatically Connect To Your Remote Allstar Nodes

You probably know how to use ssh to connect to your various Allstar nodes and also realize that (depending on which ssh client you use), you need to enter your password each time you log in.

Fortunately for us, Linux provides an easy way to allow us to ssh to a remote node without requiring a password be entered every time we do so, FROM A PARTICULAR COMPUTER. This is very secure and allows for unattended operation, such as from a cronjob.

The first thing we need to do is to generate a public key from the Allstar server/node that will serve as our “master” or “hub” so we can send commands to the remote Allstar nodes without needing a password.

Log onto your Allstar node and drop to the bash shell. Change your directory to /root if not already there.

From the command line, type `ssh-keygen` and hit ENTER

From the output, we will see that it created a private-public key pair saved in `/root/.ssh/id_rsa` and `/root/.ssh/id_rsa.pub` respectively.

In order to connect to a remote host with your private key, you need to copy the public key to it. This needs to be done only once. As we need to connect as user root, use the following command:

```
ssh-copy-id root@destination_address
```

“destination\_address” is the IP (or domainname) of the remote node you want to connect to

You can test that this worked by sending a command to the remote node. For example, from the command line:

```
ssh root@destination_address ls /etc
```

If all is working, you should receive a directory listing back from the remote node of its /etc directory.

You need to copy your public key to each remote node you wish to connect to without the need of providing a password. Remember however that this will only work from the current node you’re working with.

In addition to playing the NewsLine news file, my scripts also play various advisories sound files

<http://www.ah6le.net/audiofiles/newstonight.ul>

[http://www.ah6le.net/audiofiles/newsline\\_5.ul](http://www.ah6le.net/audiofiles/newsline_5.ul)

[http://www.ah6le.net/audiofiles/newsline\\_10t.ul](http://www.ah6le.net/audiofiles/newsline_10t.ul)

You can click on the above files and download them for your own use. I had them professionally recorded but you may use them for your own personal non-commercial Amateur Radio use. Move them to `/etc/asterisk/audio/newsfiles/` on EACH OF YOUR NODES.

## Creating ramdisk on your Allstar node

In order for one of the new scripts used to be able to know when your node is actively receiving or transmitting, we're going to add some items to its rpt.conf file that checks for such activity and create a file so applications other than Allstar know about it. As I'm running my server/node on a Raspberry Pi, I didn't want to do continuous writes to the SD card so I created ramdisk on the server.

From the command line (as root), enter:

```
mkdir /etc/ramdisk
```

Now edit /etc/fstab to create ramdisk on startup of Linux and add the following line

```
tmpfs /etc/ramdisk tmpfs nodev,nosuid,size=1M 0 0
```

Reboot your server and you'll see the new ramdisk

```
df -h |grep ramdisk
tmpfs      1.0M    0 1.0M    0% /etc/ramdisk
```

## **Editing the /etc/asterisk/rpt.conf file on your local ("hub") node**

There are some additions that are needed to your Allstar node's rpt.conf file in order to generate the file that is used by the scripts to determine activity.

Under the node stanza, add the following line;

```
events = events1234
```

At the end of the file, add the following lines:

```
[events1234]
;the following 2 lines create a file and delete it for tailmsg in response to either Allstar RX or TX activity/inactivity
touch /etc/ramdisk/cor = s|t|RPT_RXKEYED
touch /etc/ramdisk/cor = s|t|RPT_TXKEYED
rm /etc/ramdisk/cor = s|f|RPT_RXKEYED
rm /etc/ramdisk/cor = s|f|RPT_TXKEYED
```

Save the changes and restart your Allstar node

## Editing the /etc/asterisk/rpt.conf file on your remote nodes

In order to allow the remote nodes to send pre-configured DTMF to the RC210 (in order to run the Macros to disable Timeout, etc and later re-enable it), we need to define DTMF digits to send and the code to send them.

Under the [functions] stanza:

```
;Send DTMF commands to disable repeater receiver and run macro to stop timeout, ID, etc.  
880 = cop,48,A,A,C,D
```

```
;Send DTMF commands to re-enable repeater receiver and run macro to stop timeout, ID, etc.  
881 = cop,48,A,A,B,D
```

As you see above, if we send the DTMF command \*880 to the remote node, it will send DTMF AACD to the controller. Likewise, if we send the DTMF command \*881 to the node, it will send DTMF AABD to the controller. You may want to edit as necessary for your particular setup. Save the file and restart the node

## SCRIPTS

The first script we need is the one to download the NewsLine mp3 news file. If not already present, create a directory /etc/asterisk/custom then change to it (cd /etc/asterisk/custom), create a file called news\_download and place the following in it (between the -----lines).

```
-----  
#!/bin/bash  
  
#-----  
# Set the variables for this script  
NEWSDIR=/etc/asterisk/audio/newsfiles  
  
# Clear the bad_download flag if it was previously set  
if [ -f $NEWSDIR/bad_download ] ; then  
    rm -f $NEWSDIR/bad_download  
fi  
if [ -f $NEWSDIR/wav_format ] ; then  
    rm -f $NEWSDIR/wav_format  
fi  
  
FILE="news"  
  
# Delete the files from the last run  
rm -f $NEWSDIR/$FILE.mp3  
rm -f $NEWSDIR/error.log  
  
# Lets get the latest news  
echo "Downloading "$FILE".mp3 . . . "  
  
/usr/bin/wget -t 300 https://www.arnewsline.org/s/$FILE.mp3  
/bin/mv /root/$FILE.mp3 $NEWSDIR  
  
# Check that file is present. If not, set the bad_download flag.  
if [ ! -f $NEWSDIR/$FILE.mp3 ] ; then  
    echo "Cannot find $FILE.mp3" >> $NEWSDIR/error.log  
    touch $NEWSDIR/bad_download  
fi  
done  
-----
```

Save the file and, from the command line, enter

```
chmod u+x news_download
```

Note that Allstar (asterisk) only knows how to deal with ul type sound files. As such, we need to convert the downloaded news.mp3 file to news.ul. We need to do this in 2 steps and then send the converted news file to each node. The following script does this automatically for us

Create the file /etc/asterisk/custom/copy\_news and place the following in it (between the ----- lines):

```
-----
#!/bin/bash

#convert mp3 to wav and finally to ul format for Allstar (must be done in 2 steps)
/usr/bin/lame --decode /etc/asterisk/audio/newsfiles/news.mp3 /etc/asterisk/audio/newsfiles/news.wav
/usr/bin/sox /etc/asterisk/audio/newsfiles/news.wav -r 8000 -c 1 -t ul /etc/asterisk/audio/newsfiles/news.ul

#send news file to each node
/usr/bin/scp /etc/asterisk/audio/newsfiles/news.ul root@10.48.13.20:/etc/asterisk/audio/newsfiles/news.ul
/usr/bin/scp /etc/asterisk/audio/newsfiles/news.ul root@10.150.180.36:/etc/asterisk/audio/newsfiles/news.ul
/usr/bin/scp /etc/asterisk/audio/newsfiles/news.ul root@10.79.158.106:/etc/asterisk/audio/newsfiles/news.ul

#remove both wav and mp3 as neither is needed any longer
rm /etc/asterisk/audio/newsfiles/news.mp3
rm /etc/asterisk/audio/newsfiles/news.wav
-----
```

**NOTE: You MUST CHANGE the addresses in each "root@" lines and node numbers to reflect your particular system. I HAVE 3 REMOTE NODES. YOU MAY NEED MORE OR LESS IN YOUR SYSTEM**

Save the file and, from the command line, enter

```
chmod u+x copy_news
```

## news\_tx

This is the script that commands all nodes to announce the 10 and 5 minute warnings, send DMTF to the controller (instructing it to run the Macros to disable timeouts, repeater receiver and IDs), play the NewsLine audio file and re-enable timeouts, receiver and IDs after the broadcast has concluded.

Create the file /etc/asterisk/custom/news\_tx and place the following in it (between the ----- lines):

```
-----
#!/bin/bash
# Set the time intervals for the warning announcements
COUNTDOWN=600
WARNING_2=300

WARNING_1_FILE=/etc/asterisk/audio/newsfiles/newslines_10
WARNING_2_FILE=/etc/asterisk/audio/newsfiles/newslines_5

echo "Starting countdown"
timer=$COUNTDOWN
X=0
PRE=TRUE

while [ "$PRE" = "TRUE" ] ; do
  if [ $timer = $COUNTDOWN ] ; then
    /usr/sbin/asterisk -x "rpt playback 1100 $WARNING_1_FILE"
    ssh root@10.48.13.20 "/usr/sbin/asterisk -x 'rpt playback 1101 $WARNING_1_FILE'"
    ssh root@10.79.158.106 "/usr/sbin/asterisk -x 'rpt playback 1102 $WARNING_1_FILE'"
    ssh root@10.150.180.36 "/usr/sbin/asterisk -x 'rpt playback 1103 $WARNING_1_FILE'"
    X=1
  fi
done
-----
```

```

if [ $X = 1 ] ; then
    let timer=$timer-1
fi

echo $time

# If the countdown has reached the value of WARNING_2, play the warning.
if [ $timer = $WARNING_2 ] ; then
    /usr/sbin/asterisk -rx "rpt playback 1100 $WARNING_2_FILE"
    ssh root@10.48.13.20 "/usr/sbin/asterisk -rx 'rpt playback 1101 $WARNING_2_FILE'"
    ssh root@10.79.158.106 "/usr/sbin/asterisk -rx 'rpt playback 1102 $WARNING_2_FILE'"
    ssh root@10.150.180.36 "/usr/sbin/asterisk -rx 'rpt playback 1103 $WARNING_2_FILE'"
fi

# Check if the timer has reached 0 yet..
if [ $timer = 0 ] ; then
    echo "Time's up!"
    #disable repeater receiver and run macro to stop timeout, ID, etc.
    /usr/sbin/asterisk -rx "rpt fun 1100 *880"
    ssh root@10.48.13.20 "/usr/sbin/asterisk -rx 'rpt fun 1101 *880'"
    ssh root@10.79.158.106 "/usr/sbin/asterisk -rx 'rpt fun 1102 *880'"
    ssh root@10.150.180.36 "/usr/sbin/asterisk -rx 'rpt fun 1103 *880'"

    sleep 5
    #start playback on each node
    /usr/sbin/asterisk -rx "rpt playback 1100 /etc/asterisk/audio/newsfiles/news"
    ssh root@10.48.13.20 "/usr/sbin/asterisk -rx 'rpt playback 1101 /etc/asterisk/audio/newsfiles/news'"
    ssh root@10.79.158.106 "/usr/sbin/asterisk -rx 'rpt playback 1102 /etc/asterisk/audio/newsfiles/news'"
    ssh root@10.150.180.36 "/usr/sbin/asterisk -rx 'rpt playback 1103 /etc/asterisk/audio/newsfiles/news'"
    PRE=FALSE
fi

sleep 1

done

echo "leaving"

#Renale the receiver that we turned off at the start
/usr/sbin/asterisk -rx "rpt fun 1100 *881"
ssh root@10.48.13.20 "/usr/sbin/asterisk -rx 'rpt fun 1101 *881'"
ssh root@10.79.158.106 "/usr/sbin/asterisk -rx 'rpt fun 1102 *881'"
ssh root@10.150.180.36 "/usr/sbin/asterisk -rx 'rpt fun 1103 *881'"

exit 0

```

**NOTE: As before, You MUST CHANGE the addresses in each "root@" lines and change node numbers to reflect your particular system. I HAVE 3 REMOTE NODES. YOU MAY NEED MORE OR LESS IN YOUR SYSTEM**

Save the file and, from the command line, enter

```
chmod u+x news_tx
```

# tailmsg

This is the script to send the announcement 2 minutes after the last use of any repeater in the system. Create the file /etc/asterisk/custom/tailmsg and place the following in it (between the ----- lines). Note that this script is written in Perl as it runs continuously in the background and Perl does this "better" than a bash or PHP script.

```
-----!usr/bin/perl

use strict; use warnings;
use Time::ParseDate;

(my $Second, my $Minute, my $Hour, my $DayOfMonth, my $Month, my $Year, my $RealMonth);

my $keyed=0;
my $active=0;
my $lastmsg=0;
my $inactive_halfseconds=1;
my $showoften = 120;
my $msgtimer=0;
my $filename = '/etc/ramdisk/cor';

while(1){
    ($Second, $Minute, $Hour, $DayOfMonth, $Month, $Year) = localtime(time);
    print "hour=$Hour, Minute=$Minute\n";
    if ($Hour == 17 && $Minute > 49) || $Hour > 17 ){
        print "Quitting\n";
        exit;
    }
    select(undef,undef,undef,.5);

#Check to see if Allstar has been active
    if (-e $filename) {
        print ("Success\n");
        $active=1;
    }

    if($active) {
        print "ACTIVE\n";
        $inactive_halfseconds=1;
    } else {
        $inactive_halfseconds++;
    }

# When inactive, send the message every $showoften minutes
    if(($inactive_halfseconds / 120) > $showoften) {
        send_tailmessage();
        print "TAIL MESSAGE!!! $lastmsg \n" ;
        $inactive_halfseconds = 0;
    }

# When active, send the message 2 minutes after last activity.
    $msgtimer=0;
    while($active==1) {
        sleep 1;
        $msgtimer++;
        print "Timer $msgtimer \n" ;
        if($msgtimer > 120){
            $active=0;
            send_tailmessage();
            print "TAIL MESSAGE!!! $lastmsg \n" ;
        }
    }
}
}
```

```

sub send_tailmessage {
# abort if last message was less than 300 seconds (5 minutes) ago
my $now = time();
print "now=$now, lastmsg=$lastmsg\n";
if( ( time() - 900) < $lastmsg) && $lastmsg > 0) {
    print "tail message aborted, too soon \n";
    return;
}
sleep 5;

select(undef,undef,undef,.5);
system("/usr/sbin/asterisk -rx 'rpt playback 1100 /etc/asterisk/audio/newsfiles/newstonight'");
my @args = ( '-l', 'root',
    '10.48.13.20',
    "/usr/sbin/asterisk -rx 'rpt playback 1101 /etc/asterisk/audio/newsfiles/newstonight'"
);
system('ssh',@args);

@args = ( '-l', 'root',
    '10.79.158.106',
    "/usr/sbin/asterisk -rx 'rpt playback 1102 /etc/asterisk/audio/newsfiles/newstonight'"
);
system('ssh',@args);

@args = ( '-l', 'root',
    '10.150.180.36',
    "/usr/sbin/asterisk -rx 'rpt playback 1103 /etc/asterisk/audio/newsfiles/newstonight'"
);
system('ssh',@args);

select(undef,undef,undef,.5);
$lastmsg=time();
}

```

---

**NOTE: You MUST CHANGE the addresses and node numbers to reflect your particular system. I HAVE 3 REMOTE NODES. YOU MAY NEED MORE OR LESS IN YOUR SYSTEM**

You should already have Perl installed on your system. You can check by typing (at the command line)

```
perl -v
```

Next, check that cpan is installed as we need to use the perl module Time::ParseDate.pm

You can check by typing (at the command line)

```
cpan
```

Once in, use the following command to install the needed module

```
Install Time::ParseDate.pm
```

Once done, chmod the script as you did the others

```
chmod u+x tailmsg
```



And finally, we need to program up some crontabs so we can automate the process

At the command prompt, type and add the following lines:

```
crontab -e
```

```
10 16 * * 5 (/etc/asterisk /custom/news_download newsline > /dev/null 2>&1)
```

```
52 17 * * 0 (/etc/asterisk/custom/copy_news > /dev/null 2>&1)
```

```
49 17 * * 0 (/etc/asterisk /custom/news_tx newsline > /dev/null 2>&1)
```

```
49 17 * * 1 (/etc/asterisk /custom/news_tx newsline > /dev/null 2>&1)
```

```
0 7 * * 0,1 /etc/asterisk /custom/taimsg&
```

Save the crontab

The first line runs on Friday at 1610 hours local time to download the news.mp3 from the ARNewsLine website

The second line runs on Sunday at 1752 hours local time to convert the downloaded news.mp3 file to news.ul and then sends the converted file to each of our nodes

The third and fourth lines run on Sunday and Monday respectively at 1749 hours local time to give the 10 and 5 minute warnings and plays the actual NewLine broadcast.

Of course, you may wish to modify the days and/or times to fit your needs. Just remember that you need to run copy\_news before your NewsLine broadcast starts and be sure to give it at least a few minutes to copy to each of your nodes.

And there you have it. You can automate your NewsLine broadcast on your multiple node Allstar system easily and without too much fuss

73,

Ken, AH6LE