

## INSTALLATION NOTES FOR AUTOSKYWARN Version 6 – Version 180207

AutoSkyWarn is a program developed to run on the AllStar DIAL distribution allowing an AllStar Server and Node to automatically report and announce about fifty National Weather Service (NWS) Warnings, Watches, Advisories and Statements (WWAS). The program was created and developed by Steve Mahler - KF5VH in late 2016 and 2017. This project started as a personal, for fun, project and has grown into what you see today. Steve says “GO BOILERS”!!

AutoSkyWarn (ASW) can use the AllStar “tail messages” and the Linux crontab. If you are not familiar with these two concepts you might want to do a little review before installing and configuring AutoSkyWarn. Documentation for AutoSkyWarn and MID-JB-PRI can be found by going to <http://KF5VH.com> and selecting the link to ASW or MID.

### SPECIAL NOTICE:

Starting with ASW 6K, ASW has code to integrate the MID-JB-PRI software. The integration allows ASW to tell MID-JB-PRI to switch to a correct repeater ID before keying up the transmitter. See also MID INTEGRATION near the end of this document.

### CHECK FOR ASSOCIATED SOFTWARE

ASW as distributed uses SOX and FESTIVAL. Start by checking if these items exist. Use the “which” command. If you get a pathname in return the package is already loaded. No path name? You need to load the software (see below).

```
# which sox
```

```
/usr/bin/sox
```

```
# which festival
```

```
/usr/bin/festival
```

ASW calls “catwav” which calls SOX (Sound Exchange). SOX must be installed on the platform to run the ASW code. Just load it from the Debian Application Site.

I also have the 12 control commands using "Talkit" to voice confirmation that the command executed. Talkit calls "festival". Load the free package if you would like to use Talkit. [ <https://www.howtoinstall.co/en/debian/jessie/festival> has installation instructions] If not, just comment out the line in each command or change the way the command confirms operation.

## THE ACTUAL INSTALLATION

The default directory for AutoSkyWarn (ASW) is

```
/usr/local/AUTOSKY
```

So first move the tar file (autoskywarn6Xdist.tar) to “/usr/local”. If this is a re-install you can `mv AUTOSKY AUTOSKY-OLD` and save the current working version (just in case) or you can delete /usr/local/AUTOSKY and its contents. In either case it is time to expand the tar file.

With your current directory (use `pwd` to check) /usr/local enter the command

```
# pwd
```

```
/usr/local
```

```
# tar -xvf autoskywarn6Xdist.tar
```

This will scroll a list of files and install them in /usr/local/AUTOSKY for you (takes only seconds). You might want to move or delete the tar file after installation.

The next installation step is to review "autoskywarn.conf" in the directory /usr/local/AUTOSKY. Read the whole thing and sections marked for KF5VH should be changed to the general user setting. Make the edits as necessary:

- BASEDIR is where the ASW package lives.
- NODE is the AllStar node number where the package will run.
- WXT is where ASW will build the weather tail message. (most users will use /usr/local/AUTOSKY)
- LIMTOT is the number of events before BobTail thresholding.
- AUDIOTIC points to the attention getting sound file for each event.
- OFILE is the string that pulls the weather data for your county/parish.

- And other variables as necessary ... the variables listed above should be enough to get started.

To find your OFILE ...

- Go to <https://alerts.weather.gov>
- Scroll down and find your state
- Click on the words "County List"
- Find your county/parish
- Click on the ATOM icon in line with your county
- Then look at the browser address bar.
- The below OFILE is for Lafayette Parish Louisiana. Think of it as
- LA - Louisiana / C - "County" / 055 - County number 55.
- OFILE='wwaatmget.php?x=LAC055&y=0'

## A QUICK TEST

[The audio results of these instructions are IFF there are no Audio Tags. If Audio Tags exist then that audio will be added to the information presented below.]

```
# cd /usr/local/AUTOSKY (or wherever BASEDIR is located)
```

```
# ./AutoSky1 1
```

You should hear: Updated Weather Information, tic Tornado Warning tic Tornado Watch. [These test messages might be preceded by "for Lafayette Parish"]

```
# ./AutoSky1 2
```

You should hear: Updated Weather Information tic Red Flag Warning tic Tornado Watch with Multiples tic and Other Alerts.

```
# ./AutoSky1 (if the weather is All Clear and you haven't changed A1FILE)
```

You should hear: The National Weather Service has no Alerts, Watches or Warnings for Lafayette Parish.

If the above test sequence works as described then the basics of ASW are working.

The "ALL CLEAR" message is in .../SOUNDS/asn96Z.wav

This wav file and all ASW WAV files are 8K. The lead sound effect that is designed to be unique and grab attention. There are now several 96X or 96XX series messages. A suggested starting point is 96WY and then use the Audio Tagging system to patch in your county name. Pick the one you like and in autoskywarn.conf set variable A1FILE to the file name (no extension). Using your favorite sound editor you can add, change or remove these sounds. The Audio Tagging system allows you to add a sound to the beginning or end of the All Clear message (or the list of weather events).

## MAKE IT AUTOMATIC

You can use `# crontab -l` ; to get a peek at what a crontab looks like and “man crontab” if you need more information. Next use

```
# crontab -e
```

and add the line making ASW work (write and quit the editor to install):

```
* /4 * * * * /usr/local/AUTOSKY/AutoSky1
```

The above crontab line has AutoSky obtaining NWS data every four (4) minutes.

To test this addition:

```
# cd /usr/local/AUTOSKY
```

```
# ./AutoSky1 1
```

And you will hear the Tornado Warning and Watch message. Within 4 minutes the system should play the current conditions (probably the all clear message).

To be sure which functions are ON or OFF you can

```
# cd /usr/local/AUTOSKY
```

```
# ls *enbl* ← This provides a list of ENABLED functions.
```

The returned information for everything on looks like:

```
alert.enbl.bobtail alert.enbl.inform alert.enbl.master alert.enbl.rptctl alert.enbl.saytail
```

## CONTROL COMMANDS

AutoSky.Master.[ON|OFF] - enable/disable all ASW activity

AutoSky.BobTail.[ON|OFF] - enable/disable truncating event list, and if truncated say "and MORE"

AutoSky.Informe.[ON|OFF] - enable/disable immediate announcement of a change of event conditions to ALL CLEAR

AutoSky.Informf.[ON|OFF] - enable/disable immediate announcement of a change of event conditions to UPDATED WEATHER INFORMATION

AutoSky.SayTail.[ON|OFF] - enable/disable immediate announcement of the new WX-TAIL message

AutoSky.RptCtl.[ON|OFF] – enable/disable calling of repeater control scripts

## AUDIO TAGS

A new feature in ASW Version 6A is Audio Tags. Users who are interested in having ASW announce weather events in multiple counties wanted a way to have the repeater announce the name of the area associated with the weather events. For example, “for Saint Joseph County”. The implementation allows each of the four tag points to have different audio.

You make a tag by generating one or more 8K WAV file(s). The ASW program looks in directory /usr/local/AUTOSKY/SOUNDS for four file names.

1. “countyfe.wav” – the audio tag will be placed “following empty” message
2. “countyle.wav” – the audio tag will be placed “leading empty” message
3. “countyft.wav” – the audio tag will be placed “following wx-tail” message
4. “countylt.wav” – the audio tail will be placed “leading wx-tail” message

And that is it, nothing to turn on (just create the filename) and nothing to turn off (just remove the file name). Yes, you can control the existence of and the audio content from any shell script. That would include “RptCtl[01]” changing the Audio Tagging files. RptCtl1 is called with parameters: # of warnings, # of watches, # of advisories, # of special statements, if BobTailing truncated the wx-tail list (0=No 1=Yes) and the number of events actually processed to build the weather tail.

## REPEATER CONTROL

If Repeater Control is enabled, each time the system plays the All Clear message **or** builds a new weather tail message based on NWS alerts you have the option of running a shell script. The script has the ability to do many things. ASWs author uses it to change the Repeater ID to something that indicates SkyWarn Mode is set on the repeater. The primary scripts are called with additional information parameters so you can be selective in how the repeater acts. More information is in the README1st document.

## MID INTEGRATION

Each time AutoSkyWarn switches from CLEAR to EVENT mode, and each time NWS changes the alerts, ASW attempts (using MID) to set the Repeater ID to a SkyWarn style ID before keying the transmitter. By default the integration code uses “jb” requesting playlist “play2” at priority 2.

Each time AutoSkyWarn switches from EVENT to CLEAR mode ASW attempts (using MID) to set the Repeater ID to a non-SkyWarn ID before keying the transmitter. By default the integration code calls PRI to set the MID priority level to level 9 (using “jbunlock”) and then calls “jb” requesting playlist “play8” at priority 8.

This is using my system of playlist and priority levels. If you insist on using something different you must change the script. File AutoSky3 has the code for going from EVENT to CLEAR. File AutoSky4 has the code for going from CLEAR to EVENT or EVENT to NEW EVENT. To find the code in either file search for “MID-JB-PRI”. If you do not want to use MID to enhance your repeater IDs, then mv or rm directory /usr/local/MID so the software will not run the integration code.

## BLOCKING

Refer to the README1st document and the BLOCKING document.

## OTHER INSTALLATION CONSIDERATIONS

The WX-TAIL Message can be linked to the AllStar Tail Message system (this is the way KF5VH uses the weather tail) via "tailmessagelist" in rpt.conf (multiple cycling files are allowed). Remember, if the WX-TAIL message is played by the tail message system then a local keyup will abort the reading of events and it will be rescheduled. When the list of events is played via SayTail a keyup will be ignored and the message will continue to play.

My /etc/asterisk/rpt.conf file looks like ...

```
; TAIL MESSAGE CONTROL
; The tailmessagetime,tailsquashedtime, and tailmessages need to be set
; to support tail messages. They can be omitted otherwise.
tailmessagetime=180000 ; Play a tail message every 3 mins (180000)
tailsquashedtime=60000 ; If squashed by a user TX, try again after 60 seconds (60000)
; TAILMESSAGELIST HAS LIST OF 2 ROTATING MESSAGES PLAYED FOR TAIL MESSAGE
; WX&MAINT IN THE FIRST FILE AND NETS&MEETINGS IN THE SECOND FILE
tailmessagelist=/usr/local/AUTOSKY/wx-tail,/etc/asterisk/SCRIPTS/WXA/oo-tail
```

This allows the ASW wx-tail file to be played, and the next time a tail message plays it will be oo-tail. File oo-tail is a message that contains net and meeting reminders. If you only use wx-tail, I would suggest that "tailmessagetime" be increased to 5-7 minutes.

File rpt.conf can also be programmed to use DTMF codes (examples 96 & 97 below) to run the Real-time Controls. You could also have a command to immediately run AutoSky1 and not wait for crontab (ex. 99). You can build a command to play the WX-TAIL Message (/usr/local/AUTOSKY/wx-tail.wav) on demand (ex. 98).

96=cmd,/usr/local/AUTOSKY/AutoSky.BobTail.OFF

97=cmd,/usr/local/AUTOSKY/AutoSky.BobTail.ON

98=localplay,/usr/local/AUTOSKY/wx-tail

99=cmd,/usr/local/AUTOSKY/AutoSky1

Not required for ASW but extremely useful is the "rpt" shell script. You can move the script to a directory in your normal PATH like "/usr/local/bin". From the Linux

prompt it allows you to execute repeater DTMF commands directly. Use of "\*" is optional, for example:

```
# rpt 71    or    # rpt *32500
```

#### LISTEN TO RECORDED MESSAGES

Shell Script "playasn ##" causes file /usr/local/AUTOSKY/SOUNDS/asn##.wav to be played.

I hope you find AutoSkyWarn an aid to your SkyWarn operations.

...STeve – KF5VH