

README1 for AUTOSKYWARN Version 6 – Version 170919

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 early 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.

The default directory for AutoSkyWarn is

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

So first move the tar file (autoskywarn6X.dist) 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 remove the AUTOSKY directory 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

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

```
/usr/local ←if you don't see this you need to know why
```

```
# tar -xvf autoskywarn6X.dist
```

This will scroll a list of files and install them in `/usr/local/AUTOSKY`, you might want to move or delete the tar file at this time.

The next installation step is to review "autoskywarn.conf". 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.
- 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.

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'

OVERALL FLOW

A) AutoSky1 – started via crontab (suggested every 4 minutes), if MASTER is ENABLED then pull data from NWS. But if MASTER is DISABLED then immediately and quietly exit. Next decide if data is same as previous NWS data, if it is the same then do nothing and exit, or if the data has changed start AutoSky2.

B) AutoSky2 - begin processing data because it is different than the previous version received from the NWS. Part of pre-processing is the BLOCKING function (new in version 6C). Please refer to the Blocking document for details. After pre-processing determine if the data says "all clear". If the data says there are no WWAS call AutoSky3, if there is new data call AutoSky4.

C) AutoSky3 - All events are clear. Clear the tail message (N.B., actually you don't "clear" the message, you make the contents a very short silent sound). If INFORME is enabled play a non-interruptible all clear message. Make the CURRENT NWS DATA the PREVIOUS NWS DATA. See AUDIO TAGGING for information about pre and post-pending audio to the All Clear message. If RPTCRL is enabled call shell script "RptCtl0".

D) AutoSky4 - Events have changed. Rebuild the WX-TAIL message. If BOBTAIL is enabled execute code to limit length of tail message. If INFORMF is enabled play a non-interruptible message that conditions have changed. If SAYTAIL is enabled play a non-interruptible WX-TAIL message now (the list the NWS events (subject to BobTailing)). Make the CURRENT NWS DATA the PREVIOUS NWS DATA. See AUDIO TAGGING for information about pre and post-pending audio to the Weather Tail Message. If RPTCRL is enabled call shell script "RptCtl1".

E) Real-time control of 12 command options ...

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

F) The WX-TAIL Message can be linked to the AllStar Tail Message system 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 reschedule. When the list of events is played via SayTail a keyup will be ignored and the message will continue to play.

G) File rpt.conf can also be programmed to use DTMF codes (examples below) to run the Real-time Controls (E above). You could also have a command to immediately run AutoSky1 and not wait for crontab. You can build a command to play the WX-TAIL Message (/usr/local/AUTOSKY/wx-tail.wav) on demand. See the Install-Notes for more information.

TESTING FACILITIES

I) In file AutoSky1 there is a method for testing so you can install and test on a sunny day. Run AutoSky1 manually with a single parameter, range 1-8. Change directory to /usr/local/AUTOSKY then: “ ./AutoSky1 2” will overwrite the NWS data with test file data (test.file2). In crontab for normal operations you call AutoSky1 with no parameters.

II) Remember that if you run a test of AutoSky1 the system should talk to you. If you run the same test again, it won't speak because the data has not changed. You can run a different test “./AutoSky1 4” and the system should speak again. Turn off your crontab entry (just comment it out, but remember to put it back in for production) ... it can be confusing testing with Crontab starts the software.

III) AutoSky[1-4] have a "set -x" debug command commented out (you see the trace/debug only if you run the program from the command line). If you want to get in the weeds it is a good way to see what is going on. Want to go deeper? Change the command to “set -xv”.

IV) ASW is designed to run from any starting point during testing. If AutoSky 1 and 2 are properly calling AutoSky4 that fails, you can just run ./AutoSky4 to have it reprocess the data.

BOBTAILING APPROACH

ASWs first large weather event generated user comments about the length of the tail message. The ASW solution is BOBTAILING. The rules are:

- * If one [WARNING|WATCH|ADVISORY|STATEMENT] is voiced, then all [WARNINGS|WATCHES|ADVISORIES|STATEMENTS] are voiced.

- * The events are presented in the following order: WARNINGS, WATCHES, ADVISORIES and STATEMENTS

- * ALL WARNINGS are always included. At the beginning of each following group, if the existing number of entries is greater than LIMTOT then stop copying groups

and add an "and more events" message. LIMITOT can be found in file autoskywarn.conf

REMOVING DUPLICATE EVENTS

In a single county/parish you can have the same alert more than once. For example, Flood Warnings for two distinct areas in the same county. In ASW Version 4 the software says "Flood Warning" twice. AutoSkyWarn Version 5 takes any "multiples" and says the event ("Flood Warning") only once and deletes all other multiples ... and it adds the words "with Multiples".

CHANGING THE BASE DIRECTORY

In file autoskywarn.conf you can change the BASEDIR (where the program lives, or where you wish to put a test, backup or multiple production copy (more later)). A side effect of moving the BASEDIR is that the shell scripts (A*[1234] A*OFF A*ON) all expect to source the configuration file from /usr/local/AUTOSKY. If you move the BASEDIR and you are running multiple copies of ASW, you must edit the "source" line in the shell scripts to the new BASEDIR.

If you are running a single copy of ASW in a different BASEDIR then you can either: 1) edit the 14 shell scripts as mentioned above or 2) mkdir /usr/local/AUTOSKY and copy the autoskywarn.conf file to that location.

INFORMATIVE SOUND MESSAGES

The "ALL CLEAR" message is in .../SOUNDS/asn96x.wav (can be disabled)

This wav file (all ASW WAV files are 8K) has two items that you might want to change. The lead sound effect that is designed to be unique and grab attention. There are now several 96 series messages. Pick the one you like or using your favorite sound editor you can change or remove these sounds.

The "CONDITIONS CHANGED" message is in .../SOUNDS/asn97.wav (can disable)

This wav file has a lead sound effect that is designed to be unique and grab attention. Using your favorite sound editor you can change or remove the sound. This is also called INFORMATIVE MESSAGE.

The "ERROR ALERT" message is in .../SOUNDS/asn98.wav (can not disable)

You should never hear this sound. It sounds when an event from NWS does not match the expected list of events. If you do hear it, please quickly make a copy of file "alert.save" and alert.save versions ".1, .2 and .3" and send those files as part of an error report.

RUNNING MULTIPLE COPIES OF ASW

As the project went outside the first repeater, almost the first question asked was could ASW support more than one alerting area (multiple county/parish). The software was not developed with that as a goal. Here are some thoughts on how you might be able to do that.

With the addition of Variable BASEDIR in autoskywarn.conf the problem is more one of coordination than capability. Here are some of the things that would have to be considered (not in any particular order).

Let us assume that I want to cover Lafayette (LFT) and St. Mary Parish (STM) from one repeater. AutoSkyWarn was designed to build a weather tail message (force INFORM / force SAYTAIL / BOBTAILING are added features).

- A. You install the software twice ... once as /usr/local/AUTOSKY-LFT and once as /usr/local/AUTOSKY-STM (this includes changing the "source" lines as described in a previous section).
- B. Each parish would have its own control system AutoSky.*[ON|OFF] but the user would really have to understand the ramifications of using the forced messages (INFORM[EF] and SAYTAIL). These two messages would have controlled time of presentation (first RX drop after each 4 minute crontab window).
- C. You will have two "wx-tail.wav" files (in /usr/local/AUTOSKY-[LFT|STM]). You can put them into rotation easily via rpt.conf but the timing of when they speak is

effectively not scheduled. That is, you will not know which weather tail message you are hearing. This can now be addressed with AUDIO TAGGING

D. rpt.conf can be edited to execute a forced local play of **each** wx-tail message (manual announce example below).

E. I believe the simplest way to identify which events are for which parish is to use the "Attention Sound" a.k.a "Audio Tic" ... default is the wood block sound that appears before the name of each event. So leave LFT default but in STM change the sound to a beep, boop or ding.

F. A second way to do this (have to change code in several places) is to put the parish name at the front of the tail message and the INFORM[FE] / SAYTAIL messages. It makes the message longer. But ... This is no longer a problem in ASW Version 6! See the section on AUDIO TAGGING.

G. Maybe all of the SOUNDS/asnXX.wav files could be re-recorded. If I was doing this I might use LFT as a female voice and STM as a male voice. I don't like this as much as "F" above.

H. It would be possible to have a shell script run continuously that combines the LFT and STM wx-tails. That combination could be pointed to in rpt.conf as a single wx tail message. IF you turn off INFORM[EF] and SAYTAIL then both installs can be run at the same time (I don't know if a RPI has enough horsepower to do this).

I. This appears to be extensible beyond two parishes. The limitation appears to be how much time you want the system to be talking.

J. The function of section "F" above is included in AutoSkyWarn 6A. It is called Audio Tagging and is described below.

K. Audio files can be made by Text To Speech at <http://www.fromtexttospeech.com/> using settings US English / Alice / medium. Some "extra effort" is needed to speak some words like "Atchafalaya" or "Acadiana" and you will probably run into this problem. Just spell it like it sounds. Use something like Audacity to trim down the audio and convert it to an 8K, mono sound clip. I do use different voices for Weather, IDs and Earthquakes (yes, working on AutoQuake).

AUDIO TAGGING

Audio Tagging allows you to add audio (words or sounds) to 4 locations in the audio flow of ASW. You can tag at these locations ...

- 1) Leading or Following the All Clear Message
- 2) Leading or Following the Weather Tail Message

The Audio Tagging is control by the presence or absence of an audio file. The software looks in `/usr/local/AUTOSKY/SOUNDS` for the following file names. The function type is “embedded” in the file name. If the file exists the audio is automatically included in the announcement at the specified location.

- `countyle.wav` – Leading the **Empty Message** (a.k.a. All Clear message)
- `countyfe.wav` – Following the **Empty Message** (a.k.a. All Clear message)
- `countylt.wav` – Leading the weather **Tail Message**
- `countyft.wav` – Following the weather **Tail Message**

Audio Tagging was added for a different reason locally, but it should function to allow multiple areas to be reported by one AllStar node (See Running Multiple Copies of ASW).

Remember the “county” in the file name is just a name. It can carry any audio information required. They are just files so remember that you can use shell scripts to Create, Replace and Delete the audio tagging.

REPEATER CONTROL

The Repeater Control function allows ASW to call shell scripts located in `/usr/local/AUTOSKY/SOUNDS`. The user edits shell scripts `RptCtl0` and `RptCtl1` to call other shell scripts or (not recommended) putting commands directly into the file. Repeater Control can be enabled or disabled using the commands `AutoSky.RptCtl.[ON|OFF]` just like other controllable ASW functions.

`RptCtl1` is called at the end of processing an event change in the National Weather Service information. N.B., it is possible for this script to be called multiple times in a row. Every time conditions change (except for All Clear) this script is called. This script does whatever the user wants it to do and can be thought of as

“SkyWarn Mode ON”. RptCtl1 is called with 6 parameters: number of warnings, number of watches, number of advisories, and number of special statements, if BobTailing occurred (0=No 1=Yes) and the number of events processed to make the weather tail message.

RptCtl0 is called at the end of processing an All Clear message. N.B., it is possible for this script to be called multiple times in a row. This should only happen during testing of the software. This script does whatever the user wants it to do and can be thought of as “SkyWarn Mode OFF”. RptCtl0 is called with one parameter, a flag that signals if the list of events is empty because NWS has no further WWAS or because BLOCKING has emptied the list. Reading the Blocking document will shed some additional light.

OTHER SOFTWARE COMMENTS

A) 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. To test if already installed use “which sox”, if you get a pathname it is installed.

B) 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. To test if already installed use “which festival”, if you get a pathname it is installed.

C) Not required for ASW but extremely useful is the "rpt" shell script. You can move the script to a directory in the PATH like “/usr/local/bin” From the Linux prompt it allows you to execute DTMF commands directly. Use of "*" is optional, for example:

```
# rpt 71
```

```
# rpt *12500
```

D) Shell Script “playasn ##” causes file SOUNDS/asn##.wav to be played

E) Crontab Entry Example:

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

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

F) Controlling ASW via DTMF Command – examples from rpt.conf

```
84=localplay,/usr/local/AUTOSKY/wx-tail ;Play Current WX Messages
90=cmd,/usr/local/AUTOSKY/AutoSky.Master.OFF ;AutoSky Master OFF
91=cmd,/usr/local/AUTOSKY/AutoSky.Master.ON ;AutoSky Master ON
92=cmd,/usr/local/AUTOSKY/AutoSky.Informf.OFF ;AutoSky Inform OFF
93=cmd,/usr/local/AUTOSKY/AutoSky.Informf.ON ;AutoSky Inform ON
94=cmd,/usr/local/AUTOSKY/AutoSky.SayTail.OFF ;AutoSky Say Tail OFF
95=cmd,/usr/local/AUTOSKY/AutoSky.SayTail.ON ;AutoSky Say Tail ON
96=cmd,/usr/local/AUTOSKY/AutoSky.BobTail.OFF ;AutoSky BobTail OFF
97=cmd,/usr/local/AUTOSKY/AutoSky.BobTail.ON ;AutoSky BobTail ON
99=cmd,/usr/local/AUTOSKY/AutoSky1 ;AutoSky – Force Update
```

(more)

LIST OF SUPPORTED WATCHES/WARNINGS/ADVISORIES/SPECIALS

Blizzard Warning	Flood Warning	River Flood Warning	Thunderstorm Watch
Coastal Flood Advisory	Flood Watch	River Flood Watch	Tornado Warning
Coastal Flood Warning	Freeze Warning	Severe Thunderstorm Warning	Tornado Watch
Coastal Flood Watch	Freeze Watch	Severe Thunderstorm Watch	Tropical Storm Warning
Dense Fog Advisory	Freezing Rain Advisory	Small Craft Advisory	Tropical Storm Watch
Excessive Heat Warning	Frost Advisory	Special Marine Warning	Wind Advisory
Excessive Heat Watch	Gale Warning	Special Weather Statement	Wind Chill Advisory
Extreme Wind Warning	Heat Advisory	Storm Surge Warning	Wind Chill Warning
Fire Weather Watch	High Wind Warning	Storm Surge Watch	Winter Storm Warning
Flash Flood Warning	High Wind Watch	Storm Warning	Winter Storm Watch
Flash Flood Watch	Hurricane Force Wind Warning	Thunderstorm Warning	Winter Weather Advisory
Flood Advisory	Hurricane Warning		
	Hurricane Watch		
	Ice Storm Warning		
	Red Flag Warning		