DireWX

A Direwolf add-on written purely in Python

Write-up by Owen Graham, KE0SBX

January 18, 2025

**INTRODUCTION**

This is a Python script that is used to fetch API data from the National Weather Service’s CAP API (Common Alerting Protocol) and retransmit the fetched information over the Automatic Packet Reporting System via Direwolf. You can use this add-on for any packet radio application such as an APRS digipeater, a BBS packet node, and more! Or, you could make a standalone radio station that beacons out packets for your area.

As of right now, the script only returns the alert type (Tornado Warning, Winter Weather Advisory, etc), but because it is open source, this means that you could return more data values within the fetching method.

**HOW TO INSTALL**

Installation is simple – just download the zip file into a directory of your choice, but do make a note of where you downloaded it! We’ll need that path later.

**SETUP AND CONFIG**

We’ll need to now open the direwolf.conf file. Scroll down to the bottom of the file until you get to the beacon section. You’ll need to now create a CUSTOM beacon. When you create a custom beacon in Direwolf, you need a few things:

1. The type of beacon (CBEACON, PBEACON, etc.)
2. Any time constraints (delay before sending first packet, repeat every x minutes)
3. The data needed to transmit

To create the beacon, simply add the following line in the direwolf.conf file (all on one line). This line creates a CBEACON, adds a startup delay of 30 seconds, transmits and executes the Python file every 2 minutes, and issues the command to execute the Python script:

On Linux, type:

CBEACON DELAY=00:30 EVERY=02:00 INFOCMD=“/pathtoyour/pythonfile/pythonscript.py”

On Windows, type:

CBEACON DELAY=00:30 EVERY=02:00 INFOCMD=“python /pathtoyour/pythonfile/pythonscript.py”

The infocmd field is important as this will issue a command instead of a preset information field like what we would use if we ran a plain info field. The plain info field beacons a preset message.

Change the startup delay to however long you want as well as the beacon frequency. Note: The beacon frequency should be set so it transmits at a reasonable pace. Anywhere from 2 to 5 minutes per beacon should suffice if the frequency is relatively quiet. One thing to also note is that the National Weather Service has a set rate limit on its API, so too frequent beaconing could throw a 429 error, or too many requests.

If you run Direwolf as it is up to this point, you will receive an error message – this is normal.

We now need to set the Python file as an executable. Only issue the following below if you are on a Linux system. For Windows, just ensure that you have permissions enabled on the file and the folder(s) it’s within.

chmod +x pythonscript.py

We also need to edit only one field in the Python script, so you’ll need to nano into the file or use whatever environment you prefer. Remember you need to still be in the same directory we executed the chmod command in:

nano pythonscript.py

There should be a commented area asking you for a UGC code, or a Universal Geographic Code. This can be found at <https://alerts.weather.gov/>. On the page, select “Land areas with zones,” find your state, and select “County Zones” on the right. This should give you a 6-character code that corresponds with your county. For mine, Nodaway County, I’d use MDC002. Once you’ve edited this code, save the file and close. You should now be able to run Direwolf with severe weather alerting capabilities!

**NOTE ABOUT ALERTS**

No one can be certain except the National Weather Service when a tornado will strike your place, a winter storm unleashes a snow squall – no one really knows.

In a nutshell, ALWAYS RELY ON THE NOAA WEATHER RADIO for the latest severe weather information. This should ONLY be used as a backup!

I have not tested this on any other platforms such as Windows or Mac. If you have the capability and you’d like to try it, go for it! This is open source software after all.

If you have any questions, concerns, or comments, let me know! I hope this guide helps you a bit.

73s de KE0SBX