Weather Bot v2 Documentation of Important Changes

Information in this document is to summarize and discuss changes being made to the v2 version of the bot compared to the v1 version. Important notes and discussions will be made more visible, and important changes will be documented. Highlights, bolds, and italics will be used emphasize different parts of this document. Highlights will be done to show the main differences between v1 and v2, as well as to mark any important notes. You can read just the highlights and still get the main idea of these changes.

Changes to Handling of and Broadcasting of Alerts

CONFIRMED: PARTIALLY IMPLEMENTED

Version 2 will bring with it updates to the alert information and the way the alert is broadcast across different channels. Some noticeable changes have been made, and some are more technical than others. However, these have all been done within reason.

New alerts with v2 will now contain several key parts of a message:

(1) (#TrackId) Message header (title of the alert)
(2) Preamble
(3) Main alert information
(4) Precautionary/Preparedness Actions (Sometimes)
(5) Tail
(6) Alert Information

The message header will be the same between v1 and v2, where the main header for the alert is stored. Note that the trackId has been moved to the header, and a Preamble and Tail was added.

The most major and main difference between v1 and v2 stored information for alerts is:

(1) messageType – Tells what type of alert this is, between "Update" and "Alert".
(2) Base – Internal reference for coordinates, between "polygon" and "area".

(3) References – The list of referenced alerts for the specific alert. This is more relevant while the alert is active, not so much when the alert has expired.

(4) replacedBy and replacedAt – Two new key features which will be implemented with the purpose of determining whether or not an alert is the repeat of another alert, or an update to a previous alert. Logic is still being determined for handling this information, but a new trackid will not be issued for alerts which replace alerts, only for new alerts.

(5) BLOCKCHANNEL – BlockChannel is the determiner for which channels weather alerts are not broadcasted, and includes types such as "EAS", "CMAS", and "NWEM". "PUBLIC" is also a valid channel type, but this will never be included in the block channels.

(6) VTEC – Formatted as "/O.CON.KMLB.FL.W.0002.000000T0000Z-000000T0000Z/"

(7) AWIPSidentifier – The identifier in AWIPS that includes the alert type plus sender.

(8) WMOidentifier – Another identifier specific to a local WFO.

(9) EventMotionDescription – Storm-only information, provides specific information on storm motion and location.

(10) ExpiredReferences – The list of expired references for that specific alert. Ie., for a flood alert, a complete and total list of all previous alerts which were sent to replace an alert before them. Being saved currently for internal purposes.

The alert **preamble** will include the WMOidentifier, AWIPSidentifier, VTEC, eventName, and the name of the sender.

The alert **tail** will include EventMotionDescription as it is applicable.

This new information will need to be assessed in order to figure out how it fits functionally into our system.

At the current time of writing this: BLOCKCHANNEL's reliability needs to be confirmed. The replacedBy and replacedAt parameters need to be assessed.

After some thinking, it would appear comparing references and checking the alerts mentioned in references and comparing them to our current alerts would be the optimal method. We can make an api call to the specific alert itself the same way we make a call to the alerts.

Alternatively, a different method of fetching the alerts could be implemented. Instead of getting *just* active alerts, a parameter can be passed into api.weather.gov which allows to call for a specific start time for alerts, and fetch all alerts, even ones no longer in effect. Two dictionaries can be made: "ActiveAlerts" and "OldAlerts". Depending on what's what, we can compare "ActiveAlerts" and "OldAlerts", or fetch alert information and check if the new alert has the "replacedBy" and "replacedAt" parameter.

Pings Based Upon Context

CONFIRMED: NOT IMPLEMENTED

The logic for this would be simple: for alerts replacing alerts, no ping would be necessary, unless important information was changed on the alert. Ie., if a Severe Thunderstorm Warning is upgraded to a "CONSIDERABLE" tag, a ping would be necessary and would reflect the fact that the alert was upgraded. To make this change clearer; pings would be included *after* the alert message, not *before* the alert message.

Ultimately, the goal would be to, in the future, make pings contextual to the alert, and to begin supplementing information for the alerts based upon the alerts, but via creating new information discussing previous alerts. This is where the TrackId would become useful, as it would allow for the easier comparison of alerts, as the NOAA api does not provide a clean method for comparing alerts. Updates to previously issued alerts could be done and users could compare between alerts easily.

In example, a Hurricane Watch being upgraded to a Hurricane Warning would still require a ping, because this is an additional notification saying that Hurricane conditions *are* expected in the near future. However, say, a Severe Thunderstorm Warning receives a Severe Weather Statement. If the Severe Weather Statement is a small update to the Severe Thunderstorm Warning with no noticeable changes, no ping would be issued.

Update To TrackId

CONFIRMED: NOT IMPLEMENTED

The current trackid is a misc feature, and is a number that increments when new alerts are issued. Eventually, this number *could* become larger, though by the time that happens I doubt I will be involved with the club, and the code would likely be maintained by a different person. At that point, the decision on trackids would be up to them.

Regardless, the current trackids would be reset.

Instead of being a misc counter, the trackids will instead become a feature for easily knowing which alert is being referred to. This is important, because even in the cases like the FLS where it is an alert being replaced, the new alert still provides information that could be considered useful, even if no one in the server currently finds it to be. However, instead of generating a unique alert id every time the alert is replaced, it will maintain the same id. This is because the alerts effectively are the same, and have no reason to be considered unique. This same logic will apply to the SWSs (Severe Weather Statements), or to any alert which is being replaced.

Though a Hurricane Warning replacing a Hurricane Watch would not receive an alert, a Severe Thunderstorm Warning, even in an active Severe Thunderstorm Watch, would. I would think the reasoning here is clear; but it's because a SVR and SVA are different alerts entirely.

In any case, moving forward, trackids will no longer be an incrementing number, and instead will be a string. They will be issued as follows:

#AAA-00000

This is a difference to the previous, where TrackIds are just #INT.

This provides a clear and easier to digest id. As it involves disseminating weather alert information, this is important, because knowing which alert is being referenced is key for properly issuing severe weather information, and references are important for *any* weather information. Effectively, this change is being done to improve clarity. If it is of concern, this format provides a total

1,757,600,000 unique ids. This will be hard to cap out. [redacted]

However, changes may need to be considered to this format.

Changes to Mapping and Geometry

CONFIRMED: IMPLEMENTED

Important changes have been made to mapping and geometry, including internally.

To state it once more, this change was done with the objective of better disseminating weather information, as the main goal of these changes has, and always will be, to include relevant information to the alerts, and to release the information in a readable, actionable method, as well as providing visuals [redacted]

Things are being considered for accomplishing the aforementioned task, which will be discussed later.

Internally, geometry.py has been updated for alert image generation to now take *any* polygons or *list of* polygons and will make them a *MultiPolygon* .

Effectively, alerts will now *always* have coordinate information, but that coordinate information can vary. It can be a list of polygons or one singular polygon, but to make transitions easier, alert mapping will always convert it to a MultiPolygon .

The MultiPolygon class allows us, internally, to make one singular polygon object via creating multiple polygons. This prevents weird lines being made when trying to draw several polygons, which is important with the latest change.

Do note, however, that the ucf_in_or_around_polygon will always take a single polygon; for coding purposes, assume that if Orange County is listed and no individual polygon was produced with the alert, that UCF is included in the alert, in which case it is unnecessary to run the aforementioned function. The variable base provides a simple method for determining if the alert is mapped to an area or a polygon. The base variable will either be Polygon or Area .

From now on, several key features have been changed as it comes to mapping and creating alerts images:

1. Aforementioned internal use of MultiPolygon ;
2. The polygon color will be generated and attached to the alert during the alert curation process;
3. Alert images will now contain a title;
   - (a) leads with "Alert Area";
   - (b) then the SAME code for the alert;
     - (1) This filters between SAME and NWS listing
   - (c) then the trackId.
4. All alerts will produce an alert image:
   - (a) based upon affectedZones information, counties or forecast zones will be mapped;
   - (b) the zones utility makes an api call for each zone connected to a monitored county, and then places them into a table for reference;
   - (c) this table returns the geometry for that zone or county;
   - (d) we will now use that geo information for producing an alert image for *every* alert.

It is being considered to include more geometry on alert images, ie., providing more detailed parks, lakes, and river information (assuming we can get it working).

Reference natural earth for these libraries.

Expanded Hurricane Information

DISCUSSING: NOT CONFIRMED

As previously mentioned in a previous, unreleased document, expansion of Hurricane Information is something to be considered.

Expansion of hurricane information would include fetching cones, wind probabilities, wind arrivals, current forecast track, etc. for storms which are to impact Central Florida, or are within range of Florida and may bring impacts to

Florida. Supplying this information, plus receiving alerts from the National Weather Service and via NWS forecasts, information can be issued for storms and more information can be gathered by users in the Discord.

Furthermore, posts for Tropical Genesis Outlooks will only be issued during the active hurricane season – June to November. Outside of these times, the bot should post at the start of the month that the hurricane season is not currently active. Once we approach hurricane season, the bot should post a hard-coded message discussing preparation for hurricane season, and what resources FEMA, as well as other resources, recommend, as well as a list of essentials.

All information above has not be implemented, it is to be considered.

Alert Counter

TO IMPLEMENT: PARTIALLY CONFIRMED, NEEDS DISCUSSION

As its own service, alert numbers will be tracked each day.

At the end of the day, around 10pm, the bot should post a message detailing the day's information, including two boards:

(1) The top 5 counties with the most alerts;
(2) The top 5 alerts issued.

This end-of-day-summary can also include rainfall amounts from observation stations located across the monitoring area. These stations can be a list that users input manually from the NWS api. However, a hard-coded limit for how many stations to reference will be included, as too much information will bog down the user and make them less likely to read the information, even if this information is not critical.

Plans for Reformatting of NWS Alerts

DISCUSSING: NOT CONFIRMED

Ultimately, sort of per request, alerts should eventually be reformatted into our own style, and that style should be cleaner, more readable, and effectively convey the information that needs to be conveyed.

It is important to note, however, there are **strict** conventions for reformatting these alerts, and they should be compliant with FEMA's guide for properly announcing IPAWS alerts. Ref FEMA's guide for proper reformatting and re-issuance of alerts.

Inclusion of Severe Weather Outlook Information and Mesoscale Discussions

DISCUSSING: NOT CONFIRMED

Implementation of SPC severe weather outlooks and mesoscale discussions is information related to severe weather. Disseminating this severe weather information could prove useful and maybe help prepare people for severe weather impacts.

SPC outlooks would be fine tuned to not include the full information, and, instead, fished through to find relevant information pertinent to the area.

The outlooks would be issued based upon whether or not a severe risk exists. For example, if no risk exists on the D1, D2, or D3 outlooks, nothing would be posted by the bot. However, if a risk is included, it will be posted. Any day which includes a severe risk would have the relevant information pushed to the forecast channel.

Furthermore, as previously mentioned, logic would be integrated to shorten the total length of the SPC severe weather outlook discussion, and would primarily serve to highlight what impacts Florida has from the severe weather outlook. Unnecessary information would be promptly cut from the text product issued by the bot. Assessments could be made via utilizing other RSS feeds by the SPC to determine severe weather hazards, and highlight which areas are to expect what impacts.

Mesoscale discussions serve to discuss severe weather features. They also discuss what is currently going on in the atmosphere, as well as watch probabilities. Though, at a first glance, this is more technical information, such information could be utilized to assess the current severe risk, perhaps not by the daily user, but maybe by more experienced users who understand more of the dynamics in the atmosphere.