JS8Net de WH6GGO

User Guide

v 1.0.2

Author: Lawrence Byng

Date: July 2022

Table of Contents

Preamble

What is JS8Net?

JS8 Net is a python application developed specifically for hosting JS8 ham radio nets typically on HF. The application provides many enhanced features that are simply not acheivable using a keyboard-only style of operating. An overview of some of the main features is as follows:

- Roster to display participating station call signs, operator name, net status and offset
- Point and click style of net operation.
- Can be run as net control station view or as participant station view
- Convenient notepad area to pre-type ‘report’ and ‘announcement’ type communications prior to the net
- Auto save feature
- Built in Dynamic Content Macro language allowing construction of wide variety of point and click messages. The Macros provide access to the relevant fields and functions of JS8Net.
- Customization feature allowing customization of all point and click messages
- Built in state machine and text mode parser to track progress of the net. This allows participant stations to view upto date net information as if connected via the internet.
- No internet connection required. Uses text mode communication over JS8 only.
- Real time tracking of SNR, bad frames and time delta for each station.
- Offset plan distribution functionality built in.
- Use of the JS8Net application is optional. Stations using JS8Call only as well as stations using JS8Call + JS8Net are both able to participate in the net.

Why was JS8Net Developed?

I developed JS8 net primarily as a tool for facilitating running a JS8 net. After having run several JS8 nets using a keyboard only approach, it seemed to me there was an opportunity for a software solution to assist net control run the net. Using a keyboard only approch requires lots of fast typing and results in lots of typos and frequent delays while responses are constructed and reviewed prior to sending to the net. The whole process seemed to me to be less than optimal so this was the initial impetus to build the app.

As a result you will see there is large emphasis on allowing the majority of net control features to be handled as simply point and click operations with the minimum of typing. There is however still the flexibility that allows net control full control over message content and does not preclude any of the keyboard only mode type actions for constucting and sending messages.

Providing an enhanced visual experience for net participants was also a consideration. Features such as a visual roster, indication of who has taken their turn and who is next as well as details about the net itself such as start time, frequency and group name. Additionally, the ability to communicate this information out to the various participant stations as if they were connected via internet but without the use of any internet connections was also high on the list.

In order to provide this station synchronization functionality, the application needs to be able to track the progress of the net and parse the mssages from other stations and net control. This requires a fairly elaborate state machine and parsing logic. At the core of JS8Net you will find a state machine that handles the various transitions between different net states and different participant states as the net progresses. In addition JS8Net uses a novel approach of parsing the various messages and attempting to follow along based on the content of the various text messages going back and forth. This is completely experimental and unique but so far it appears to work quite well.

On occasion, HF nets can sometimes be a little chaotic and or unpredictable; sometimes people join the net late or two stations double or the signal is not synchronized properly and so on. In order to continue functioning there are several manual overrides. It is good to have these ready in the back of your mind just in case propagation or some other factor causes some interesting net conditions.

If all else fails you can still fall back to using text typing only in the ‘other’ multi line entry field on JS8Net or you can switch back to JS8Call and type directly into the send text field there. The advantage of using the ‘other’ field in JS8Net is that the text will not dissappear if you abort the transmit by clicking the TX button in JS8Call. Sometimes you may decide to abort transmission for example if another station starts sending their message right at the same moment you did. The ‘other’ field will keep the message on screen until you delete it. The ‘other’ field will also pre-pend the necessary groupname onto the front of the message so you do not have to worry about setting that manually.

So How Do I Run a Net with JS8Net?

First if you are one of those people than can type everything perfectly at 100 words per minute and then review it instantaneously or prefer to use keyboard only with no additional software when using digital modes, then this software solution is not for you so please close this manual and application and go ahead and continue with what you were doing.

If however you are like me and your typing is less than perfect, and you need some time to read thru and review what you typed and then correct it not to mention having to squint a little when reviewing the mssage, then please read on. Yes there is a small learning curve when you first start to use the application but in my view it is worth it. Having used JS8Net extensively to run JS8

nets for some time now, I find it very usefull and lots of fun also. I would not consider running the net any other way currently. Remember, JS8Net was born out of necessity and designed and engineered by someone who actually runs a JS8 net on a regular basis...me.

Probably the best way forward here is to give a start to finish example from the perspective of net control running the net, along with an explanation.

How to Run a Net using JS8Net

Starting the Application

First start JS8Call. Make sure you have enabled the external TCP connections interface in JS8Call.

Now start the JS8Net application in net control mode as follows...

```
$ python ./js8_net_client.py --interface=netcontrol
```

This will use the default save file for saving and loading any data used by the spplication. The default data file provided on github is a good starting point and should be used to start off with.

Once the application is installed and configured corretly, when you start it up a window will display. This is the main window where all net activity occurrs.

Fill out the top three rows of the screen with information about your JS8 net. All data is required and will be used in subsequent steps.

If there is any data on the roster this will be from the last time the application was run. Click the ‘Clear’ button to clear the roster data. Now the only data on the roster will be your call sign and <NCS> showing that you are net control station.

If the name field shows as ‘-’, click on your call sign on the roster then in the name entry field put your name then click ‘update’. Now the roster should display your name along side the call sign. The name will be saved in the save data, so once the name is set this will be remembered by the application indefinately.

You can go ahead and add any call signs with names manually to the roster now in the same manner. These will also be saved in the save data file. It is a good idea to initially enter the names manually so that the application has some data saved so that when the auto check in feature is used, the application can automatically add the station and the name to the roster automatically.

Sending a QST

First make sure ‘Flash Buttons’ is checked. This will flash the buttons that are relevant as a guide as the net progresses.

Second, set the mode for passing the data to JS8 call as ‘Post to JS8Call only’. This mode will set the text in JS8Call and allow it to be previewed prior to sending. A second mouse click is required to actually send the message. This is done clicking the ‘Go’ button to the right of the preview window at the bottom of the screen. Posting first then sending is a good place to start and allows greater control over sending the messages. The other option is ‘Post to JS8Call + Send’. This option is for advanced users only. It will automatically send the message after about 2.5 seconds delay allowing automatic sending with even less clicks of the mouse.

Prior to the net, typically 30 minutes prior and 15 minutes prior it may be usefull to send out a QST to all stations as a reminder. If you look at the lower portion of the screen, you will see two rows of 7 buttons each. The top left of these buttons is marked ‘QST’. Go ahead and click this button. The preview window in JS8Net will now show the QST message based on the net control info that you entered in the starting the application step. Now click the ‘QST’ button a second time. You will see that the message preview window provides a different format but largely the same content to the message. Most buttons have several different message formats available. When you have previewed the message and are happy with it, make sure the TX button at the top right of JS8Call is enabled then go ahead and click the ‘Go’ button (flashing red/blue) at the lower right of the preview window in JS8Net. JS8Call will begin sendig the message.

Congratulations! You have now gone thru the initial small learning curve on how to operate JS8Net. This is about 90% of the process of using the application. The rest follows this same pattern but relates to the different stages of the net.

The QST message also sends out much information about the net that can be decoded by any participant station that is also running the JS8Net software. Participant stations can start up the program then clear all of the fields and wait for the QST. When the participant station hears the QST message it will populate most of the fields automatically. At this point there should be sufficient information for the participant station to decide to join the net and construct the appropriate messages. The remeaining fields will be populated when the net starts.

Asking for Checkins

Now click the ‘Ckin?’ button. Prior to the net time that you specified earlier, the messages will be pre-check messages. After the net starts these will change to regular check in messages. Click the ‘Ckin?’ button a couple more times to cycle thru the available message formats. Once you have selected the format you want, go ahead and click the flashing ‘Go’ button. JS8Call will send the message

Processing Checkins

Make sure that ‘Auto Check-in’ checkbox is checked at the upper right of the screen and wait for a station to appear on frequency. When the sation sends a JS8 message their call sign will appear on the roster automatically. If you prefer to add them manually then simply uncheck the ‘Auto Check-in’ checkbox and proceed to enter the station in the three fields above the roster. Initially any stations checked in automatically will appear with a status of <HEARD> or <CHECKIN>. If you know this is a station that wants to be checked in to the net go ahead and click the ‘Got u’ button. This will provde several messages for confirming those stations that just appeard on the roster. Now go a head and click the flashing ‘Go’ button. You will see that the message is sent and also the status of the stations in the roster is updated to <STANDBY>. This means they are ready and standing by for the net. If you are not sure if a station wants to be added or not you can enter free-form text in the ‘Other’ field to ask for clarification and then click the ‘Go’ button. If this station responds that he is not participating you can set his status in the roster to <IGNORE> or <SWL> or you can simply select that station on the roster then click delete to remove it from the roster. At the end of the chckin process you should have a contiguous list of station in the <STANDBY> state. It is usually a good idea to just delete any extra stations. You can optionally turn off auto-checkin checkbox at this point so that the roster does not add any additional call signs.

Opening the Net

When the start time rolls around, you will see the Timer at the top right of the screen change to “ON AIR” This is your cue to start the net. Go ahead and click the ‘Open’ button. Again there are a couple of format options. Choose the one you want then click the ‘Go’ button. After the message is sent you can click the checkin button to ask if there are any more checkins. Note that the checkin messages now are regular checkin message rather than pre-check messages. If there are additional stations to add go ahead and add them in the same way as before in the checkins section.

Any stations that started the JS8Net application late and who wish to join the net can listen to the opening message. The JS8Net application will automatically decode the information and populate the screens of the participant stations.

If a participant station opens up the application later than the opening message, they will need to manually specify the group name and frequency to the application to be able to use it. Without this minimum information, all of the buttons will remain disabled. Specifying this information can be done quite easiy from the command line, from loading the data from the save file or simply by typing it in on the screen.

Sending the Roster

Now click on the ‘Roster’ button then click the flashing ‘Go’ button. All of the stations stations participating in the net will be set to <STANDBY> status and the roster sent out to all of the

participant stations. Any participant station that is also running JS8Net will see the roster appear on their screens that reflects the same roster as is on net control screen.

If you are planning to use an offsets plan, it is a good idea to send this information out immediately folling the roster message. The message can be found on the ‘Extras’ button

Offset Plans

JS8Net has built in a pre-set offsets plan. This is a list of offsets that can be used by all stations participating on the net to ensure that stations are distributed as evenly as possible. The idea behind this is to help minimize collisions if two stations happened to hit transmit at the same time. If the stations have non-overlapping signals then they can both be decoded simultaneously.

In order to use this feature effectively, net control and all of the stations must be running the JS8Net application. After the roster is sent to all stations is probably the best time to send out the offsets plan. This will assign each station an offset based on their relative position in the roster. The macro to send out the offsets plan to all of the stations can be found under the ‘Extras’ button on the main screen. Once the new offsets plan is sent out, each participant station will now have an offset set per the plan. This can be seen in the ‘offset’ field on each of the stations. Each participant station can still override this offset but that will reduce the effectiveness.

If you wish to set your own offsets plan that is different from the pre-set plan provided, This can be done byu specifying you new offests plan on the command line when the application is run.

The offsets plan is a comma separated list. The first offset refers to net control only. The second offset thru the end of the list of offsets is used to set each of the participant stations. If there are more stations than there are offsets on the list, then the list will wrap around an double up, triple up etc stations on the offsets provided until all stations have been assigned an offset. For this reason it is better to provide an longer offsets list to handle a large number of stations and to not use all offsets on the list rather than have a short list and to double up stations on the same offfset just because of taking shortcuts in the pre-planning steps for the net.

Sending Announcements

If you have any pre-typed announcements now is probably a good time to send them. These will be pre-typed in the ‘Announcements’ multi line entry field. No go ahead and click the ‘Go’ button at the immediate right of the ‘Announcements’ field. The message will send. If you do not have any announcements for the net you can simply pre-set the text in this field to ‘Are there any announcements for the net?’ and just click the ‘Go‘ button when you are ready to send the message.

Handing Over to the First Station

Click on the ‘First’ button. You will see that the first station on the roster is now queued up by updating its status to <NEXT>. Now go ahead and click the flashing ‘Go’ button. This will send the message. After a short delay, you will also notice the countdown timer in the middle of the screen starts counting down. This is a guide to time the response of the station. If the station responds, then the status on the roster will be updated to <TALKING>. If the station does not respond in a certain amount of time such as that amount of time from the countdown timer, then you can send a reminder to the station. You can do this by clicking on the ‘Nudge’ button and then click ‘Go’. You will see the countdown timer start again. This timer is as a reference point only. If you decide that the station is MIA then go ahead and skip the station by pressing the ‘Skip’ button then the ‘Go’ button. This will update the roster to <SKIP> and queue up the next station on the roster and update the status to <NEXT>

Recap

Lets pause for a minute to review where we are and what we have acheieved so far. So up to this point, net control has sent out a whole sequence of messages including QST, PreCheck request, Checking in acknowledgements, Net Open, Sending of the full roster, sending an offsets plan and handing over to the first participant all without typing a single letter on the keyboard at all. All we have needed to do so far is click this button then click go, click that button twice then click go, click this button then click go etc etc. One of the design goals was to minimize typing and this is a perfect example to illustrate this.

And on the other side of the equation, each paritipant station who also runs the JS8Net application has been able to receive all of the information for the net, populate all of the fields on the main window and populate the roster and join the net all without typing a single character at the keyboard either. Their station data has been kept in synchronization with the net control station data all without the use of any internet connections!

Directed Nets, NCS Responding to a Station

Once the first station responds, for a directed net that is the queue for net control to construct and send a response. This is where the ‘Next’ button comes in. The next button and the ‘End’ button have some very neat features under the covers. Let me explain…

Once the first station has finished his response and handed back to net control go ahead and click the ‘next button’. You will see a message appear in the preview window and you will also see the roster update to reflect that the first station is <DONE> amd to queue up the next station by marking it as <NEXT>

Now before you click the ‘Go’, go ahead and check some of the checkboxes above the preview window such as ‘UR Welcome’ , ‘Tks’ , ‘SNR’, and the one just to the left of ‘Great Evening’. Now click the next button again. Note how the preview window has been updated with an expanded text version of this information including an SNR report. Now click the ‘Next’ button

again, you should see a full SNR report including bad frames and time delta. Now type some text into the ‘other’ window such as ‘awesome signal from your new vertical antenna and good luck with the contest’. Now click the ‘Next’ button one more time. You will see that the message has been constructed automagically.

If you wish to give the first station another opportunity to talk on his turn on the net, instead, go ahead and click the checkbox to the left of ‘More’. You will notice that the response goes back to the first station but does not hand off to the next station just yet. When you have the format of message you wish to use go ahead and click ‘Go’ to send the message. Once the station responds back a second time, Click the appropriate checkboxes including the one next to the ‘Great Evening’ checkbox then select the format you want by clicking on the ‘next’ button several times then click ‘Go’. This will reply back to the first station and hand off to the next on the roster.

So all of the net control aspects were fully automated and the only text we needed to type was the specific text of the additional part of the message to send to the first station.

If the station does not respond, you can hit ‘Nudge’ and then ‘Skip’ same as before.

Continue thru the roster until all stations had an opportunity to speak.

End of Round

When the last station has transmitted click on the ‘End’ button. This has some messages that relate to the end of the round. You can also use the row of checkboxes for building a message back to the last station including their SNR and so on in the same way as the next button did.

Sending Your Report

You should have a pre-typed report ready to send in the ‘report’ multi line entry field. Go ahead and click the ‘Go’ button that is immediately to the right of the ‘my report’ field. Your report will now be sent.

Station Awards

Included with the program are a couple of extras! These are meant as a fun but also as usefull feedback to the participant stations. These focus on three key areas of weak signal communication; SNR, dropped frames and Time Delta.

There are two pre-coded awards:

1) The station that has the lowest Signal to noise ratio while still maintaining perfect communication. This is the ‘weak signal’ award

2) The station with the time sync that is a closest match for net control. This is the ‘Time Sync’ award

The messages for these awards can be found on the ‘Awards’ button. The stations are computed automatically when the button is clicked by using the most up to date information from the roster. If there are multiple winning stations then these will all be listed in the award.

Questions and Comments

This part of the net may involve much sending of customized messages. To do this you can use the ‘other’ multi liine entry field along with the ‘Go’ button that is immediately to its right. JS8Net will automatically pre-pend any group names onto the front of the message so all you have to worry about is the content itelf then click ‘Go’ to send it.

Closing the Net

Once everything is complete go ahead and click the ‘Close’ button then the ‘Go’ button. You have now completed your first JS8Net.

Manual Overrides

On occasion, HF nets can sometimes be a little chaotic or unpredictable. There are many many possible things that can go wrong aside from propagation effects. Sometimes two stations may accidentally start talking at the same time. If the two stations are on the same offset, this can make decoding a challenge. Also stations may have the clock slightly adrift and not properly synchronized. This can sometimes cause intermittent decode failures especially of the initial call sign. Other potential issues include other stations starting up right over the top of your net...once I had a vara station start blasting right on top of the net frequency literally seconds after I opened the net. And then there is my personal favorite...the station that tracks your offset and sends out a heartbeat signal every 15 seconds right on top of your main station frequency + offset. As you can see lots of potential challenges.

What do you do when things don't run completely as expected? How do you handle these potential issues? In order to continue functioning there are several manual overrides. It is good to have these ready in the back of your mind just in case propagation or some other factor causes some interesting net conditions. Each of these are detailed below.

Manually Updating the Status

This can be very usefull for situations where the next station begins his message but the call sign does not decode. The application attempts to match the offset of the transmitting station with its known offstes but if the transmitting station is using the same offset as someone else then this will not work. The easiest way to fix this issue is while the station is talking, select that station in the roster then click on the status drop down and set the status to <TALKING>. This will not only set the current station status correctly but it will also queue up the next station on the roster. This is a very handy technique to have ready just in case.

Manually Setting the Next Field

The next field is used by the state machine to see which station is next. If you wish to manually override this, click the checkbox immdiately to the left of the next field. Now you can either type the callsign that you wish in there or you can click on the roster and that selected callsign will be entered into the next Field.

Manually Setting the Previous Field

The prev field is used by the state machine to see which station just completed transmitting. If you wish to manually override this, click the checkbox immdiately to the left of the prev field. Now you can either type the callsign that you wish in there or you can click on the roster and that selected callsign will be entered into the prev field

Manually Overriding the Offset

Click the checkbox immediately to the left of the offset field then type in your new offset. This will be used for future transmits.

Manually Updating the Current Round

The current round has a drop down combo with the choices. Select the round manually from the combo.

Altering Dynamic Content Macro Messages

The Dynamic Content Macros produce their message in the bottom preview field on the screen. If most of the message is correct but it just needs a little tweak, you can click the mouse on the text in the preview field and correct it before hitting ‘Go’. You should first select the message format that you are going to use, as by changing the message format this will overwrite any edits that you do in the preview field. Editing the preview field is a last minute alteration that you will do immediately prior to actually sending the message with the ‘Go’ button.

‘Other’ Field

The multi line entry field labelled ‘Other’ can be used to construct free form messages and the ‘Go’ button to the right of this field will send these directly. JS8Net will prepend the group name to these messages so you dont have to worry about fiddling around and setting that.

Please note that when the ‘auto clear’ checkbox is checked, clicking the go button next to the preview window will auto clear the text in the ‘other’ field. This is a very usefull feature when this is used for sending the ‘next’ and ‘end’ messages using the Dynamic Content Macro buttons. When you are using the other field for free-form text outside of that you should use the ‘Go’ button that is directly to the right of the ‘Other’ field as this does not auto clear the text.

Additional Techniques

Just to round out this section on manual overrides, there are three very usefull techniques that are available in JS8Call only and not available via the JS8Net application.

The first of these allows you to change offset on-the-fly actually as your station is transmitting. To do this, in JS8Call you must have the option ‘Allow Tx frequency changes while transmitting’ checked. This is on the ‘General’ / ‘Behavior’ tab. If you see another station start blasting on your transmit offset after you have hit transmit, you can gently steer your signal away from the other station by clicking on the offset numbering area just above the bandscope in JS8Call.

Second, there is a big red/green TX button at the top right of JS8Call. If you start a transmit but you then see another station has just started transmitting (you can usually see residual effects during the silent phase of the frames sequencing when you station is not transmitting), you can click the big TX button and transmits will stop immediately.

Third, just in case you missed a typo earlier, it is not too late to do an on-the-fly edit in the JS8Call transmit text field after a transmit has started. You can still edit the latter parts of the message that have not yet been processed by JS8Call. Just be careful not to get stuck half way thru an edit at the moment the message part is being sent for transmit, as this could make the message less readable rather than correct the issue.

I have found a need to use all of these techniques at some point or other while running a JS8 net. Keep them in mind. They could be usefull.

Dynamic Content Macro Language

This section is included for advanced users only who may wish to edit their own messages instead of using the standard set of messages provided with the application.

Each of the main buttons in JS8 net have behind them a set of messages that make extensive use of JS8Net dynamic content macros. Included in JS8Net is large number of these Dynamic content macros that tie into all aspects of the application.

The macros can be used to construct text messages based on the latest up-to-the-second information in JS8Net. Here is an example of a dynamic content macro in use in an actual message. This example is from the participant station's 'QRT' button

%GROUPNAME %SEQ I AM GOING QRT TKS

%SEQ GOING QRT ALOHA ALL %ENDSEQ

This is one of the more straight forward examples. Each of the dynamic content macros begins with a % sign. %GROUPNAME for example inserts the current group name from the main window. %SEQ and %ENDSEQ allow for a sequency of different message formats that can be cycled thru by clicking the 'QRT' button multipl times. The text between the first %SEQ and the next %SEQ or %ENDSEQ is shown when the 'QRT' button is clicked the first time. Next time the 'QRT' button is clicked, the text changes to that between the second %SEQ and the %ENDSEQ macros. So this results in the following…

on the first click the message show in the preview window is:

@MYGROUP I AM GOING QRT TKS

and on the second click the message the preview windowchanges to:

@MYGROUP GOING QRT ALOHA ALL

on the third click the message cycles around again and show the first message and so on. So it is a very convenient way to pre-write several different message formats that can be clicked thru in sequence during the net and sent as a message.

Click Sequencing thru Macro Messages

%SEQ %ENDSEQ

Allows multiple different formats of message to be created for click sequencing.

%SEQ text-1 %SEQ text-2 %SEQ text-n %ENDSEQ

Time Related Macros

'%ZULUTIME

in line replace with the current zulu time

%LOCALTIME

in line replace with the current local time

Net Field Macros

%NETTYPE

in line replace with the value from the ‘net type’ field. This will be wither ‘Directed’ or ‘Round Table’

%NETSTARTTIME

in line replace with the value of the net srt time field

%NETFRE

in line replace of the net frequency field

%NUMROUNDS

in line replace of the number of rounds field

%CURRENTROUND

inline replace for the current round field

%GROUPNAME

in line replace for the group field

Net State Macros

%IFNETSTARTED %ENDIFNETSTARTED

This macro will include the text between the bookend macros only if the net is started. This is determined by comparing current zulu time to the net start time.

%IFNNETSTARTED %ENDIFNNETSTARTED

This macro will include the text between the bookend macros only if the net is *not* started. This is determined by comparing current zulu time to the net start time.

%IFPREV %ENDIFPREV

This macro will include the text between the bookend macros only if the ‘Previous’ field is not empty.

%IFNPREV %ENDIFNPREV

This macro will include the text between the bookend macros only if the ‘Previous’ field *is* empty i.e. if there is not a previous station

%IFNEXT %ENDIFNEXT

This macro will include the text between the bookend macros only if the ‘Next’ field is not empty.

%IFNNEXT %ENDIFNNEXT

This macro will include the text between the bookend macros only if the ‘Next’ field *is* empty i.e. if there is not a Next station

Checkbox Macros

The set of standard messages included with the application use these on the ‘Next’ and ‘End’ buttons. These however can be used on any of the buttons by using the customization –edit command line option to override the set of standard messages.

%IFSNR %ENDIFSNR """"

This macro will include the text between the bookend macros only if the ‘SNR’ checkbox is checked.

%IFGOODEVE %ENDIFGOODEVE

This macro will include the text between the bookend macros only if the good evening checkbox is checked.

%GOODEVE

This inline replacement macro replaces the text with the value of the combo_aloha combo box.

%IFOTHER %ENDIFOTHER

This macro will include the text between the bookend macros only if the good ‘other’ checkbox is checked.

%OTHERMSG

This inline replacement macro replaces the text with the text that was typed into the ‘Other’ multi line entry field.

%IFFURTHER %ENDIFFURTHER

This macro will include the text between the bookend macros only if the ‘further’ checkbox is checked.

%IFNFURTHER %ENDIFNFURTHER

This macro will include the text between the bookend macros only if the ‘further’ checkbox is *not* checked.

%IFTKS %ENDIFTKS

This macro will include the text between the bookend macros only if the ‘TKS’ checkbox is checked.

%IFNTKS %ENDIFNTKS

This macro will include the text between the bookend macros only if the ‘TKS’ checkbox is *not* checked.

%TKSMSG

This inline replacement macro replaces the text with the value of the combo_tks combo box.

%IFURW %ENDIFURW

This macro will include the text between the bookend macros only if the ‘Ur Welcome’ checkbox is checked.

Roster Macro

%ROSTER

This inline replacement macro replaces the text with the full roster including all call signs and names.

Checkin Macros

%IFCHECKIN %CHECKINCALL %ENDIFCHECKIN

This macro is used to construct the automatic reply to any station that checks into the net. The default message is:

%IFCHECKIN OK I HAVE ADDED %CHECKINCALL TO THE ROSTER. GE TKS FOR CHECKING IN. %ENDIFCHECKIN

Call Sign Macros

%NCSNAME

This returns the name of the NCS station

%NCSCALL

This return the call sign of the NCS station

%NETNAME

This returns the name of the net from the main window

Offsets Plan Macro

%OFFSETSPLAN

This macro does an inline replace with the current offsets list.

Station Signal Macros

%LBADFRAMES

This macro returns the number of bad frames that have been counted so far for the last transmitting station.

%LTIMEDELTA

This macro returns the time delta for the last transmitting station.

Last Station Macros

%LC

This macro returns the call sign of the last transmitting station.

%LN

This macro returns the name of the last transmitting station.

%LSR

This macro returns the signal to noise ratio number for the last transmitting station.

Next Station Macros

%NC

This macro returns the call sign of the next queued up station.

%NN

This macro returns the name of the next queued up station.

Award Macros

%WEAKSIGNALAWARD

This does an inline replace with the calculation of the station that wins the weak signal award.

%SYNCAWARD

This does an inline replace with the calculation of the station that wins the time sync award.

Edition Field Macros

%PROF

This macro returns the value of the ‘Edition’ field on the main page. This is typically set to the day of the week and is useful in constructing the opening message to the net.

%PROF(...) %ENDPROF

This macro is used to conditionally include the text bettwen the bookends if the text in the brackets maches the content of the ‘Edition’ field on the main page.

Example:

%PROF(Tuesday) It is Tuesday! %ENDPROF

This will include the text ‘It is Tuesday’ if the contents of the edition field is equal to ‘Tuesday’

Select Message Format by Random Choice

%CHOICE %CHOICE %ENDCHOICE

This macro will conditionally include one of the text phrases specified based on randmo. The phrases are between the bookends %CHOICE / %CHOICE and %CHOICE / %ENDCHOICE.

Example:

%CHOICE Good evening %CHOICE Hello and welcome %CHOICE greetings %ENDCHOICE

in the above example, the included phrase will be chosen at random from the three phrases provided.

Pre-coded Phrases

This section is included for advanced users only who may wish to edit their own messages instead of using the standard set of messages provided with the application.

When editing messages, it is important to consider the parser has several pre-coded paring phrases. As long as your edited message contains one or more of the pre-coded phrases then the station synchronization with the parser will continue to function. If however you alter the text of a message so that none of the pre-coded phrases appear in the message then that will essentially break the synchronization feature.

A list of the pre-coded phrases for each main section of the state machine are provided below

ROSTER:

general format: WH6NCS: @HINET ... ROSTER IS: <CALLSIGN> <NAME>, <CALLSIGN> <NAME> …

There is nothing to break in this hardcoded phrase as all of the information is inside the macro iteself.

QST

specific format: ... THE <NET NAME> STARTS AT <START TIME> ZULU ON <NET FREQUENCY> MHZ. PLS USE <NET GROUP> GRP ...

example: WH6NCS: @HINET QST QST THE HAWAII JS8 NET STARTS AT 04:30 ZULU ON 7.095MHZ. PLS USE @HINET GRP

This is a very specific format and should not be altered

Opening the Net

specific format 1: WELCOME TO THE <PROFILE> EDITION OF THE <NET NAME> NET. GE THIS IS <NCS NAME> UR HOST. THIS IS A <NUM ROUNDS> ROUND <NET TYPE>'

example: WELCOME TO THE TUESDAY EDITION OF THE HAWAII JS8 NET. GE THIS IS LB UR HOST. THIS IS A ONE ROUND ROUND TABLE NET'

Check-In Request

specific format: WH6NCS: @HINET ANY CHECK-INS?

note it is not necessary to decode all check in requests. one is more than sufficient and not even necessary.

Check-In

general format1: ... <SWL>...

general format2: ... <CHECK>...

general format3: ... <CK-IN>...

general format4: ... <CK IN>...

general format5: ... <CKIN>...

general format6: @HINET...

example: WH6ABC: @HINET CHECK ME IN PLS?

example: WH6ABC: @HINET CHECK ME IN SWL ONLY PLS?

QSY

specific format: WH6NCS: @HINET QSY TO <FREQ>

example: WH6NCS: @HINET QSY TO 7.078

please note that stations wishing to automatically adjust the freq based on this message must use the command line option '--update_freq_on_qsy'

Offsets Plan

process a new offsets plan sent by net control

specific format: WH6ABC: @HINET ... OFFSETS PLAN IS: <COMMA SEPARATED LIST>

example: WH6ABC: @HINET OK THE NEW OFFSETS PLAN IS: 1337,500,600,700,800,900

First on Roster

specific format 1: WH6NCS @HINET ... FIRST ON THE ROSTER IS <CALLSIGN>. ...

example: WH6NCS: @HINET OK LETS GET STARTED. FIRST ON THE ROSTER IS WH6ABC. GE JIM START US OUT PLS.

specific format 2: WH6NCS @HINET ... FIRST ON THE LIST IS <CALLSIGN>. ...

example: WH6NCS: @HINET OK LETS START AT THE TOP. FIRST ON THE LIST IS WH6ABC. GE FRED YOUR REPORT PLS.

general format 3: ... START WITH <CALLSIGN> ...

example: WH6NCS @HINET OK LETS GET STARTED. WE START WITH WH6ABC.

example: WH6NCS @HINET OK LETS START WITH WH6ABC.

general format 4: ... BEGIN WITH <CALLSIGN> ...

example: WH6NCS @HINET OK LETS GET STARTED. WE BEGIN WITH WH6ABC.

example: WH6NCS @HINET OK LETS BEGIN WITH WH6ABC.

general format 5: ... <CALLSIGN> IS FIRST ...

example: WH6NCS @HINET OK LETS GET STARTED. WH6ABC IS FIRST ON THE LIST. GE TOM YOUR REPORT PLS.

please note <CALLSIGN> is verified to make sure it is a valid participant of the net and that net control is making the message

End Of Round

general format 1: WH6NCS: @HINET ... THE END OF ROUND ...

example: WH6NCS: @HINET OK WE ARE AT THE END OF ROUND 1

example: WH6NCS: @HINET OK WE HAVE REACHED THE END OF ROUND 1

general format 2: WH6NCS: @HINET ... THE END OF THIS ROUND ...

example: WH6NCS: @HINET OK WE ARE AT THE END OF THIS ROUND

general format 3: WH6NCS: @HINET ... THE END OF THE ROUND ...

example: WH6NCS: @HINET OK WE ARE AT THE END OF THE ROUND

example: WH6NCS: @HINET OK WE HAVE REACHED THE END OF THE ROUND

Over To

general format 1: WH6NCS: @HINET ... OVER TO <CALLSIGN>. ...

example: WH6NCS: @HINET NEXT OVER TO WH6ABC. GE FRED TAKE A TURN PLS

please note message must have a period symbol at the end of the call sign.

QRT

general format: WH6ABC: @HINET ... GOING QRT...

example: WH6ABC: @HINET THANKS FOR THE NET IM GOING QRT. ALOHA!

Back To Net

general format 1: WH6ABC: @HINET ... BACK TO NET

general format 2: WH6ABC: @HINET ... BTN

general format 3: WH6ABC: @HINET ... BTU

example: WH6ABC: @HINET THANKS FOR DOING THE NET BTU

Command Line Options

--interface= , --i=

default = Net Control

This option will run the program in one of its two modes net control or participant:

examples:

Participant Station

```
$ python ./js8_net_client.py --i=participant
```

Net Control Station

```
$ python ./js8_net_client.py --interface=netcontrol
```

Please note that participant stations should not use the net control option if they plan to participate as a participant. Net control option is for net control only.

--net_file= , --n=

default = js8net_save_data.txt

```
$ python ./js8_net_client.py –interface=netcontrol --net_file='js8net_save_data.txt'
```

This option allows the user to specify alternate data files for the application. The js8net_save_data.txt is the default file and should be used as the starting point. If you wish to create you own alternative files, first copy this file to your new data file then start the program pointing to the new data file.

--js8call= , --j=

default = 127.0.0.1:2442

```
$ python ./js8_net_client.py –interface=netcontrol --js8call='127.0.0.1:2442'
```

This option allows you to specify a different ip and or port number for js8 call. These must point to the computer ip that has the JS8Call application running and the port that is used for the TCP connection.

--group= , --g=

```
$ python ./js8_net_client.py –interface=participant --group=’@HINET’
```

This option is primarily for participant station and allows the group field to be pre-specified from the command line. This is prefereable if you know the group that will be used for the net that you plan to participate in.

--frequency= , --f=

```
$ python ./js8_net_client.py –interface=participant --frequency='fromjs8call'
```

This option allows the application to use the same frequency that is specified in JS8Call. When the JS8Call frequency is changed then the frequency field in the JS8Net application will also change.

```
$ python ./js8_net_client.py –interface=participant –frequency='7.080'
```

This option will set the frequency field in the application to the frequency specified in MHz.

--main_offset= , --m=

default 1000

```
$ python ./js8_net_client.py –interface=participant --main_offset=1000
```

This sets the offset used for transmitting messages to the value specified

```
$ python ./js8_net_client.py –interface=participant --main_offset=’from_file’
```

This set the value of the offset field to the value save in the save file

```
$ python ./js8_net_client.py –interface=participant --main_offset=’from_plan’
```

This sets the value of the offset field to the value from the offsets plan

--boundary= , --b=

default 700

```
$ python ./js8_net_client.py –interface=netcontrol --boundary=800
```

This option sets the bondary offset between the main offset and the sidebar offset. Messages using an offset above this number will appear in the main window. Message using an offset below this value will appear in the sidebar window.

--client_read_details , --r

default = False

```
$ python ./js8_net_client.py –interface=participant --client_read_details
```

This option is ofr participant stations to read the details of the main window from the save data file.

--profile= , --p=

Default = day of week

```
$ python ./js8_net_client.py –interface=netcontrol --profile='Tango'
```

This will pverride the value of the 'Edition' field on the main page.

--counter= , --c=

default = 200 and visible

```
$ python ./js8_net_client.py –interface=netcontrol --counter='off'
```

This will disable and hide the countdown timer on the main page.

```
$ python ./js8_net_client.py –interface=netcontrol --counter='300'
```

This will set the 'countdown' / count-up timer to count to 300

--delay= , --d=

default = 25

```
$ python ./js8_net_client.py –interface=netcontrol --net_file=’js8net_save_data.txt’
```

This option changes the delay when using the post + send option for true one-click operation. The delay is to give you time to review the message or to click on another message format prior to the message being sent. Each time you click on a new format, the delay timer starts again. Once it reaches the value specified, send activates automatically when the post + send option is used.

--visual= , --v=

default =
'background:LightGray,main:SeaGreen1,side:LightBlue1,flash1:red,flash2:blue'

```
$ python ./js8_net_client.py –interface=netcontrol --
visual='background:LightGray,main:SeaGreen1,side:LightBlue1,flash1:red,flash2:blue'
```

This option allws the screen colors to be modified.

Advanced Features

Distributing an Offset Plan

--offsets= , --o=

default = '1337,700,870,1140,1210,750,920,1190,1260,800,970,1240'

```
$ python ./js8_net_client.py –interface=netcontrol --offsets='1337,400,500,600,700'
```

A new offsets plan can be specified on the command line as in the above example. Typically the offsets plan is sent to the sations after the roster is sent out to them. The message to send the offsets plan can be found on the ‘Extras’ button

Modifying Pre-defined Messages

--combo_tks= , --t=

default ='Report,Good Report,Great Report,Signal Report,Great Question,Good Idea,Good Comment'

```
$ python ./js8_net_client.py –interface=netcontrol --combo_tks='comments,heads
up,info'
```

This option allows the customization of the combo drop down linked to the ‘Tks’ checkbox which is used by net control. The options in the list are provided as a comma separated list as in the above example. These values are used by the Macro language. The current value of this field can be accessed from the macro language by using the %TKSMSG macro.

The default macros provided with the application (Next and End macros) use this value in the following context:

'%IFTKS Thanks %LN for the %TKSMSG. %ENDIFTKS '

If you wish to change to values of the combo dropdown as well as the contxt then you will also need to edit the pre-defined macros so that the combination of the edited macro + the combo text makes grammatical sense. This can be done using the –edit feature. Please also refer to previous section about Dynamic Content Macros.

--combo_aloha= , --a=

default ='Great Evening,Good Evening,Great Rest of the Day,Good morning'

```
$ python ./js8_net_client.py –interface=netcontrol --combo_aloha='great morning,nice
day,great rest of the day'
```

This option allows the customization of the combo drop down linked to the ‘good evening’ checkbox on the far right of the screen as used by net control. The options in the list are provided as a comma separated list as in the above example. These values are used by the Macro language. The current value of this field can be accessed from the macro language by using the %GOODEVE macro.

The default macros provided with the application (Next and End macros) use this value in the following context:

%IFGOODEVE have a %GOODEVE %LN. %ENDIFGOODEVE '

If you wish to change to values of the combo dropdown as well as the contxt then you will also need to edit the pre-defined macros so that the combination of the edited macro + the combo text makes grammatical sense. This can be done using the –edit feature. Please also refer to previous section about Dynamic Content Macros.

Editing the Built-In Macros

--edit , --e

```
$ python ./js8_net_client.py –interface=netcontrol --edit
```

This will put the application into edit mode so that the macros can be edited. Only macros relating to the current view are able to be edited and these are different for netcontrol and participant.

WARNING WARNING WARNING

By editing the built in messages, it is easy to break the station synchrnization feture that relies on parsing of specific phrases. If you do override the built in message with your own, be sure to read the section on ‘Pre-Coded phrases’ as you will need to make sure that your messages contain one of these phrases as required. You can also test your edited message using the simulate mode descibed below.

Please also note there is no validation of edited messages that contain Dynamic Content Macros. Please be sure to review your changes very thoroughly as it could break the outbound message parser that handles the Dynamic Content Macros.

WARNING WARNING WARNING

Simulation Mode

--simulate , --s

```
$ python ./js8_net_client.py –interface=netcontrol --simulate
```

This is an advanced option and is used only for testing of new messages. It allows a simulated messaged to be sent to the application as if it had been received from another station. The command line option will show a 'simulate' checkbox on the main windows. The checkbox must be clicked to activate simulate mode. Do not use this option unless you have mastered all other aspects of the application.

Updating Frequency from a QSY Message

--update_freq_on_qsy , --u

default = False

```
$ python ./js8_net_client.py –interface=netcontrol --update_freq_on_qsy
```

This option will automatically update the value of the frequency field upon receipt of a QSY message from net control. The recommendation is to keep this option set to off unless your station automatically activates re-tuning when the frequency is changed.

Other Examples

Running the application as net control and specifying data file for saving application data:

```
$ python ./js8_net_client.py --interface=netcontrol --net_file=js8net_save_data.txt
```

Running the application as net control with customized colors:

```
$ python ./js8_net_client.py --interface=netcontrol --net_file=js8net_save_data.txt
--visual='background:yellow,main:green,side:blue,flash1:purple,flash2:white'
```

Running the application as net client or participant. In this example the program reads the data from the data file, acquires the frequency to be used from JS8Call and overrides the group name from the command line parameter.

```
$ python ./js8_net_client.py --interface=participant --net_file=js8net_save_data.txt
--simulate --client_read_details --frequency='fromjs8call' --group='@HINET'
```

Running the application as net client or participant. In this example the program uses the simulate mode to allow testing of modified edit data

```
$ python ./js8_net_client.py --interface=participant --net_file=js8net_save_data.txt
--simulate --frequency='fromjs8call' --group='@HINET'
```