

# JS8Call de KN4CRD

## Précédemment connu sous le nom de FT8Call

2019-04-05 – v1.0.0 – General Availability (traduction approx. F4HXE)

---

Le FT8 est sur les ondes un mode de communication numérique pour réaliser des QSO sur HF / VHF / UHF. Ce mode a été largement utilisé dans l'application WSJT-X de K1JT dont il est la dernière offre. Le FT8 repose sur les épaules des modes JT65, JT9 et WSPR pour une communication de signal faible, mais transmet beaucoup plus rapidement avec une sensibilité légèrement réduite.

Alors que FT8 est un mode de signal faible incroyablement robuste, il est conçu pour tirer parti des ouvertures de bande courtes sur HF / VHF / UHF et offre seulement un cadre QSO minimal. Cependant, de nombreux opérateurs utilisent ces faibles qualités de signal pour réussir des QSO sur les bandes HF où d'autres modes échouent.

L'idée avec JS8Call est de prendre la robustesse du mode FT8 et la couche de messagerie et le protocole de réseau pour une communication de signal faible sur HF avec une interface clavier à clavier. JS8Call est fortement inspiré par WSJT-X, Fldigi et FSQCall et n'existerait pas sans le travail et le dévouement des nombreux développeurs de la communauté des radioamateurs.

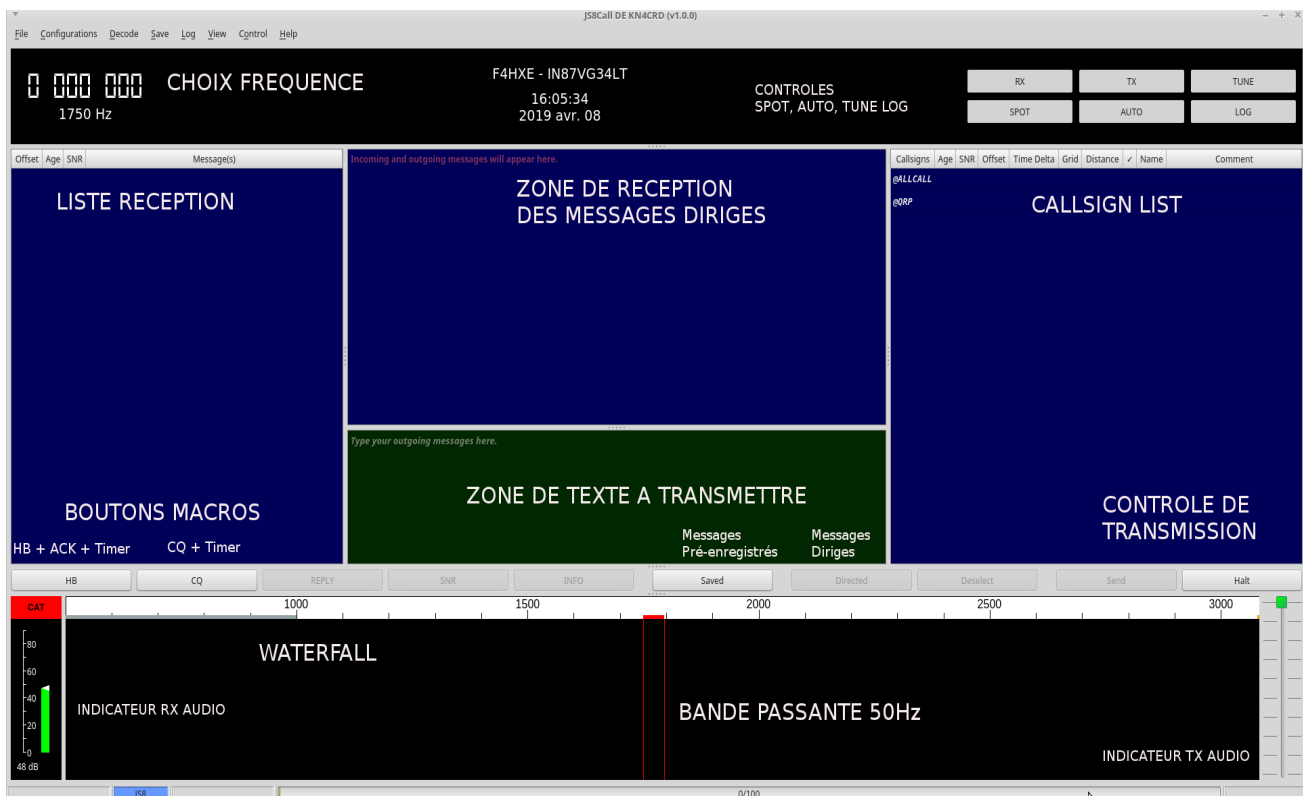
*JS8Call stands on the shoulder of giants...the takeoff angle is better up there.*

Read more on the original [design inspiration here](#).

Pour les annonces de sortie de nouvelles versions et des discussions, rejoignez la liste de diffusion :  
<https://groups.io/g/js8Call>

## History

- **July 6, 2017** - The initial idea of using a modification to the FT8 protocol to support long-form QSOs was developed by Jordan, KN4CRD, and submitted to the WSJT-X mailing list: <https://sourceforge.net/p/wsjt/mailman/message/35931540/>
- **August 31, 2017** - Jordan, KN4CRD, did a little development and modified WSJT-X to support long-form QSOs using the existing FT8 protocol: <https://sourceforge.net/p/wsjt/mailman/message/36020051/> He sent a video example to the WSJT-X group: <https://widedido.wistia.com/medias/7bb1uq62ga>
- **January 8, 2018** - Jordan, KN4CRD, started working on the design of a long-form QSO application built on top of FT8 with a redesigned interface.
- **February 9, 2018** - Jordan, KN4CRD, submitted question to the WSJT-X group to see if there was any interest in pursuing the idea: <https://sourceforge.net/p/wsjt/mailman/message/36221549/>
- **February 10, 2018** - Jordan KN4CRD, Julian OH8STN, John N0JDS, and the Portable Digital QRP group did an experiment using FSQ. The idea of FT8Call, combining FT8, long-form QSOs, and FSQCall like features was born.
- **February 11, 2018** - Jordan, KN4CRD, inquired about the idea of integrating long-form messages into WSJT-X: <https://sourceforge.net/p/wsjt/mailman/message/36223372/>
- **February 12, 2018** - Joe Taylor, K1JT, wrote back: <https://sourceforge.net/p/wsjt/mailman/message/36224507/> saying no and “Please don't let my comment discourage you from proceeding as you wish, toward something new.”
- **March 4, 2018** - Jordan, KN4CRD, published a design document for FT8Call: <https://github.com/jsherer/ft8call>
- **July 6, 2018** - Version 0.0.1 of FT8Call released to the development group
- **July 15, 2018** - Version 0.1 released - a dozen testers
- **July 21, 2018** - Version 0.2 released - 75 testers
- **July 27, 2018** - Version 0.3 released - 150 testers
- **August 12, 2018** - Version 0.4 released - (“leaked” on QRZ) - 500 testers
- **September 2, 2018** - Version 0.5 released - 3000 testers
- **September 14, 2018** - Version 0.6 released - 5000 testers
- **October 8, 2018** - Version 0.7 released - 6000 testers, name changed to JS8 & JS8Call
- **October 31, 2018** - Version 0.8 released - ~7000 testers
- **November 15, 2018** - Version 0.9 released - ~7500 testers
- **November 30, 2018** - Version 0.10 released - ~7800 testers
- **December 18, 2018** - Version 0.11 released - ~8200 testers
- **January 1, 2019** - Version 0.12 released - ~9000 testers
- **January 23, 2019** - Version 0.13 released - ~9250 testers
- **February 7, 2019** - Version 0.14 released - ~9600 testers
- **February 21, 2019** - Version 1.0.0-RC1 released - ~10000 testers
- **March 11, 2019** - Version 1.0.0-RC2 released - >10000 testers
- **March 26, 2019** - Version 1.0.0-RC3 released - >11000 testers
- **April 1, 2019** - Version 1.0.0 general availability - Public Release!



## Notice

JS8Call est un dérivé de l'application WSJT-X, restructuré et repensé pour le passage de messages clavier à clavier, utilisant une version adaptée de la modulation FT8 appelée JS8. Il n'est pas soutenu ni approuvé par le groupe de développement WSJT-X. Bien que le groupe WSJT-X conserve les droits d'auteur sur l'œuvre et le code originaux, JS8Call est une œuvre dérivée concédée sous la [licence GPLv3](https://www.gnu.org/licenses/gpl-3.0.html). Le code source peut être trouvé dans ce dépôt public : <https://bitbucket.org/widefido/js8call/>

JS8Call est et restera open-source et gratuit.

Vous vous demandez peut-être... Pourquoi ce nom de JS8Call ? Pourquoi renommer le FT8Call ?

Pourquoi pas quelque chose d'autre comme BACON ou HF Messenger ? Bonne question ! Il fut renommé ainsi en hommage à son héritage :

You might be asking...why is this named JS8Call? Why was it renamed from FT8Call?

Why not something else, like BACON or HF Messenger? Good question! It is named this way as an homage to its heritage:

- JS8Call était précédemment appelé FT8Call.
- JS8Call utilise une modulation FT8 personnalisée appelée JS8 (Jordan Sherer designed 8-FSK modulation). C'est la base du transport RF.
- JS8Call a un protocole de « communication directe » lié à la couche de transport RF afin de supporter l'envoi de texte libre et des commandes dirigées.

JS8 + Directed Calling = JS8Call. And in case you didn't get that :

- The mode is: **JS8**
  
- The app is: **JS8Call**

## Téléchargement & Installation

JS8Call vient actuellement dans une variété de versions :

- Desktop Linux (64-bit x86\_64, deb)
- Desktop Linux (32-bit i386, deb)
- Raspbian Stretch (armv7, deb)
- Windows 10 (win32\_64)
  - Windows 10 is the only officially supported Windows build at this time, but the application has been confirmed to work all the way back to Windows XP.
- Mac OSX 10.11+ (x86\_64)

Pour les liens de téléchargement les plus récents, rejoignez le

<https://groups.io/g/js8call/join> et vérifiez :

- [JS8Call Release Announcements](#)
- [JS8Call Release Download Links](#)

Bien sûr, vous êtes toujours libre de regarder le [code source](#) également !

NOTE: Les versions de développement du JS8Call ont une validité d'une quinzaine de jours. Après expiration, vous devrez mettre à jour votre version de l'application. Ceci aide les opérateurs à utiliser des versions compatibles pendant le développement du logiciel et son beta test. Cette expiration n'existe pas dans les versions générales du logiciel.

## Débuter avec JS8Call

### Synchronisation d'horloge / Clock Sync / Timing

Dans l'application vous pouvez voir l'heure au format UTC. Une horloge précise est importante avec JS8Call, puisque le décodeur fonctionne sur un intervalle de temps de 15 secondes de transmission (frames). Si votre horloge accuse une différence de plus de 2 secondes par rapport au temps UTC le décodage des messages risque de ne pas s'opérer à votre station (bien que cela puisse fonctionner jusqu'à un décalage de 2400ms NDT). Il est recommandé d'utiliser une source Internet, NTP ou GPS pour synchroniser votre horloge aussi précisément que possible.

JS8Call inclut un outil manuel de compensation du décalage de l'horloge grâce auquel vous pourrez modifier le temps dans l'application pour le caler sur les trames de transmission que vous observerez ou entendrez (ou sur une source de temps externe telle qu'une montre, un GPS portable, WWV, or the rooster crowing ou le « chant du coq » (**VERIF**). Ceci dans le but de permettre vos communications lorsque vos outils distants de

synchronisation de temps ne sont pas disponibles (activités portables, zone blanche hors réseau ou sans réception GPS).

NOTE: Actuellement vous n'avez pas à avoir une synchronisation parfaite... Calez vous sur le début d'une trame, +/- 2 seconds. La plupart des opérateurs peuvent synchroniser leur horloge en se basant sur la réception des signaux dans le waterfall et avec la différence de temps envers les autres stations (view > show time drift controls).

## **USB - Upper Sideband Seulement**

Soyez sûrs que votre RX/TX est réglé en mode upper sideband (USB), et ceci pour chaque bandes.. Si vous êtes en lower sideband (LSB), vous recevrez des signaux que vous ne pourrez pas décoder.

## **Duty Cycle**

Le modulateur JS8 est une enveloppe constante, modulation full-duty qui transmet en trames de 12.6 second. En raison de « zone morte » entre les trames de transmission, les messages multi-trames ont un rendement de 84 % sur une fenêtre de 15 secondes ( $12.6 / 15 = 0.84$ ).

S'il vous plait, reportez vous aux restrictions de puissances recommandées par le constructeur de votre transceiver pour les transmissions en digimodes. En cas de doute, utilisez seulement 50 % de la pleine puissance en sortie.

## **Audio Levels**

Vos niveaux d'entrée et sortie audio contrôlent la qualité de vos transmissions et réceptions. Trop hauts et l'audio devient distordue. Trop bas et vous n'avez pas de modulation / demodulation. La calibration est une étape important pour bien démarrer.

## **Output & ALC (VERIF)**

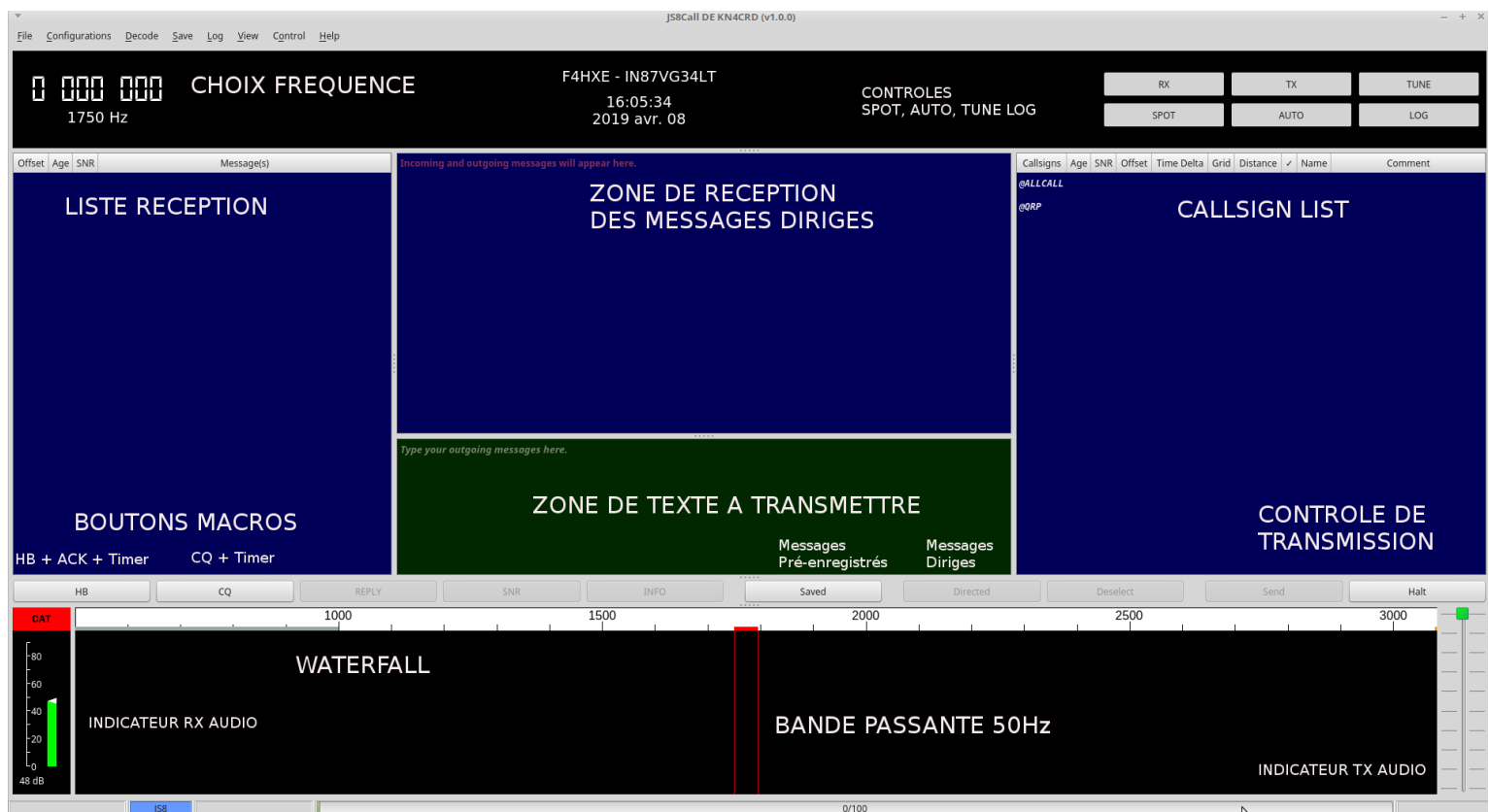
Une règle simple est de régler votre sortie audio juste assez pour commander votre transmetteur juste pour engager votre ALC. Si vous paramétrez votre audio trop fort, votre ALC distordra vos tonalités et beaucoup de stations ne pourront pas décoder vos transmissions.

## **Input & AGC**

Pour de meilleurs décodages, le mieux est d'éteindre votre AGC (ou de la mettre sur rapide) et paramétrer votre entrée audio suffisamment pour lire c'est-à-dire environ 30-40dB sur le S-mètre de l'application. Vous aurez certainement à expérimenter afin de trouver ce qui fonctionne le mieux pour votre station, et vous devrez probablement engager votre atténuateur sur les signaux forts.

## **Trafiquer avec JS8Call**

Si vous avez déjà utilisé Fldigi ou WSJT-X, vous vous sentirez à l'aise avec JS8Call. La prémisse est que JS8Call modifie la structure de codage des messages JS8 modulés de style FT8, en décomposant les messages de long terme en plusieurs trames / cycles de transmission de 15 secondes, dans lesquelles 12,6 secondes sont effectives.



Voici ce que vous verrez lorsque vous démarrez l'application :

## Liste de réception / Band Activity

L'activité de la bande (Band Activity) est affichée à gauche. Les indicatifs que vous avez entendus (Call activity) sont à droite. Avec un clic-droit sur l'un de ceux-ci, votre décalage RX / TX sera déplacé vers cette fréquence audio (QSY) ou afficher différentes commandes possibles, ou encore la liste des commandes dirigées (commandes automatique).

## Callsign list

Dans le Call Activity, quand une station vous répond, l'indicateur ★ s'affiche à côté de l'indicatif. Ceci vous aide à trouver les opérateurs capable de vous entendre de façon confirmée.

Lorsqu'une station émet un appel (CQ), l'indicateur 📞 s'affiche à côté de l'indicatif pendant 5 minutes. Ceci vous aide à trouver les opérateurs recherchant des contacts

Si une station vous a laissé un message, l'indicateur 📄 s'affiche à côté de l'indicatif. Vous pouvez lire ce message avec un clic-droit sur la station et en choisissant "Show Message Inbox".

Les distance et azimuth avec les stations sont codées depuis les 6 premiers chiffres des maidenhead grid locators. Ceci suffit à une résolution de 2.8 mi / 4.6 km. Plus proche que cela et vous lirez distance 0 et azimuth 0°.

## Waterfall

Il y a un waterfall en bas de l'écran pour voir les signaux dans votre bande passante. Vous pouvez cliquer sur ce waterfall et choisir votre fréquence.

Il existe également une option pour QSY à cette fréquence en centrant votre offset audio sélectionné sur le center de la bande passante de votre RX/TX. L'emploi de filtres étroits s'en trouve facilitée.

## Messages

La zone de texte jaune pâle au milieu en haut affiche les messages correspondant au décalage de fréquence sur lequel vous êtes ou ceux qui vous ont été envoyés ( ce sont des messages contenant votre indicatif).

Vous écrivez dans la case blanche juste en dessous pour préparer un message à transmettre. Les restrictions de caractère FT8 normales ne s'appliquent pas ! Le jeu de caractères étendu inclut tous les caractères ASCII majuscules imprimables. (A-Z 0-9 Space ./?+-`~!@#\$\$%^&\*()\_=[\{}|;':",<>).

La structure du message est codée de manière variable, de sorte que les caractères les plus courants prennent le moins de place et que l'envoi des caractères spéciaux prend plus de temps.

*(ndT : certains groupe de caractères peuvent être automatisés afin d'économiser des bits de transmission et réduire les trames, voir plus bas Appendice A , table de code d'Huffman)*

Si vous tapez votre message, le bouton d'envoi affichera le nombre de trames (15 cycles de transmission) qu'il vous faudra pour envoyer votre message complet. Tout ce que vous avez à faire est de cliquer sur Envoyer/Send (ou sur Entrée) pour commencer à transmettre à l'intervalle suivant. Comme chaque image est transmise l'une après l'autre, le bouton se met à jour avec le nombre de trames restant.

En raison de cet encodage de variables spécial, les messages en JS8Call ne peuvent pas être décodés par WSJT-X. Ceci est réciproque, les messages WSJT-X ne seront pas affichés dans JS8Call.

Les messages sont de 3 types :

1. messages JS8Call standard, en texte libre
2. messages JS8Call dirigés (directed)

### 3. messages JS8Call non dirigés (undirected)

#### Messages standards

Les messages standards sont des messages libres qui ne commencent pas par un indicatif ou une commande dirigée. Ces messages ne s'imprimeront chez une autre station que si la fréquence de réception de cette dernière s'aligne avec un décalage de 10 Hz par rapport à votre fréquence de transmission. Cette opération est similaire aux autres modes numériques clavier-clavier, tels qu'Olivia, RTTY et PSK.

#### Messages dirigés

Les messages dirigés sont des transmissions JS8Call spéciales qui préfixent automatiquement votre message avec votre indicatif, de la même manière que FSQCall. Les messages dirigés sont utiles pour communiquer en ce sens que vous n'avez pas à inclure votre indicatif dans votre message, ce qui vous permet d'utiliser davantage la ou les trames de transmission pour le texte du message, et d'avertir le destinataire qu'un message lui a été envoyé. Tant que vous êtes dans la même bande passante, vous n'avez pas besoin d'être sur le même décalage de fréquence pour recevoir un message dirigé.

Pour envoyer un message dirigé, il vous suffit d'inclure l'indicatif de la station réceptrice comme premier mot du message ou de sélectionner l'indicatif dans la liste et il sera automatiquement préfixé.

Vous noterez un caractère spécial à la fin des message : ala ↵. Il est nommé [Electric Arrow](#), et est un symbole pour indiquer la fin d'une transmission. JS8Call l'affiche après la dernière trame du message lorsque rien ne suit. Cela signifie que vous avez un indicateur visuel d'une transmission terminée et vous pouvez dès lors répondre. C'est aussi le caractère de base de l'icone JS8Call.

Les messages dirigés qui vous sont adressés (ainsi qu'à @ALLCALL) dans la fenêtre de réception (au milieu en haut).

Quand en milieu de réception d'un message dirigé (ex : après la première trame d'un message en comportant plusieurs), votre station ne répondra pas automatiquement à la commande (même avec AUTO enclenché) tant que le message ne sera pas réceptionné en entier ou que suffisamment de temps se soit écoulé (une minute après la réception de la dernière trame – à vérifier NDT)

#### Messages non-dirigés

Les messages non-dirigés sont des transmissions en JS8Call spécialement formatées qui annoncent votre station via CQ or Heartbeats (HB). Elles sont non dirigées puisque sans destination.



## @ALLCALL Callsign

Il existe un indicatif spécial "ALLCALL" que vous pouvez utiliser pour envoyer le message à toute personne capable de recevoir votre message. Quelques exemples:

- @ALLCALL HELLO NET PSE QSY 14300
  - Will be sent as: **KN4CRD: @ALLCALL HELLO NET PSE QSY 14300 ↗**

## Group Callsigns / Indicateurs de groupe

Les Groupes sont un composant spécial des indicateurs et commencent avec un caractère '@' pouvant être suivi de 8 caractères alpha-numériques (A-Z 0-9). (ex @GROUP1,@QRP...)

If you modeled that in a regular expression, that would be:

```
[@][A-Z0-9]{0,3}[V]?[A-Z0-9]{0,3}[V]?[A-Z0-9]{0,3}
```

La fonctionnalité des Groupes d'indicateurs vous permet de diriger vos messages vers tous ceux qui ont rejoint ce Groupe. You join the group by adding the group name in the settings.

Par exemple, je fais partie du Groupe ARES GA et souhaite envoyer à ceux de ce groupe un message :

KN4CRD: @ARESGA QSL?

C'est similaire à @ALLCALL, puisque tous ceux qui composent ce groupe @ARESGA auront ce message sur leur écran.

Il y a un certain nombre de groupe préconstruits dans le formatage du JS8Call, lesquels se transmettent aussi efficacement qu'un indicatif standard :

- Any net can use these net groups:
  - @JS8NET
  - @NET
- Continental DX Groups
  - @DX/NA
  - @DX/SA
  - @DX/EU
  - @DX/AS
  - @DX/AF
  - @DX/OC
  - @DX/AN
- ITU Regions
  - @REGION/1
  - @REGION/2
  - @REGION/3
- Generic Groups

- @GROUP/0
- @GROUP/1
- @GROUP/2
- @GROUP/3
- @GROUP/4
- @GROUP/5
- @GROUP/6
- @GROUP/7
- @GROUP/8
- @GROUP/9
- Operator Groups
  - @COMMAND
  - @CONTROL
  - @NTS

## Commandes dirigées :

Il existe des messages dirigés spéciaux que vous pouvez envoyer et auxquels les stations répondront automatiquement si AUTO est activé. Leur forme est de type [CALLSIGN] [COMMAND]

- SNR? - Quel est mon report ?
- GRID? - Quel est votre grid locator?
- INFO? - Quelles sont les informations de votre station ?
- STATUS? - Quel est le statut de votre station ?(activité, auto, hb, spot, version, etc)
- HEARING? - Quelles stations entendez vous ?
- >[MESSAGE] – Pouvez vous relayer ce message à sa destination ?
  - S'il est reçu en entier, la station concernée enverra un ACK (Acknowledge) en réponse à ce message.
  - Optionnellement ce message peut être relayé à travers de multiples stations relais, en préfixant les indicatifs additionnelles au message:
    - KN4CRD>HELLO!  
(Enverra le message à KN4CRD qui renverra un ACK)
    - KN4CRD>DR4CNK>HELLO!  
(Enverra un message à DR4CNK relayé par KN4CRD)
    - KN4CRD>DR4CNK>J0Y>HELLO!  
(Enverra le message à J0Y relayé par DR4CNK relayé déjà par KN4CRD)
  - Les stations répondront également à une série de messages automatisés (SNR, INFO, GRID, MSG, MSG TO:, etc) et répondrons en suivant le chemin donné.
- MSG [MESSAGE] – S'il vous plait stockez ce message et affichez-le dans votre inbox ('boite de réception')
  - Vous accéderez à votre message avec un clic-droit sur l'indicatif puis en sélectionnant "View Message Inbox..."
- MSG TO:[CALLSIGN] [MESSAGE] – S'il vous plait, stockez ce message à votre station pour le délivrer plus tard, sur demande, à l'indicatif[CALLSIGN] spécifié.
  - Stocke le message de façon persistante (sur le disque) pour être récupéré plus tard en utilisant la commande "QUERY MSGS".

- [CALLSIGN] est le premier mot après les : et il s'agit du destinataire final du message.
- *Le message est retransmis textuellement par la station réceptrice avec l'ajout de "DE [CALLSIGN]" ajouté à la fin du message ... ce qui signifie que vous n'avez pas besoin de l'ajouter à votre message.*
- QUERY CALL [CALLSIGN]? - Pouvez vous communiquer directement avec l'indicatif ?
  - Si une station entend l'indicatif, elle répondra "YES" et ensuite un ACK relatif a la station demandée avec un report (SNR)
- QUERY MSG [ID] – Pouvez vous délivrez le message ID (numéro du message)
- QUERY MSGS – S'il vous plait délivrez les message que vous avez stocké pour ma station
  - Chaque réponse contiendra les ID des derniers messages à délivrer
  - Pour récupérer le(s) texte(s) d(u)(es) message(s) lancez la commande QUERY MSG avec l'[ID]
- AGN? - S'il vous plait pouvez vous renvoyer votre dernière transmission ?
- ---
- SNR – Envoyer un report
- INFO – Envoyer les informations de sa station.
- GRID – Envoyer son grid locator au format maidenhead afin d'apparaître sur une carte, via PSKReporter & APRS-IS

Il y a aussi un certain nombre de messages courts qui peuvent être inclus dans la trame d'un message dirigé, et qui seront transmis en un seul cycle vers des indicatifs standards (non-composés, ne faisant pas partie d'un @Group):

- QSL? - Avez-vous reçu ma dernière transmission ?
- QSL – J'ai reçu votre dernière transmission
- YES – Je confirme votre dernière requête
- NO – Je ne confirme pas votre dernière requête
- HW CPY? - How do you copy ? Comment me recevez-vous ?
- RR - Roger. Received. I copy.
- FB - Fine Business
- TU - Thank You
- 73 - I send my Best Regards
- SK - End of Contact
- DIT DIT - End of Contact / Two Bits

### Exemple de Message :

Si nous voulions demander à DR4CNK, quelles sont les informations de votre station, nous enverrions :

- DR4CNK INFO ?
  - Et la station nous répondra avec un message dirigé : "DR4CNK: KN4CRD INFO 50W VERT IN THE SOUTH OF FRANCE", automatiquement si AUTO reply est activé.

Si nous voulions transmettre un message « relais » à OH8STN par l'intermédiaire de DR4CNK, nous utiliserions la commande relais et enverrions :

- DR4CNK>OH8STN>HELLO JULIAN!
  - Pendant le relais à chaque étape, l'indicatif de l'expéditeur original est ajouté au message.
  - La station intermédiaire répondra par un ACK back, puis retransmettra le message comme suit :
    - KN4 station sends:  
KN4CRD: DR4CNK>OH8STN>HELLO JULIAN!
    - DR4 station relays:  
DR4CNK: OH8STN>HELLO JULIAN! DE KN4CRD

Vous pouvez aussi mélanger message standards et texte libre, mais la plupart du temps vous n'en aurez pas besoin.

## **Inbox - Boite de réception – Stockage des messages et récupération des messages**

L'envoi de messages dirigés comprennent l'utilisation de 3 commandes pour stocker les messages et les récupérer sur une station intermédiaire :

- MSG TO:[CALLSIGN] [MESSAGE] – Stocker un message chez une station intermédiaire.
- QUERY MSGS – Questionner la destination pour avoir les messages stockés à destination de votre indicatif.
- QUERY MSG [ID] – Récupérer un message spécifique [ID] à destination de votre station.

## **AUTO - Automatic Replies**

Lorsque AUTO est enclenché, le logiciel répond automatiquement aux commandes dirigées, comme "SNR?", "INFO?", et "GRID?". Quand AUTO n'est pas sélectionné, JS8Call met en cache les réponses aux requêtes des commandes dirigées dans la zone d'envoi des messages (milieu en bas), jusqu'à ce que vous puissiez envoyer ces réponses manuellement.

Si vous voulez engager le mode AUTO mais ne souhaitez pas participer aux commandes de relais, il vous suffit de cocher la case 'disable relays while AUTO is enabled' dans File>Settings>General>Behavior>Autoreply

## **LOG - Station Log**

Il y a un LOG directement accessible depuis l'interface. Vous pouvez aussi presser F5 pour lancer entrer un nouveau contact. Le logiciel fera de son mieux pour préremplir les champs. Toutefois vous aurez à remplir manuellement les informations manquantes car le QSO est libre et non automatisé.

Le LOG est consultable dans les fichiers JS8Call.log & JS8Call.adif dans log directory (Vous pouvez y accéder en cliquant "Log -> Open log directory" depuis la fenêtre principale).

Communément la fonction de logging du JS8Call, enregistrera chaque contact selon les spécifications ADIF en tant que MFSK mode et JS8 submode. Il existe un fonction dans les paramètres de Logging pour utiliser DATA à la place de MFSK et JS8.

Une fois le contact enregistré, l'indicatif du correspondant est automatiquement désélectionné par défaut, ceci pouvant être modifié dans la configuration.

## SPOT - Callsign Spotting

Quand SPOT est sélectionné, JS8Call reportera les indicatifs entendus (ou votre indicatif s'il est entendu par d'autres station) sur PSKReporter mode JS8.

JS8Call reportera également la commande GRID comportant 6 caractères ou plus sur APRS-IS / [aprs.fi](http://aprs.fi). Vérifiez que vous avez entré votre locator avec 6-12 caractères pour plus de précision. Si vous n'êtes pas sûr de votre localisation, cette carte peut vous aider : <http://k7fry.com/grid/>. Si vous possédez la latitude et la longitude vous pouvez utiliser le script lonlat2maiden script ici : <http://www.jidanni.org/geo/maidenhead/>

## HB - Heartbeat Transmission

Un mécanisme automatique (Heartbeat / pulsation/ 'balise') transmet à intervalles réguliers une fois activé ( bouton HB à gauche barre de menu du milieu). Cet intervalle peut être changé avec un clic-droit sur le bouton. A chaque fois que vous transmettez, le compte à rebours est réinitialisé. Chaque Heartbeat est transmise sur une plage de fréquence inoccupée aléatoire entre 500Hz et 1000Hz afin d'éviter du QRM. Une option dans les Settings permet d'enlever cette contrainte et d'émettre sur toute la zone, ce qui peut être particulièrement utile sur 160m et 630m.

Lorsque vous avez les réponses AUTO activées et que vous avez sélectionné l'envoi d'Heartbeats Acknowledgements, votre station enverra un signal ACK à la station reçue l'informant que vous l'avez entendue. Ces réponses sont considérées comme des 'pulsations légères' de votre station, et réinitialisent le compte à rebours des HB.

L'intention des heartbeat n'est pas de tester la propagation, mais de vous aider à remplir votre Call list (fenêtre de droite) et ainsi savoir qui est potentiellement joignable pour un contact. Vous ne pouvez pas espérer de contact si vous ne les entendez pas (ou s'ils ne vous entendent pas).

Gardez à l'esprit que les HB n'ont pas vocation à démarrer une conversation. Lorsque vous allumez les HB, vous rejoignez le réseau des HB. Ceci vous permet de planifier des relais et d'envoyer des messages à stocker à ces différentes stations. Voyez ces HB et ACK comme un jalonnement du réseau et des stations relais (">") possible, tout comme un moyen d'envoyer des messages à lire plus tard (sortes de SMS) à travers ce réseau.

Pendant les HB, si une station a un message à délivrer à une autre station, ceci sera annoncé comme tel dans l'ACK :

KN4CRD: KM4ACK ACK -12 MSG 32 ↵

Pendant un QSO (ex lorsque vous recevez une transmission dans votre fenêtre de réception), la fonction HB se met en veille afin d'empêcher les interférences sur votre QSO.

Par ailleurs, sachez que les transmissions incontrôlées peuvent être contraires aux autorisations de votre licence. Par sécurité les HB ne devraient être envoyées que lorsque vous êtes devant votre station. Il existe une alerte d'inactivité (Settings>General>Behaviour> Idle Timeout) qui éteint votre HB après un temps sans action manuelle de votre part.

## CQ - Calling CQ

L'appel configuré par défaut se fait avec le message "CQCQCQ", mais vous pouvez le paramétrer dans les Settings. Voici les messages qui peuvent être contenus dans une trame de transmission de 15 secondes :

- CQCQCQ
- CQ CQ CQ
- CQ CQ
- CQ
- CQ CONTEST
- CQ FIELD
- CQ QRP
- CQ DX

Lorsque vous utilisez ces formats, vous pouvez aussi inclure votre locator avec 4 caractères et il sera encodé sur un cycle de transmission :

- CQCQCQ EM73
- CQ CQ CQ AA99
- CQ CQ BB88
- CQ CC77
- CQ CONTEST FN04
- CQ FIELD AB01
- CQ QRP JO42
- CQ DX GC28

Vous pouvez commencer votre message de CQ avec n'importe lequel de ces formats et il sera dirigés, c'est-à-dire que votre indicatif sera automatiquement inclus. Vous pouvez également ajouter sans souci ceci :

- CQ QRP 500MW CQ?

Si vous n'optez pas pour ces formats, vous n'enverriez pas un message dirigé, votre Grid ne serait pas incluse et vous auriez à ajouter votre indicatif.

Vous pouvez envoyer des CQ par intervalles, avec un clic-droit sur le bouton CQ afin de choisir le laps de temps pour la répétition. Votre station répétera le CQ jusqu'à ce que vous receviez une réponse.

## REPLY - Replying to a CQ

Par défaut la réponse à un CQ se fait avec "HW CPY?". Ceci permet à l'appelant de choisir avec qui se connecter et de lui envoyer un report SNRT. Vous pouvez personnaliser ce message avec votre propre réponse mais n'oubliez pas que la plupart des stations répondent avec des contenus qui tiennent dans un cycle de 15 secondes (ou en un minimum de cycle). Voici un exemple d'échange .

- → **KN4CRD: CQ QRP EM73 ↵**
- ← **DR4CNK: KN4CRD HW CPY? ↵**
- → **KN4CRD: DR4CNK SNR -12 TU 4 CALL QSL? ↵**
- ← **DR4CNK: KN4CRD RR -22 FB INTO GO28 GUD QRP DX! ↵**