

260911- Paroles dans ma musique / paroles sur Navidrome

En une phrase

Navidrome n'affiche dans son lecteur web que les paroles **synchronisées et embarquées dans les fichiers** ; on les fait chercher sur LRCLIB par Beets (en conteneur sur le NAS), un script les écrit dans les tags sans rien toucher d'autre, on pousse les fichiers modifiés vers kDrive, Navidrome les relit.

1. Le diagnostic — pourquoi « Absence de paroles » était normal

Le symptôme : *Dancing Queen* d'ABBA affichait « Absence de paroles » dans le lecteur web, alors qu'un fichier `Dancing Queen.lrc` synchronisé était posé à côté du MP3 et que l'API Subsonic (`getLyricsBySongId`) renvoyait bien ces paroles horodatées.

Vérifié dans le code de Navidrome 0.63.2 :

- **Le lecteur web ne passe pas par l'API Subsonic.** Il lit le champ `lyrics` de l'objet chanson (`ui/src/reducers/playerReducer.js`), c'est-à-dire la colonne `media_file.lyrics` de la base, remplie **au scan, depuis les tags embarqués seulement** (USLT/SYLT pour les MP3, `LYRICS` pour FLAC, `©lyr` pour M4A).
- **Il n'affiche que les entrées `synced: true`.** Un tag USLT en texte brut donne « Absence de paroles ».
- **Les fichiers `.lrc` / `.txt` posés à côté ne sont lus qu'à la volée,** par `core/lyrics` au moment d'une requête `getLyricsBySongId`, selon `LyricsPriority` (`.lrc` avant `embedded`). Donc Symfonium, Feishin, DSub... voient les 58 `.lrc` de la bibliothèque, le web non.

- `ND_LRCLIB_ENABLED=true`, posé sur le conteneur de la stack 42, **est inerte** : le binaire ne contient aucune chaîne `lrclib`. Navidrome ne va rien chercher en ligne sans plugin `LyricsProvider`.
- Navidrome lit bien la trame **SYLT** avec ses horodatages (ses tests `gotaglib_test.go` le montrent : `[00:02.50]English SYLT`), ainsi qu'un USLT contenant du texte LRC.

État de la bibliothèque le 2026-09-11 avant travaux : **16 077 titres, 485 avec paroles en base, 15 synchronisées** — donc 15 titres affichables dans l'interface web, sur 16 000.

2. Architecture retenue

Élément	Choix	Pourquoi
Où ça tourne	NAS sasnexe , Container Manager, sur <code>/volume1/music</code>	C'est l'original. Le VPS ne lit qu'un miroir kDrive (<code>/home/debian/music</code>) que <code>sync_kdrive_complete.sh</code> écrase chaque nuit : y écrire serait perdu
Recherche	Beets 2.14.0 (<code>lscr.io/linuxserver/beets</code>), plugin <code>lyrics</code> , source LRCLIB seule	Gratuit, sans clé, la seule source qui fournit des paroles synchronisées . Genius/Google ne donnent que du texte brut, invisible du web
Écriture dans les fichiers	<code>ecrire_paroles.py</code> (mutagen), pas Beets	Voir le piège n°1 : Beets réécrit tous ses champs
Propagation	<code>propager_paroles.sh</code> (rclone forcé)	Voir le piège n°2 : le padding ID3
Enchaînement	<code>finir_paroles.sh</code> en <code>setsid</code>	7 h de traitement, le NAS finit seul

Les fichiers : `docker-compose.yml`, `config.yaml` (Beets), `ecrire_paroles.py`, `propager_paroles.sh`, `finir_paroles.sh`, `README.md` — dans `Jux-scripts/Navidrome-Paroles/`, déployés sur le NAS dans `/volume1/homes/SAS_NEXTE/scripts/` et `/volume1/docker/beets-paroles/config/`.

Configuration Beets, l'essentiel :

```
import:
  autotag: no          # jamais de re-identification MusicBrainz
  copy: no
  move: no             # jamais de déplacement / renommage
  write: no            # Beets N'ECRIT JAMAIS dans les fichiers (piege n°1)
  incremental: yes
  duplicate_action: keep # piege n°4
ignore: ['#recycle', '@eaDir', '@__thumb', '.*', '*~', 'System Volume Information',
'lost+found']
```

```
plugins: lyrics web
lyrics:
  auto: no
  sources: [lrclib]
  synced: yes          # preferer les paroles horodatees
  keep_synced: yes     # ne jamais retoucher un titre deja synchronise
  force: yes           # re-interroger les 485 titres qui n'ont que du texte brut
  fallback: null       # rien trouve -> fichier intact
```

3. La procédure, telle qu'elle a été jouée le 2026-09-11

Sur le NAS (SSH `sas_nexte`, `docker` exige `sudo` et vit dans `/usr/local/bin/`) :

```
D=/usr/local/bin/docker
cd /volume1/docker/beets-paroles && sudo $D compose up -d
sudo $D exec beets-paroles beet version          # 2.14.0
sudo $D exec beets-paroles beet lyrics --help | grep keep-synced # l'option doit exister
```

Inventaire — Beets indexe la bibliothèque sans rien écrire (`-A` = tel quel, `-W` = aucun tag réécrit) :

```
sudo $D exec beets-paroles beet import -A -W /music      # 25 min, 15 770 titres
# + 25 dossiers reimportes avec -I (piege n°4)           -> 16 069 titres
sudo $D exec beets-paroles beet stats
```

Essai sur un album, avec repère posé avant :

```
/volume1/homes/SAS_NEXTE/scripts/propager_paroles.sh --repere
sudo $D exec beets-paroles beet lyrics 'album:Gold - Greatest Hits' # 19/19 en 20 s
sudo $D exec beets-paroles python3 /config/ecrire_paroles.py --simulation --filtre "ABBA/Gold"
sudo $D exec beets-paroles python3 /config/ecrire_paroles.py --filtre "ABBA/Gold" --controle
19
/volume1/homes/SAS_NEXTE/scripts/propager_paroles.sh # 19 fichiers, 4 s
```

Côté VPS, rafraîchir le listing du montage et scanner :

```
rcclone rc --rc-addr 127.0.0.1:5576 --rc-user=rcadmin --rc-pass='RcMusic2026!' vfs/refresh
recursive=true dir='ABBA/Gold - Greatest Hits'
curl "https://navidrome.juxjux.ovh/rest/startScan?u=julien&p=...&v=1.16.0&c=claude&f=json"
```

Résultat : *Dancing Queen* `synced: true` en base Navidrome, `start: 20320` (20,32 s), après un simple **quick scan** — la date du dossier ayant changé, Navidrome relit ses fichiers.

Toute la bibliothèque, sans surveillance :

```
/volume1/homes/SAS_NEXTE/scripts/propager_paroles.sh --repere
sudo $D exec -d beets-paroles sh -c 'beet lyrics >> /config/lyrics_run.log 2>&1; echo "FIN
rc=$?" >> /config/lyrics_run.log'
setsid /volume1/homes/SAS_NEXTE/scripts/finir_paroles.sh < /dev/null > /dev/null 2>&1 &
```

`finir_paroles.sh` attend la ligne `FIN`, lance `ecrire_paroles.py --controle 50` (50 fichiers tirés au hasard dont l'empreinte audio est comparée avant/après — le moindre écart arrête tout), puis `propager_paroles.sh`. **Il n'y a pas de propagation si l'écriture a le moindre échec** : on préfère un état non poussé à un état partiel.

Suivi :

```
tail /volume1/homes/SAS_NEXTE/logs/paroles_finir.log
sudo $D exec beets-paroles sh -c 'echo "$(grep -c "Found lyrics" /config/lyrics_run.log) /
$(grep -c "Fetching lyrics" /config/lyrics_run.log)'"
```

Mesures : lancé à 18h08, terminé à 0h34, **0,64 titre/s, 86 % de paroles trouvées** (voir le bilan en §7) ; le cron VPS de 4h30 (`vfs/refresh` + quick scan) relit ensuite la bibliothèque. Pour ne pas attendre : `bash /home/debian/navidrome_fullscan.sh`.

Contrôle final, sur le VPS :

```
sudo cp /home/debian/docker/navidrome/data/navidrome.db /tmp/nd.db
sudo cp /home/debian/docker/navidrome/data/navidrome.db-wal /tmp/nd.db-wal
sudo sqlite3 /tmp/nd.db "select count(*), sum(lyrics like '%"synced\':true%') from media_file
where lyrics not in ('','[]')"
```

Avant : `485|15`. **Dans le lecteur web, recharger la page et remettre le titre dans la file** : la file de lecture conserve l'objet chanson tel qu'il était au moment de l'ajout, sans les paroles.

4. Les pièges — dans l'ordre où ils ont mordu

1. Beets écrit tous ses champs, pas seulement les paroles. Premier essai avec

`import.write: yes` : sur *Dancing Queen*, Beets a réencodé toutes les trames texte et **ajouté** `TRCK 0/0`, `TPOS 0/0`, `TDRC 0000`, `TDOR 0000`, `TBPM 0`, `TCMP 0` et un `UFID` vide — des valeurs qu'il n'avait pas, écrites comme des défauts. Sur 16 000 fichiers, une pollution durable des tags. Les 19 originaux d'ABBA ont été restaurés depuis kDrive (`rclone copy ... --ignore-times`), Beets passé en `write: no`, et `ecrire_paroles.py` écrit lui-même, en ne touchant qu'aux trames de paroles : SYLT (horodatée) + USLT (texte brut) pour les MP3, `@lyr` pour M4A, `LYRICS` pour FLAC. Vérifié à l'octet sur ABBA : trames d'origine de tailles identiques, empreinte audio identique, ID3v1 conservé, propriétaire `admin:users` conservé.

2. Le padding ID3 rend les modifications invisibles à la synchro. Le WebDAV kDrive n'expose ni empreinte ni date fiable : `sync_kdrive_complete.sh` ne compare que les **tailles**. Or, mesuré sur 40 MP3 au hasard, **30 ont ≥ 2 048 octets de padding** — des paroles y tiennent souvent sans changer la taille du fichier. Le NAS aurait eu les paroles et Navidrome ne les aurait jamais vues. `propager_paroles.sh` liste les fichiers audio postérieurs à un repère et les pousse avec `rclone copy --files-from --ignore-times`, taille identique ou non. Le ré-envoi donne au fichier une nouvelle date côté kDrive, ce qui invalide aussi le cache VFS du VPS.

3. Les ACL Synology ne laissent passer que root dans le conteneur. `docker exec -u 1024:100` (l'uid `admin`, propriétaire des fichiers) échoue en `Permission denied` sur `/music` : le partage est en mode `000`, ses droits sont dans les ACL DSM, que le conteneur n'évalue pas pour un utilisateur ordinaire. Tout tourne donc en root dans le conteneur. Sans conséquence sur les fichiers : mutagen réécrit en place, même inode, le propriétaire reste `admin:users`. En revanche, un `rclone copy` de restauration lancé par `sas_nexte` **recrée** les fichiers sous cet utilisateur → `chown admin:users` derrière.

4. Vingt-deux albums sautés comme « doublons ». Même artiste et même titre d'album dans deux dossiers (*The Suburbs* et *The Suburbs _ Month Of May*, les disques 2 et 3 d'un best of, une édition deluxe...) : en mode silencieux Beets les saute (« This album is already in the library! »), et **l'état incrémental les note comme traités** — une relance ne les reprend pas. Correctif : `duplicate_action: keep`, puis réimport explicite des dossiers avec `-I`. Pour les extraire du journal, prendre la ligne qui précède chaque « already in the library » ; **les multi-disques y sont regroupés sur une seule ligne séparée par ;**, à découper. 15 770 → 16 069 titres.

5. Beets 2.x stocke les chemins relatifs à `directory`. `items.path` contient `ABBA/Gold - Greatest Hits/...`, pas `/music/...` : première passe de `ecrire_paroles.py` en 19 échecs « No such file ». Le script préfixe `/music`.

6. Deux fois du LRC = chaque ligne en double. Beets met du texte LRC dans SYLT **et** dans USLT ; Navidrome en fait deux entrées synchronisées, et le lecteur web concatène toutes les

entrées synchronisées. `ecrire_paroles.py` écrit le LRC dans SYLT et le texte brut dans USLT.

7. `nohup` ne détache pas sur DSM : `setsid ... < /dev/null > /dev/null 2>&1 &`, et vérifier par `ps -eo pid,sid,args -- ps w` tronque et fait croire à un processus mort.

5. Ce que la chaîne ne fait pas

- Elle ignore les 58 `.lrc` et 12 `.txt` déjà présents (LRCLIB retrouve ABBA lui-même). Les laisser : ils servent encore aux clients Subsonic.
- Elle ne renomme, ne déplace, ne ré-identifie rien.
- Elle n'efface jamais des paroles existantes et ne retouche pas un titre déjà synchronisé.
- Un titre absent de LRCLIB reste tel quel : normal pour les mixes, les instrumentaux et les raretés. Ne pas ajouter `genius` / `google` : texte brut seulement, invisible du web.

6. Pour un nouvel album, plus tard

```
D=/usr/local/bin/docker
sudo $D exec beets-paroles beet import -A -W /music          # incremental : seuls les nouveaux
dossiers
/volume1/homes/SAS_NEXTE/scripts/propager_paroles.sh --repere
sudo $D exec beets-paroles beet lyrics 'added:-lw..'          # ou 'album:Titre'
sudo $D exec beets-paroles python3 /config/ecrire_paroles.py
/volume1/homes/SAS_NEXTE/scripts/propager_paroles.sh
```

Puis le cron VPS de 4h30 fait le reste. Le conteneur `beets-paroles` peut rester arrêté entre deux usages (`sudo $D compose stop`) : sa base `/config/beets.db` et l'état `paroles_ecrites.json` survivent.

7. Bilan du 2026-09-12

Étape	Heure	Résultat
<code>beet lyrics</code> (16 069 titres)	18h08 → 0h34	13 830 paroles trouvées (86 %) , 0,64 titre/s
<code>ecrire_paroles.py --controle 50</code>	0h34 → 1h51	13 139 fichiers écrits, 10 979 synchronisés , 50 contrôles audio OK, 1 échec : un <code>.mp3.part</code> (téléchargement inachevé) — la chaîne s'est arrêtée avant propagation, comme prévu

Étape	Heure	Résultat
<code>propager_paroles.sh</code>	6h05 → 6h46	13 139 fichiers, 102,7 Gio, 0 erreur, 50 Mio/s — en session <code>setsid</code> , a survécu à une coupure SSH
<code>navidrome_fullscan.sh</code>	8h10 → 8h32	21 min
Base Navidrome		16 077 titres, 13 160 avec paroles, 10 998 synchronisées (avant : 485 / 15)

Poids ajouté : 48,6 Mio sur 102,6 Gio (+0,046 %), 3,9 Ko par fichier en moyenne. **8 815 fichiers sur 13 137 n'ont pas changé de taille** — les paroles ont tenu dans le padding ID3 : sans la propagation forcée, 67 % des fichiers seraient restés invisibles pour la synchro quotidienne. Le piège n°2 n'était pas théorique.

Le fichier `.mp3.part` de Suki Waterhouse est à supprimer à la main (ou à ajouter à `ignore` dans `config.yaml`).

Point ouvert : Julien constate 2–3 s de retard des paroles dans le lecteur web. Ce n'est ni le scan (l'affichage est piloté côté navigateur par la position de lecture) ni les données (LRCLIB et le `.lrc` d'origine s'accordent à 0,3 s près sur *Dancing Queen*). À tester dans Symfonium sur le même titre : en phase = le rendu du lecteur web (`react-jinke-music-player`) est en cause, et le modèle Navidrome connaît un `offset` LRC que le lecteur web ignore.