# qBittorrent: migrating seeding torrents from Windows to Linux

A guide for moving a large set of seeding torrents from a qBittorrent client on **Windows** to **qBittorrent in Docker on Linux** — without breaking the seeding (which matters for private trackers like TorrentLeech and their Hit-and-Run rules).

> Done 2026-06-29: 193 torrents (movies, anime, TV, PC games, OS ISOs) migrated from a Windows server to Docker qBittorrent on the arr-server. All 193 are seeding, verified to survive a reboot.

## Prerequisites

- The source files already live on the NAS (the same physical files the Linux client will seed).
- Both the Windows and Linux qBittorrent run a **standard release** (not a beta or fork) — otherwise the tracker may ban the client. qBittorrent 5.2.2 is fine for TorrentLeech.
- The torrents have already met the tracker's minimum seed time (if not: migrate anyway, but be aware of the Hit-and-Run risk).

## NAS layout (NFS on the arr-server)

| NAS | Export | Mount point |
|---|---|---|
| NAS01 | `/volume1/Movies` | `/mnt/media/movies` |
| NAS01 | `/volume1/Apps` | `/mnt/media/apps` |
| NAS02 | `/volume1/TV` | `/mnt/media/tv` |
| NAS02 | `/volume1/Anime` | `/mnt/media/anime` |
| NAS02 | `/volume1/Games-PC` | `/mnt/media/games-pc` |

Each share is in `/etc/fstab` with `_netdev` and bind-mounted into the qBittorrent container (`/mnt/media/x:/mnt/media/x`).

# Step 1 — Export from Windows

The `.torrent` and `.fastresume` files live in:

```
C:\Users\<user>\AppData\Local\qBittorrent\BT_backup\
```

Copy the **entire** folder (both the `.torrent` AND the `.fastresume` files) to somewhere Linux can reach, e.g. a temp folder on the NAS.

> **Note:** always keep an untouched copy as a "pristine source" — never touch it, always recreate from it.

# Step 2 — Docker volumes

The qBittorrent container must see the files at the **same path** the `.fastresume` points to. Add the NAS mounts to `docker-compose.yml`:

```
  qbittorrent:
    volumes:
      - ./config/qbittorrent:/config
      - /mnt/media/movies:/mnt/media/movies
      - /mnt/media/tv:/mnt/media/tv
      # ...and so on for each share
```

# Step 3 — Rewrite the paths in .fastresume

`.fastresume` is **bencode-encoded** (binary, with length prefixes). A plain `sed` will NOT work — you have to decode, swap the fields, and re-encode. Two fields hold paths:

- `save_path` (libtorrent's — the one that actually counts)

- `qBt-savePath` (qBittorrent's own)

**Mapping rule (Windows → Linux), prefix-preserving:**

| Windows save_path | Linux |
|---|---|
| `//nas01/Movies` (movie in a subfolder) | `/mnt/media/movies` |
| `//nas01/Movies/<movie>` (file directly) | `/mnt/media/movies/<movie>` |
| `//nas02/Anime/<x>` | `/mnt/media/anime/<x>` |
| `//nas02/TV/<x>` | `/mnt/media/tv/<x>` |

> **Important:** don't blanket-swap everything to the same folder. If the `save_path` includes the movie name, that means the content is the file directly — in which case the subfolder has to be preserved.

The Python script uses its own bencode decode/encode with a **safety check**: `encode(decode(file)) == file` must be byte-identical before anything is written (otherwise the encoder isn't canonical → abort). Stop qBittorrent before editing the files (it overwrites `.fastresume` on shutdown).

# Step 4 — Load them in and recheck

1. Copy the `.torrent` files + the rewritten `.fastresume` files into the Linux qBittorrent's `BT_backup`.
2. `docker compose up -d qbittorrent` (recreate if volumes changed).
3. qBittorrent rechecks and starts seeding. Torrents that find their files go to `stalledUP` / `uploading`.

**Common errors:**

- `missingFiles` = qBittorrent can't find the files → wrong path, a missing docker volume, or a folder name that doesn't match (e.g. a dedup that renamed it).
- **Trailing space in a folder name:** Windows allows/strips trailing spaces differently from Linux. Fix it qBittorrent-native with the `torrents/renameFolder` API (do NOT rename the folder on disk — Radarr/Jellyfin expect it without the space).

# Step 5 — Turn off the seeding queue

For a seedbox that should seed everything forever: turn off queueing so **all** torrents announce at once.

```
POST /api/v2/app/setPreferences   json={"queueing_enabled":false}
```

Otherwise `max_active_uploads` / `max_active_torrents` limit how many actually seed.

# Step 6 — Make it reboot-safe (critical!)

An fstab entry with `_netdev` survives a reboot — **but Docker doesn't wait for the NFS mounts by default.** If Docker starts qBittorrent before NFS is ready, empty folders get bind-mounted → every torrent becomes `missingFiles`.

**Fix** — a systemd drop-in at `/etc/systemd/system/docker.service.d/wait-for-nfs.conf`:

```
[Unit]
After=remote-fs.target
Wants=remote-fs.target
```

Then `systemctl daemon-reload`. This orders Docker after all `_netdev` mounts are ready.

**Verify with an actual reboot:** after restarting, all mounts should be active, qBittorrent up, and every torrent seeding (0 `missingFiles`).

# Verification commands

```
# Mounts active?
findmnt -t nfs4 -o TARGET,SOURCE

# fstab valid?
sudo findmnt --verify
```

```
# Can qBittorrent see the files?
docker exec qbittorrent ls /mnt/media/movies

# Torrent states (via the API, after getting a login cookie)
curl -s -b cookie 'http://localhost:8080/api/v2/torrents/info' \
  | grep -o '"state":"[^"]*"' | sort | uniq -c

# Tracker announcing? (status 2 = working)
curl -s -b cookie 'http://localhost:8080/api/v2/torrents/trackers?hash=<HASH>'
```

---

Revision #5
Created 2026-07-01 08:27:31 UTC by Admin
Updated 2026-07-10 15:24:47 UTC by Admin