

what does the "-1" mean in the NAS path /Volumes/920NAS-1 vs /Volumes/920NAS

ChatGPT said:

On macOS, the **/Volumes** directory is where external drives and network shares get mounted.

When you mount your Synology NAS share named **920NAS**, the system tries to mount it at:

```
/Volumes/920NAS
```

But if that path is **already taken** (maybe because the NAS is still mounted, or wasn't cleanly ejected, or a "ghost" mount point is lingering), macOS will automatically create a new mount point with a suffix like:

```
/Volumes/920NAS-1
```

What the -1 means

- It's just an **incremental suffix** added by macOS to avoid a naming conflict.
- If you mounted it again and both /Volumes/920NAS and /Volumes/920NAS-1 were taken, you'd see /Volumes/920NAS-2, and so on.

Why it happens

1. You disconnected from the NAS but Finder/Terminal didn't fully unmount /Volumes/920NAS.
2. A background process (like Spotlight indexing, Time Machine, or a file handle) is still holding onto the mount.
3. You remounted the same share, so macOS had to assign a new name (-1).

How to fix it

1. **Check active mounts:**
2. `mount`
3. `ls /Volumes`
4. **Eject the duplicate:**
5. `umount /Volumes/920NAS-1`

or

```
diskutil unmount /Volumes/920NAS-1
```

6. If you see ghost folders left behind (but not actually mounted), you can safely delete them:
7. `sudo rm -rf /Volumes/920NAS-1`

After a proper unmount, when you reconnect to the NAS, it should go back to `/Volumes/920NAS`.

👉 Do you want me to give you a quick **one-liner script** you can run that will automatically clean up those ghost `-1`, `-2`, etc. folders before mounting your NAS, so you always get the clean `/Volumes/920NAS` path?

You said:

Using macOS 15.7 and 16.0.1 in the Finder

ChatGPT said:

Got it — thanks for clarifying you're on **macOS 15.7 (Sequoia)** and **16.0.1 (Tahoe)**, working in **Finder**.

Here's how this plays out specifically in Finder:

Why Finder shows `/Volumes/920NAS-1`

- When you double-click your NAS share (or auto-connect on login), Finder tells macOS to mount the share at `/Volumes/920NAS`.
- If `/Volumes/920NAS` already exists — maybe because:
 - You didn't eject it last time,
 - A background process (Spotlight, Time Machine, or even Preview) is still using a file on that share,
 - Or Finder left a “ghost” mount folder behind — then macOS *automatically* adds `-1` to make a new mount point.

You'll see it in Finder's sidebar as **920NAS-1** instead of just **920NAS**.

How to get Finder back to plain `/Volumes/920NAS`

1. **Eject all duplicates**
In Finder's sidebar, eject every instance of `920NAS` (both the normal and the `-1`, `-2`, etc.).
2. **Check for ghosts**
Open Terminal and run:
3. `ls /Volumes`

If you still see `920NAS-1` (but it doesn't appear in Finder), it's just a leftover folder.

4. **Remove ghost folders** (safe as long as the NAS isn't mounted there):

5. `sudo rmdir /Volumes/920NAS-1`

(Use `rmdir` not `rm -rf` to avoid accidental data loss — if it's really mounted, `rmdir` will fail instead of deleting.)

6. **Reconnect in Finder**

Go to Finder → **Go > Connect to Server...** (⌘K), and connect again using your NAS's `smb/afp` path:

7. `smb://YourNASNameOrIP/920NAS`

Now it should mount cleanly back to `/Volumes/920NAS`.

How to prevent Finder from creating -1

- Always eject the share in Finder before shutting down or sleeping your Mac.
 - If you use **login items** to auto-connect, make sure only *one* copy of the share is in there.
 - If Time Machine or Spotlight are indexing the NAS, sometimes they delay unmounting — you may need to manually eject before disconnecting from the network.
-

✓ Bottom line: The `-1` is Finder's way of saying "I had to make a second mount because something was already using the first."

Once you eject and clear the ghosts, Finder will always mount it back as plain **920NAS**.

Do you want me to give you a simple **AppleScript / Automator shortcut** you can click in Finder that automatically ejects all NAS duplicates and remounts the clean one, so you don't have to dig into Terminal each time?

You said:

yes

ChatGPT said:

Thought for 19s

Awesome — here's a one-click solution you can use from Finder on **macOS 15.7 (Sequoia)** and **16.0.1 (Tahoe)**. It ejects any duplicate mounts (`...-1`, `-2`, etc.), cleans up ghost folders, and remounts the share cleanly as `/Volumes/920NAS`.

Option A — AppleScript app (fastest)

1. Open **Script Editor** (Spotlight → “Script Editor”).
2. File → **New**, paste this script, and edit the two lines under “CONFIG” (host/share and, if you’d like, username):

```
-- ===== CONFIG =====
set shareName to "920NAS" -- the volume name as it appears in Finder
set serverURL to "smb://YOUR-NAS-HOSTNAME-OR-IP/920NAS"
-- Examples:
-- set serverURL to "smb://192.168.1.50/920NAS"
-- set serverURL to "smb://username@nas.local/920NAS" -- will prompt for
password if not in Keychain
-- =====

-- Eject any mounted duplicates like 920NAS, 920NAS-1, 920NAS-2...
tell application "Finder"
    try
        set disksToEject to {}
        repeat with d in (get disks)
            set dn to name of d
            if dn starts with shareName then set end of
disksToEject to dn
        end repeat
        repeat with dn in disksToEject
            try
                eject disk (dn as text)
            end try
        end repeat
    end try
end tell
delay 1

-- Remove leftover ghost folders in /Volumes that aren't actually mounted
-- (safe: uses rmdir only and skips any path that is mounted)
set cleanupCMD to "
set -e
shopt -s nullglob
for d in /Volumes/" & shareName & "*; do
    [ -e \"$d\" ] || continue
    # skip if it's actually mounted
    if mount | grep -q \" on $d (\"; then
        continue
    fi
    rmdir \"$d\" 2>/dev/null || true
done
"
try
    do shell script cleanupCMD with administrator privileges
on error errMsg number errMsg
    -- Non-fatal: continue even if no ghosts
end try
delay 0.5
```

```

-- Mount cleanly
try
    mount volume serverURL
on error errMsg number errNum
    display dialog "Could not mount " & serverURL & " (" & errMsg & ")"
    buttons {"OK"} default button "OK" with icon caution
    return
end try
delay 0.5

-- Reveal in Finder
tell application "Finder"
    activate
    try
        reveal disk shareName
    end try
end tell

```

3. File → **Export...**

- **File Format:** Application
- (Optional) Check **“Run Only”**
- Save it as e.g. **“Mount 920NAS.app”** in Applications or your Desktop.

4. Double-click it any time. Add it to the Dock for a true one-click fix.

Notes

- If your NAS credentials are in **Keychain**, the mount is silent. If not, macOS will prompt once and can save them.
- Script uses `rmdir`, not `rm -rf`, so it won't delete anything that isn't an empty ghost folder.

Option B — Automator Quick Action (runs from Finder / Services menu)

1. Open **Automator** → **New Document** → **Quick Action**.
 2. “Workflow receives” → **no input in any application**.
 3. Add **Run AppleScript**, paste the exact script above.
 4. Save as **“Clean & Mount 920NAS”**.
 5. Trigger it via **Finder** → **Services** → **Clean & Mount 920NAS**, or bind a keyboard shortcut:
System Settings → **Keyboard** → **Keyboard Shortcuts** → **Services**.
-

Option C — Shortcuts app (big button or Dock tile)

1. Open **Shortcuts** → + → **New Shortcut** (name it “Mount 920NAS”).
 2. Add action **Run AppleScript**, paste the same script.
 3. Optional: Add to Menu Bar or Dock from the Shortcut’s settings (⊞ button).
-

Want me to pre-fill the script with your exact **NAS host/IP** and **username** so it’s truly turn-key?

You said:

but if I eliminate the duplicate share points or ghosts will i loose data somewhere

ChatGPT said:

Good question — and an important one.

Short answer: **no, you will not lose any data** if you remove the duplicate mount points (`/Volumes/920NAS-1`, `-2`, etc.).

Here’s why:

1. Mount points are *pointers*, not storage

- The `/Volumes/920NAS` (or `920NAS-1`) path is just where macOS **mounts** the remote filesystem.
 - Your real data lives on the NAS drive itself, not inside `/Volumes`.
 - The folders that hang around after an unclean eject are just **empty ghost directories** left behind on your Mac — they don’t contain your files.
-

2. Safe to remove

- If the share is actually mounted, the folder will be “in use” and `rmdir` will fail (so you won’t break anything).
 - If it’s not mounted, the folder is literally just an empty directory on your Mac. Removing it has no effect on the NAS.
-

3. When people get into trouble

The only risk is if someone uses `rm -rf` **blindly** inside `/Volumes/920NAS-1` while it *is* mounted. That would delete files from the NAS itself.

That's why I recommended `rmdir` (remove directory) instead:

- `rmdir` only succeeds if the directory is empty (ghost).
 - If it's a real mount, `rmdir` won't touch it.
-

4. What happens after cleanup

- You remove the ghosts → macOS has no conflicts → when you reconnect, the NAS mounts cleanly to `/Volumes/920NAS`.
 - Your data on the NAS is never touched in this process.
-

✅ Bottom line: **Removing the ghost folders is completely safe.**

They are *leftover placeholders* on your Mac, not your actual NAS files.

Would you like me to show you a **safe command** you can copy-paste in Terminal that will automatically remove only the empty ghost folders (and skip any real mounted shares), so there's zero risk?

You said:

i deleted the ghost path but when i try to make a fresh connection to the real path in finder i keep getting the incremental version.

ChatGPT said:

That's a classic Finder quirk — even after you delete the ghost folder, macOS sometimes still thinks `/Volumes/920NAS` is “busy,” so it keeps appending `-1`.

Here's how you can force it back to the clean `/Volumes/920NAS`:

1. Check if something is still mounted

Run this in **Terminal**:

```
mount | grep 920NAS
```

If you see something like:

```
smbfs on /Volumes/920NAS-1
```

then the system is still mounting it at -1.

2. Make sure nothing is hanging onto the path

Sometimes background services (Spotlight indexing, Preview, Time Machine) keep a file handle open.

You can check with:

```
lsof | grep /Volumes/920NAS
```

If any processes show up, quit them (or reboot to clear everything).

3. Kill Finder's “stale” mount

Run:

```
diskutil unmount force /Volumes/920NAS-1
```

Then check again:

```
ls /Volumes
```

Make sure neither 920NAS nor 920NAS-1 is there.

4. Manually reconnect *cleanly*

Instead of clicking the sidebar, go to **Finder** → **Go** → **Connect to Server...** (⌘K) and enter:

```
smb://<NAS-IP-or-hostname>/920NAS
```

It should now mount at /Volumes/920NAS.

5. If it still keeps making -1

Sometimes the *sidebar favorite* or *login item* is pointing to the -1 path.

- Go to **System Settings** → **General** → **Login Items** → remove any duplicate NAS entries.
- In Finder's **Sidebar Favorites**, right-click → Remove any stale shortcuts (especially ones that show -1). Then reconnect via ⌘K and drag the clean share back into Favorites.

✓ After you do that cleanup, Finder should stop defaulting to -1 and always mount to plain **/Volumes/920NAS**.

Do you want me to give you a **single Terminal script** that will:

1. Force-unmount all duplicates (-1, -2, etc.),
2. Remove ghosts,
3. Reconnect cleanly to the NAS in one step —
so you don't have to keep doing the dance manually?

You said:

yes

ChatGPT said:

Thought for 17s

Perfect. Here's a **single, safe Terminal script** that force-unmounts any duplicates (...-1, -2), removes only-empty ghost folders, and remounts your NAS cleanly as **/Volumes/920NAS**. It works on macOS 15.7 (Sequoia) and 16.0.1 (Tahoe).

1) Copy this script (edit the CONFIG at the top)

```
#!/usr/bin/env bash
set -euo pipefail

# ===== CONFIG: EDIT THESE TWO LINES =====
SHARE_NAME="920NAS" # Volume name you want in /Volumes
SERVER_URL="smb://YOUR-NAS-HOST/920NAS" # e.g. smb://192.168.1.50/920NAS or
smb://user@nas.local/920NAS
# =====

MOUNT_BASE="/Volumes"
TARGET_PATH="${MOUNT_BASE}/${SHARE_NAME}"
```

```

echo "==> Cleaning up existing mounts for ${SHARE_NAME}..."

# 1) Unmount anything already mounted as SHARE_NAME or SHARE_NAME-#
while mount | grep -E " on ${MOUNT_BASE}/${SHARE_NAME}(-[0-9]+)? " >/dev/null
2>&1; do
    # Grab each current mount path and unmount it
    while read -r __ mp _; do
        if [[ -n "${mp:-}" ]]; then
            echo "    • Unmounting ${mp}"
            diskutil unmount force "${mp}" >/dev/null || true
        fi
    done <<(mount | awk -v base="${MOUNT_BASE}/${SHARE_NAME}" ' $3 ~ "^"base"(-[0-9]+)?$' {print $0}')
    sleep 0.5
done

# 2) Remove only-empty ghost folders in /Volumes (safe: rmdir won't remove
non-empty/mounted dirs)
echo "==> Removing empty ghost folders (if any)..."
# We may need sudo if ghosts are owned by root
needs_sudo=""
if [[ ! -w "${MOUNT_BASE}" ]]; then needs_sudo="sudo"; fi

for d in "${MOUNT_BASE}/${SHARE_NAME}"*; do
    [[ -e "$d" ]] || continue
    # Skip if actually mounted
    if mount | grep -q " on ${d} "; then
        echo "    • Skipping mounted path ${d}"
        continue
    fi
    # Try to remove only if empty
    if ${needs_sudo} rmdir "$d" 2>/dev/null; then
        echo "    • Removed ghost ${d}"
    fi
done

# 3) Mount cleanly via Finder (osascript uses Keychain if creds are saved)
echo "==> Mounting ${SERVER_URL}..."
osascript -e 'mount volume "'${SERVER_URL}'/"' || {
    echo "!! Could not mount ${SERVER_URL}. Check host/share/credentials."
    exit 1
}
sleep 1

# 4) Verify where it actually mounted and prefer the clean path
actual_mount=""
while read -r __ mp _; do
    case "${mp}" in
        "${MOUNT_BASE}/${SHARE_NAME}"|${MOUNT_BASE}/${SHARE_NAME}-*)
            actual_mount="${mp}"
            break
        ;;
    esac
done <<(mount | grep -E " on ${MOUNT_BASE}/${SHARE_NAME}(-[0-9]+)? ")

if [[ -z "${actual_mount}" ]]; then

```

```

    echo "!! Mounted share not found under /Volumes. Something blocked the
mount."
    exit 2
fi

# 5) If macOS still assigned -1, tell you why and how to fix
if [[ "${actual_mount}" != "${TARGET_PATH}" ]]; then
    echo "!! macOS mounted as: ${actual_mount} (not ${TARGET_PATH})."
    echo "    This means something still reserved ${TARGET_PATH} during mount."
    echo "    Tips:"
    echo "    - Remove any Login Items or Finder sidebar favorites that point
to the old \"-1\"."
    echo "    - Quit apps indexing/using the share (Spotlight, Time Machine,
Preview), then re-run."
else
    echo "==> Mounted cleanly at ${TARGET_PATH}"
fi

# 6) Reveal in Finder
open "${actual_mount}"

```

2) How to use it

1. Open **TextEdit**, paste the script, edit the two **CONFIG** lines:
 - o `SHARE_NAME="920NAS"`
 - o `SERVER_URL="smb://YOUR-NAS-HOST/920NAS" (or`
`smb://username@nas.local/920NAS)`
2. Save as **Mount-920NAS.command** (Plain Text) on your Desktop.
3. In Terminal:
4. `chmod +x ~/Desktop/Mount-920NAS.command`
5. Double-click **Mount-920NAS.command** whenever you want a clean mount.

Notes & safety

- The script uses **rmdir** (not `rm -rf`) to remove only **empty** ghost folders. That keeps you safe from deleting live data.
- If credentials are saved in **Keychain**, the mount is silent; otherwise macOS will prompt once and can save them.
- If you still get `-1` after this, it's almost always a **login item** or **Finder sidebar favorite** that's pointing at an old path. Remove those, then run the script again.

Want me to prefill the `SERVER_URL` with your actual NAS hostname/IP and (optionally) username so it's drop-in?

You said:

is there any way to clear the mac finder "stale" mounts, i tried the terminal command 'diskutil unmount force /Volumes/920NAS-1'

ChatGPT said:

Yes — Finder is notorious for hanging onto “stale” mount points, even after you’ve run

```
diskutil unmount force /Volumes/920NAS-1
```

That command ejects the filesystem, but the *folder stub* (or Finder’s internal reference) may linger. Here are the steps to fully clear them so you can get back to the clean `/Volumes/920NAS` path:

1. Check what’s really mounted

Run:

```
mount | grep 920NAS
```

If nothing shows, the share isn’t mounted anymore — good. If you still see `/Volumes/920NAS-1`, it means something is holding onto it.

2. Make sure no processes are using the share

Sometimes Spotlight, Preview, Time Machine, or Finder itself keeps file handles open. To check:

```
lsof | grep /Volumes/920NAS
```

If you see a process, quit that app (or `kill PID` if necessary).

3. Remove the leftover folder

If the share is unmounted but the folder is still sitting there:

```
sudo rmdir /Volumes/920NAS-1
```

⚠ Use `rmdir`, not `rm -rf`.

- `rmdir` only removes empty directories.
 - If it’s actually mounted, `rmdir` fails instead of touching your data.
-

4. Restart Finder’s mount tracking

Finder sometimes caches stale volumes. Reset it by:

```
killall Finder
```

Finder will relaunch and forget about the stale mount.

5. Reconnect cleanly

Now reconnect with:

- **Finder** → **Go** → **Connect to Server...** (⌘K) and enter
 - `smb://<NAS-IP-or-hostname>/920NAS`
 - It should now mount back to `/Volumes/920NAS` (without the `-1`).
-

6. Prevent it from coming back

- Check **System Settings** → **General** → **Login Items** → remove any old `920NAS-1` entries.
 - Remove any **Finder sidebar favorites** that point at `-1`.
 - Always eject the share before shutting down or sleeping.
-

✓ With that sequence — force unmount, kill Finder, remove ghosts, reconnect — you should be able to clear the stale mounts for good.

Would you like me to give you a **single one-liner shell command** that force-unmounts, removes ghosts, and restarts Finder in one go (so you don't have to run 3–4 commands each time)?

You said:

even though the incremental share is not showing up in terminal using `'lsof | grep /Volumes/920NAS'` when i open the finder after `'killall Finder'` i still get the incremental share rather than the clean one.

ChatGPT said:

Ah, that helps narrow it down — if `lsof` shows nothing, and you've unmounted/removed the ghosts, but Finder *still* insists on `920NAS-1` after a `killall Finder`, then the problem isn't a live process hanging on. It's usually one of these two “sticky” cases:

Why Finder keeps forcing the -1

1. Stale path reference in Finder sidebar or login items

- If at some point you dragged 920NAS-1 into the Finder sidebar or added it as a Login Item, macOS will keep trying to mount at that exact path.
- It won't downgrade it to 920NAS — it honors the stored path, so you keep getting -1.

2. Hidden ghost mount still reserving /Volumes/920NAS

- Even if it doesn't show in `ls` or `mount`, sometimes `/Volumes/920NAS` exists as a *folder with root ownership or hidden ACLs*. That blocks the clean mount and forces macOS to go to -1.

Step-by-step fix

1. Make sure nothing is mounted

```
mount | grep 920NAS
```

If you see nothing, good.

2. Check for hidden /Volumes/920NAS stub

```
ls -la /Volumes | grep 920NAS
```

If you see a 920NAS folder but it's not mounted, that's your blocker.
Remove it safely:

```
sudo rmdir /Volumes/920NAS
```

3. Clear Finder sidebar favorites

Open Finder → **Preferences** (⌘,) → **Sidebar**.

- If you see 920NAS-1 listed, uncheck/remove it.
Also, right-click in the Finder sidebar itself and “Remove from Sidebar.”

4. Check Login Items

System Settings → **General** → **Login Items**.

- If you see 920NAS-1 there, remove it.
This is a common culprit: macOS keeps reconnecting the old path.

5. Reconnect fresh

Now use Finder → **Go** → **Connect to Server...** (⌘K) with:

```
smb://<NAS-IP-or-hostname>/920NAS
```

It should mount cleanly at /Volumes/920NAS.

✅ If you've done all that and Finder *still* goes to -1, the next thing is to clear **Saved Server Aliases** in Finder's "Recent Servers" and in Keychain (they sometimes remember the full mount path including -1).

Do you want me to give you the exact commands to **list and delete any stale Keychain entries** for your NAS share? That's often the final piece when Finder keeps forcing the wrong mount.

You said:

when trying terminal command 'sudo rmdir /Volumes/920NAS-1' the response is 'Resource busy' so how do i determine what's still holding on to that incremental share

ChatGPT said:

Great detail — the **Resource busy** response means /Volumes/920NAS-1 is **still an active mount point**, even if it looks like you've ejected it. macOS won't let you remove a directory that a filesystem is mounted on.

That tells us something is still holding onto the share. Let's chase it down step by step:

1. Double-check what's mounted

Run:

```
mount | grep 920NAS
```

If you see something like:

```
smbfs on /Volumes/920NAS-1 (server ...)
```

then the share is *definitely still mounted*. It's not a ghost; it's live.

2. Identify who's using it

Use:

```
lsof +D /Volumes/920NAS-1
```

This will list every process that has **any open file/handle** inside that mount.

- If the output is long, look at the first column (process name) and second (PID).
- Quit those apps or kill the PIDs if they're hanging.

Note: `lsof | grep /Volumes/920NAS` may miss things because it only looks for matches in the line, while `lsof +D` walks the whole directory tree.

3. Force-unmount the share

Once you know nothing critical is using it, try:

```
diskutil unmount force /Volumes/920NAS-1
```

If that fails, escalate with:

```
sudo umount -f /Volumes/920NAS-1
```

4. If it *still* refuses (very stubborn mounts)

Sometimes macOS keeps a network share “busy” even after processes are gone. In that case, you can:

- **Log out/in** — kills the SMB session and Finder state.
 - Or **reboot** — brute-force, but guarantees release.
-

5. After it's released

Once the mount is gone, `/Volumes/920NAS-1` should just be a folder. Then:

```
sudo rmdir /Volumes/920NAS-1
```

will succeed.

At that point, reconnecting in Finder should mount cleanly at /Volumes/920NAS.

✅ So the trick is: `lsof +D /Volumes/920NAS-1` → find the process → quit/kill it → `umount -f`.

Do you want me to give you a **one-liner** that combines those steps, automatically shows which processes are holding /Volumes/920NAS-1, and unmounts it if safe?

You said:

yes

ChatGPT said:

Thought for 15s

Here you go—two copy-paste **one-liners**. Run #1 first (safe: shows the blockers). If you're sure it's OK to close them, run #2 (aggressive: kills, unmounts, removes the stub).

1) SAFE: list exactly what's holding /Volumes/920NAS-1

```
M="/Volumes/920NAS-1"; PIDS=$(sudo lsof -t +D "$M" 2>/dev/null | sort -u); if [ -n "$PIDS" ]; then echo "🔒 Processes holding $M:"; sudo lsof +D "$M"; echo; echo "🔓 To release: sudo kill $PIDS"; else echo "No open handles. Forcing unmount..."; sudo umount -f "$M" && sudo rmdir "$M" && echo "✅ Unmounted and removed stub."; fi
```

2) AGGRESSIVE: kill holders, then unmount & remove

```
M="/Volumes/920NAS-1"; PIDS=$(sudo lsof -t +D "$M" 2>/dev/null | sort -u); [ -n "$PIDS" ] && echo "Killing: $PIDS" && sudo kill $PIDS && sleep 1; sudo umount -f "$M" 2>/dev/null || diskutil unmount force "$M"; sudo rmdir "$M" 2>/dev/null && echo "✅ Cleared $M" || echo "ℹ️ If rmdir still fails, log out/in or reboot to clear SMB session."
```

Tip: after clearing, reconnect via  with `smb://<NAS>/920NAS`. If it *still* mounts as -1, remove any Login Items or Finder sidebar favorites pointing to the old -1 path, then try again.

You said:

smbfs on /Volumes/920NAS-1 (server ...)

ChatGPT said:

Got it — since your `mount` output shows

```
smbfs on /Volumes/920NAS-1 (server ...)
```

that means the NAS share is **still actively mounted**, not just a ghost folder. That's why `sudo rmdir` gives "Resource busy" — macOS won't remove a live mount point.

Here's how to find *what's still holding it* and cleanly release it:

1. See who is using it

Run:

```
lsof +D /Volumes/920NAS-1
```

That lists every process with open files inside that mount.

- First column = process name, second column = PID.
 - Example: `Finder`, `mds_stores` (Spotlight), or `backupd` (Time Machine).
-

2. If you find processes

- Quit the app if it's something like `Finder`, `Preview`, or `Excel`.
 - Or, if it's a background process, you can kill it:
 - `sudo kill -9 <PID>`
-

3. Force unmount once it's clear

```
diskutil unmount force /Volumes/920NAS-1
```

If that doesn't work:

```
sudo umount -f /Volumes/920NAS-1
```

4. Remove the leftover folder

Once unmounted, `/Volumes/920NAS-1` will just be a directory. Then:

```
sudo rmdir /Volumes/920NAS-1
```

5. Reconnect cleanly

Use Finder → Go → Connect to Server... (⌘K) and enter:

```
smb://<NAS-IP-or-hostname>/920NAS
```

Now it should mount at `/Volumes/920NAS`.

✓ Summary:

- `smbfs` on `/Volumes/920NAS-1` means it's still live.
 - Use `ls -lsof +D` to see which process is keeping it busy.
 - Kill or quit those processes, then force-unmount, then remove the folder.
-

Do you want me to give you a **ready-to-run script** that will:

1. List the processes holding `/Volumes/920NAS-1`,
2. Kill them (with confirmation),
3. Unmount and remove the path —
all in one go?

You said:

the only pid using the incremental share is the Finder itself, so even when i try connecting to the volume on my nas cleanly i end up with an incremental share. why

ChatGPT said:

That's the "last mile" frustration with Finder mounts — if **Finder itself** is the only PID using `/Volumes/920NAS-1`, here's what's really going on:

Why Finder forces the `-1`

1. **Finder has an open handle on the mount point itself**

Even after you unmount and kill ghosts, Finder keeps a reference to the *last* location you used (the `-1` path). So when you reconnect, Finder tells macOS: *"Remount that server I know as `/Volumes/920NAS-1`."*

macOS honors that, instead of reusing `/Volumes/920NAS`.

2. The clean /Volumes/920NAS folder is “blocked”

If there is **any leftover directory** named /Volumes/920NAS — even empty — Finder can’t use it, so the SMB client adds -1.

3. Saved server paths

Finder remembers shares by their *full path* (/Volumes/920NAS-1) in:

- Finder sidebar favorites
- Login Items (auto-connect)
- Recent Servers
- Even in your Keychain as a “network password” item

So even when you think you’re connecting cleanly, Finder is reconnecting using the old -1 reference.

How to clear this loop

Step 1 — Unmount fully

```
diskutil unmount force /Volumes/920NAS-1
sudo umount -f /Volumes/920NAS-1
```

Step 2 — Remove both stubs

```
sudo rmdir /Volumes/920NAS-1 2>/dev/null
sudo rmdir /Volumes/920NAS 2>/dev/null
```

This clears both the clean and incremental folders if they’re just directories.

Step 3 — Restart Finder (to drop its handle)

```
killall Finder
```

Step 4 — Clear Finder’s stale references

- In **Finder Preferences** → **Sidebar**, remove 920NAS-1.
- In **System Settings** → **General** → **Login Items**, remove any 920NAS-1 entries.
- In **Finder** → **Go** → **Connect to Server...** (⌘K) → **Recent Servers**, clear the list (⇧⌘K to open, then remove).

Step 5 — Check Keychain

Open **Keychain Access** → search for 920NAS.

If you see an entry that includes /Volumes/920NAS-1, delete it.

(The NAS password itself is fine to leave — it’s just the stale mount path you don’t want remembered.)

Step 6 — Reconnect clean

Now use ⌘K and type explicitly:

```
smb://<NAS-IP-or-hostname>/920NAS
```

It should finally mount at `/Volumes/920NAS`.

✅ **In short:** Finder itself is the culprit — it cached the `-1` path and keeps reusing it. Once you **remove both ghost folders + clear Finder’s saved references + restart Finder**, the clean `/Volumes/920NAS` mount will work again.

Would you like me to give you a **scripted “reset” command** that does *all six steps at once* (force-unmount, remove ghosts, restart Finder, and clear Recent Servers) so you don’t have to keep chasing it manually?

You said:

Does it even matter that i can't log into volume without the incremental extension

ChatGPT said:

That’s a smart, bottom-line question. Short answer: **it usually doesn’t matter at all**. The `-1` (or `-2`, etc.) is just Finder’s way of making sure it has a unique, available mount point under `/Volumes`.

✅ What *doesn’t* change

- **Your NAS data:** The files live on the NAS, not on the `/Volumes` folder. Whether you mount at `/Volumes/920NAS` or `/Volumes/920NAS-1`, you’re pointing to the *same share* on the NAS.
 - **File integrity:** No risk of corruption or loss simply because the path has `-1`.
 - **Performance:** Access speed, reliability, and SMB behavior are identical.
-

⚠️ When it *can* matter

1. **Scripts or apps that hard-code the path**

- If you (or an app) expect files at `/Volumes/920NAS/...`, and Finder mounted it as `/Volumes/920NAS-1/...`, the path won't resolve.
 - Examples: shell scripts, backup jobs, symlinks, automations.
 - 2. **Saved aliases/shortcuts**
 - Aliases in Finder, Login Items, or apps can break if they point to `/Volumes/920NAS` but the active mount is `/Volumes/920NAS-1`.
 - 3. **Multiple mounts**
 - If you genuinely have two sessions mounted to the same NAS share, you'll see `920NAS` and `920NAS-1` simultaneously. That can get confusing if you aren't careful which one you're writing to.
-

Practical takeaway

- If you **only use Finder manually** (clicking into the NAS, browsing, dragging files): it doesn't matter if it's `920NAS` or `920NAS-1`.
 - If you **depend on fixed paths** (Terminal scripts, Time Machine, Automator workflows, rsync/cron jobs, etc.), then yes — it matters, because those paths must be stable.
-

👉 So here's the test: Do you have *any scripts or apps* that explicitly point at `/Volumes/920NAS`?

- If **no**, you can safely ignore the `-1` and move on.
 - If **yes**, then we should fix the mount path or adjust your scripts to accept either `/Volumes/920NAS*`.
-

Do you want me to show you a quick **bash trick** that automatically detects whether your NAS is mounted as `/Volumes/920NAS` or `/Volumes/920NAS-1` (or `-2`, etc.), and then uses whichever is present — so your scripts never break again?

You said:

yes

ChatGPT said:

Perfect — here's a **bash snippet** you can drop into any script that needs to use your NAS, and it will automatically resolve to the *actual* mounted path, whether Finder gave you `/Volumes/920NAS`, `/Volumes/920NAS-1`, or even `/Volumes/920NAS-7`.

Auto-detect NAS mount path

```
#!/usr/bin/env bash

# Base share name (edit this to match your NAS share name)
SHARE_NAME="920NAS"

# Look for the first matching mounted path under /Volumes
MOUNT_PATH=$(mount | awk -v base="/Volumes/$SHARE_NAME" '$3 ~ "^"base"(-[0-9]+)?$" {print $3; exit}')

if [[ -z "$MOUNT_PATH" ]]; then
    echo "❌ NAS $SHARE_NAME is not mounted."
    exit 1
fi

echo "✅ Using NAS path: $MOUNT_PATH"

# Example: cd into the NAS share
cd "$MOUNT_PATH" || exit 1

# Run your commands here
# e.g. list files
ls -lh
```

How it works

- `mount | awk ...` searches for `/Volumes/920NAS`, `/Volumes/920NAS-1`, `/Volumes/920NAS-2`, etc.
 - `exit` after the first match ensures you always get the “active” mount.
 - If no match is found, it prints an error and exits gracefully.
-

Example output

If Finder mounted -1, you'll see:

```
✅ Using NAS path: /Volumes/920NAS-1
```

If Finder mounted cleanly:

```
✅ Using NAS path: /Volumes/920NAS
```

How to use

1. Save it as `nas-path.sh` (or embed it at the top of your backup/automation scripts).

2. Make executable:
3. `chmod +x nas-path.sh`
4. Run it:
5. `./nas-path.sh`

✅ This way, it **doesn't matter anymore** if macOS gives you the plain share or an incremental one — your scripts will always adapt.

Would you like me to expand this so it **auto-mounts the NAS if it's not mounted at all** (using your `smb://` address), so the script both finds *and* connects on its own?

You said:

yes

ChatGPT said:

Absolutely. Here's a **drop-in macOS script** that:

1. Detects your NAS mount path (handles `/Volumes/920NAS`, `-1`, `-2`, ...)
2. **Auto-mounts** it if it isn't mounted (uses Keychain via Finder)
3. Returns the resolved path for your script to use

Copy, edit the CONFIG, save as `nas_mount.sh`, `chmod +x nas_mount.sh`, then run it or source it from other scripts.

```
#!/usr/bin/env bash
# macOS 15/16: auto-detect & auto-mount SMB share, returning the live
# /Volumes path.
# Safe: uses Finder/Keychain to mount; never deletes anything.
set -euo pipefail

##### CONFIG #####
SHARE_NAME="920NAS" # The volume name you expect in
/Volumes
SERVER_URL="smb://YOUR-NAS-HOST/920NAS" # e.g.
smb://192.168.1.50/920NAS or smb://user@nas.local/920NAS
MOUNT_BASE="/Volumes"
WAIT_SECONDS=15 # How long to wait for the
mount to appear
#####

find_mount() {
    mount | awk -v base="${MOUNT_BASE}/${SHARE_NAME}" ' $3 ~ "^"base" (-[0-9]+)?$' {print $3; exit}
}

ensure_not_blocked_dir() {
```

```

# If a leftover empty folder blocks clean mount, this quietly removes it.
# (No data risk: rmdir only succeeds if empty & not a live mount.)
local p
for p in "${MOUNT_BASE}/${SHARE_NAME}" "${MOUNT_BASE}/${SHARE_NAME}-1"
"${MOUNT_BASE}/${SHARE_NAME}-2"; do
    [[ -e "$p" ]] || continue
    if mount | grep -q " on $p "; then
        continue # it's a live mount, do nothing
    fi
    rmdir "$p" 2>/dev/null || true
done
}

mount_via_finder() {
    # Prefer Finder/Keychain so creds are pulled automatically (no plain-text
    passwords).
    /usr/bin/osascript -e 'mount volume "'${SERVER_URL}/\"/\\\""'
}

# 1) If it's already mounted, just use it.
MOUNT_PATH="$(find_mount || true)"
if [[ -n "${MOUNT_PATH}" ]]; then
    echo "✅ Using existing mount: ${MOUNT_PATH}"
    # Export for caller scripts if they source this file
    export NAS_PATH="${MOUNT_PATH}"
    exit 0
fi

# 2) Not mounted: clear harmless blockers, then request a clean mount.
ensure_not_blocked_dir

echo "🔗 Mounting ${SERVER_URL}... (using Finder/Keychain)"
if ! mount_via_finder; then
    echo "⚠️ Finder mount failed. Check server/creds/network. (SERVER_URL:
${SERVER_URL})" >&2
    exit 2
fi

# 3) Wait for the system to attach it (handles -1, -2, etc.)
for _ in $(seq 1 "${WAIT_SECONDS}"); do
    sleep 1
    MOUNT_PATH="$(find_mount || true)"
    [[ -n "${MOUNT_PATH}" ]] && break
done

if [[ -z "${MOUNT_PATH}" ]]; then
    echo "❌ Mounted share not found under ${MOUNT_BASE}/${SHARE_NAME}* after
${WAIT_SECONDS}s." >&2
    exit 3
fi

# 4) Report the final, real path (with or without -1). Export for caller
scripts.
echo "✅ Mounted as: ${MOUNT_PATH}"
export NAS_PATH="${MOUNT_PATH}"

```

```
# Optional: cd into it if you want this script to "do work" next.
# cd "${NAS_PATH}"
# ls -lh
```

How to use it

- **Standalone:**
- `./nas_mount.sh`
- `#` Output shows where it mounted. Use the path it prints.
- **From your other scripts (recommended):**
- `source /path/to/nas_mount.sh`
- `#` Now `$NAS_PATH` is guaranteed valid (e.g., `/Volumes/920NAS-1` or `/Volumes/920NAS`)
- `rsync -av "$NAS_PATH/Backups/" "/some/local/dir/"`

Notes (so you're never surprised)

- This prefers **Finder/Keychain** mounting, which keeps credentials out of your shell history and plays nicely with SMB on macOS.
- If Finder previously cached a `-1` path, this still works—your scripts consume `${NAS_PATH}`, which is always the *actual* live mount.
- If you *must* force the plain `/Volumes/920NAS` name for some legacy job, we can add a preflight step that fully evicts Finder references and remounts—but practically, letting scripts follow `${NAS_PATH}` is cleaner and bulletproof.

Want me to tailor `SERVER_URL` with your real NAS host/IP and (if needed) username so you can paste-run with no edits?

You said:

how can i force mount

ChatGPT said:

When you say “**force mount**” in macOS, there are two layers to think about:

1. Force unmount first (clear what's blocking)

If `/Volumes/920NAS` is already “busy” and Finder keeps pushing you to `-1`, you need to clear the old mount point. These two commands are the hammer:

```
diskutil unmount force /Volumes/920NAS
sudo umount -f /Volumes/920NAS
```

and for any incremental mounts:

```
diskutil unmount force /Volumes/920NAS-1
sudo umount -f /Volumes/920NAS-1
```

Then remove the ghost folders:

```
sudo rmdir /Volumes/920NAS /Volumes/920NAS-1 2>/dev/null
```

At this point, the path `/Volumes/920NAS` should be free.

2. Force a fresh mount

You can do this in two ways:

A. Through Finder / AppleScript

This uses macOS's Keychain-stored credentials and Finder's SMB client:

```
osascript -e 'mount volume "smb://USER@NAS-IP/920NAS"'
```

👉 Replace `USER` and `NAS-IP` with your username and NAS host/IP.

B. Directly with the SMB client (`mount_smbfs`)

This bypasses Finder completely and mounts right at the path you choose:

```
sudo mkdir -p /Volumes/920NAS
sudo mount_smbfs //USER@NAS-IP/920NAS /Volumes/920NAS
```

- `USER@NAS-IP` → your NAS account and address
- It will prompt for your password (or use your Keychain if cached)
- Because you explicitly give `/Volumes/920NAS`, you won't get `-1` unless the folder is still occupied

To disconnect later:

```
sudo umount /Volumes/920NAS
```

3. Why “force mount” matters

- Finder is polite — if `/Volumes/920NAS` looks occupied, it just creates `/Volumes/920NAS-1`.

- Using `mount_smbfs` with your own mount point **forces it to mount exactly where you tell it**, and fails if that location isn't free. That's the closest thing to a "force mount" in macOS.
-

✓ So:

- Run a **force unmount** first, clear the ghost directory,
 - Then use `mount_smbfs` to mount the share *directly* at `/Volumes/920NAS`.
-

Do you want me to write you a little **shell script wrapper** that does all of that in one go (force unmount any old copies, clear ghosts, and remount directly at `/Volumes/920NAS` with `mount_smbfs`)? That would give you a true one-click "force mount" every time.

You said:

how do i unmount a nas volume `/Volumes/920NAS`

ChatGPT said:

On macOS there are a few reliable ways to **unmount a NAS share** (like `/Volumes/920NAS`). Which one you use depends on how stubborn the mount is. Here's the rundown:

✓ Standard (cleanest)

```
diskutil unmount /Volumes/920NAS
```

This asks macOS nicely to unmount the share. If nothing is holding it open, it'll disappear immediately from Finder.

✓ Force if busy

If you get "*Resource busy*" or it refuses:

```
diskutil unmount force /Volumes/920NAS
```

or

```
sudo umount -f /Volumes/920NAS
```

These tell macOS to drop the mount even if some process is holding a file open.

✅ Finder method

- In Finder's sidebar, click the eject icon (🔌) next to 920NAS.
 - Or right-click the volume → **Eject**.
This is just the GUI wrapper for the same unmount.
-

🔍 If unmount fails with “Resource busy”

That means a process still has files open on that share. You can identify it:

```
lsof +D /Volumes/920NAS
```

This will show the process name + PID. Quit that app (or `kill <PID>` if needed), then retry the unmount.

✅ Cleanup after unmount

Once it's unmounted, check `/Volumes` for leftovers:

```
ls /Volumes
```

If `/Volumes/920NAS` still exists but is just an empty directory, you can safely remove it:

```
sudo rmdir /Volumes/920NAS
```

👉 Question for you: do you want me to give you a **ready-to-run one-liner** that will (1) kill any processes using `/Volumes/920NAS`, (2) force unmount it, and (3) remove the leftover folder all in one shot?

You said:

i am seeing `'/Volumes/920NAS'` in response to the `'ls /Volumes'` command, however `'/Volumes/920NAS'` is not showing up in the finder

ChatGPT said:

That's a really common Finder/macOS quirk with network mounts. What you're seeing is the difference between a **mount point folder** and an **actual mounted volume**:

What's happening

- `ls /Volumes` shows `/Volumes/920NAS` → This means there's a *directory* named `920NAS` under `/Volumes`.
- **Finder does not show it in the sidebar** → Because Finder only shows it when it's a **mounted volume**, not just a directory.

In other words, it's an **empty ghost folder** that got left behind after the NAS was unmounted (or unmounted incorrectly).

How to confirm it's *not* mounted

Run:

```
mount | grep 920NAS
```

- If nothing comes back, it's **not mounted**, and the folder is just a leftover.
 - If you see something like `smbfs` on `/Volumes/920NAS ...`, then it really *is* still mounted (just not showing in Finder's sidebar — sometimes Finder loses track).
-

How to safely clean it

If it's not mounted (no output from the `mount` command), you can remove the folder:

```
sudo rmdir /Volumes/920NAS
```

 Use `rmdir`, not `rm -rf`.

- `rmdir` only succeeds if the folder is empty.
 - If it's actually mounted, `rmdir` will fail with “Resource busy,” which protects your data.
-

Then reconnect cleanly

Now reconnect with:

- Finder → **Go** → **Connect to Server...** (⌘K) →
 - `smb://<NAS-IP-or-hostname>/920NAS`
 - This time it should mount as a proper volume and appear in Finder (and you won't get a -1 unless `/Volumes/920NAS` is still occupied).
-

✓ Summary:

- If `ls` shows it but `mount` doesn't, it's just an empty leftover folder. Delete it with `rmdir`.
 - Finder hides such folders, which is why you don't see them in the sidebar.
 - Once cleaned, the next mount will be fresh and visible.
-

Would you like me to give you a **one-liner** that checks if `/Volumes/920NAS` is a ghost (not really mounted), and if so, safely deletes it — but refuses to touch it if it's still in use?

You said:

how to determine if the hidden attribute is set

ChatGPT said:

Good question 👍 — on macOS, a folder (or file) under `/Volumes` can be **hidden from Finder** even if it exists, because of the `hidden` file attribute.

Here's how you can check if `/Volumes/920NAS` has the hidden flag set:

1. Use `ls` with flags

```
ls -ldO /Volumes/920NAS
```

- The `O` option shows extended file flags.
- Example output:
- `drwxr-xr-x 5 user staff hidden 160 Oct 7 15:32 /Volumes/920NAS`

→ Notice the `hidden` flag.

If no `hidden` flag shows, then the directory isn't hidden — it's just not a real mounted volume.

2. Use `GetFileInfo` (from Xcode command line tools)

If you have Xcode tools installed:

```
GetFileInfo -aV /Volumes/920NAS
```

- It will show something like `attributes: avbstclnmedz` where the letters represent flags.
 - If `v` appears, that means “invisible in Finder.”
-

3. Use `xattr`

Some files are hidden via extended attributes:

```
xattr /Volumes/920NAS
```

If you see `com.apple.FinderInfo` or `com.apple.hidden`, that’s a sign Finder is being told to hide it.

Clearing the hidden flag

If you do find it has the hidden flag, you can clear it with:

```
sudo chflags nohidden /Volumes/920NAS
```

or (for extended attributes):

```
sudo xattr -d com.apple.FinderInfo /Volumes/920NAS 2>/dev/null
sudo xattr -d com.apple.hidden /Volumes/920NAS 2>/dev/null
```

Important distinction:

- If `mount` shows nothing for `/Volumes/920NAS`, it’s not mounted anyway — it’s just a leftover folder.
 - If it *is* mounted but hidden, clearing the flag will make it visible in Finder again.
-

 Do you want me to give you a **one-liner** that tests `/Volumes/920NAS` and tells you:

- whether it’s mounted,
- whether it’s hidden,
- and whether it’s just a ghost folder?

You said:

what is the kill finder command

ChatGPT said:

On macOS, the way to "kill" (really: quit and restart) Finder is with the `killall` command.

Standard restart

```
killall Finder
```

- This sends Finder a polite **TERM** signal.
 - Finder will quit and automatically relaunch itself.
 - Safe to run — it's equivalent to doing ⌘ -right-click the Finder icon → Relaunch.
-

Force-kill (if Finder is hung)

```
killall -9 Finder
```

- This sends a **KILL** signal (cannot be ignored).
 - Finder quits immediately and will still auto-relaunch.
 - Use only if Finder won't respond to the normal command.
-

What happens after

- Your Finder windows/desktop disappear briefly, then Finder relaunches.
 - Any stale references it had (like cached NAS mount paths) are dropped.
-

⚠ Note: This does not unmount network volumes. It only clears Finder's memory of them. To truly disconnect, you still need `diskutil unmount` or `umount`.

👉 Do you want me to show you a **combined one-liner** that both force-unmounts your `/Volumes/920NAS*` mounts *and* restarts Finder in the same go, so you're guaranteed a clean slate?

You said:

```
sudo xattr -d com.apple.FinderInfo /Volumes/920NAS 2>/dev/null sudo xattr -d  
com.apple.hidden /Volumes/920NAS 2>/dev/null
```

ChatGPT said:

Exactly 👍 — those two commands attempt to remove any **extended attributes** (`xattr`) that tell Finder to hide a folder:

```
sudo xattr -d com.apple.FinderInfo /Volumes/920NAS 2>/dev/null  
sudo xattr -d com.apple.hidden /Volumes/920NAS 2>/dev/null
```

What each does

- `com.apple.FinderInfo` → holds Finder-specific metadata (including “invisible” flags).
- `com.apple.hidden` → explicitly marks the item hidden from Finder.
- `2>/dev/null` → just suppresses the error message if the attribute doesn’t exist.

So if those attributes are present, they’ll be deleted; if they aren’t, nothing happens.

How to check before/after

Run:

```
xattr /Volumes/920NAS
```

- If you see those attributes listed, Finder may be hiding it.
 - After running your two commands, run again — if nothing shows, the folder has no extra hidden attributes.
-

Important note

- If `/Volumes/920NAS` is still mounted, removing the hidden flags just makes it visible in Finder again.
 - If it’s *not* mounted and is only a leftover empty folder, Finder won’t show it regardless — because Finder only displays mounted volumes in the sidebar.
-

✅ **So those commands are safe:** they just strip away Finder’s “hide this” flags.

Would you like me to give you a single **diagnostic one-liner** that tells you whether `/Volumes/920NAS` is (a) mounted, (b) hidden, or (c) just a ghost directory?

You said:

how do i add 'wheel' as a user permission so not hidden

ChatGPT said:

Got it 👍 — let's clear up what's going on with **permissions** vs. **hidden flags** on macOS:

1. "wheel" group

- `wheel` is a **system group** on macOS (UID/GID 0).
- By default, only `root` and some system services are members.
- Normal users usually aren't in `wheel`.
- It's not used to decide whether something is "hidden" in Finder — that's controlled by **file flags** (`hidden`) or **extended attributes** (`com.apple.hidden`, `com.apple.FinderInfo`).

So: adding `wheel` ownership/permissions won't make a folder visible in Finder.

2. Visibility in Finder

Finder decides whether to display `/Volumes/920NAS` based on:

- **Is there an active mount?** (`mount` command shows it)
- **Does it have the `hidden` file flag?** (`ls -ldO` shows `hidden`)
- **Does it have extended attributes marking it hidden?** (`xattr` shows `com.apple.hidden` or `com.apple.FinderInfo`)

Permissions (`ls -l` showing owner/group like `root wheel`) don't affect visibility.

3. If you really wanted to change group

You *can* change the group ownership of `/Volumes/920NAS` to `wheel` with:

```
sudo chgrp wheel /Volumes/920NAS
```

and give group members read/write:

```
sudo chmod g+rw /Volumes/920NAS
```

...but this won't change whether it's hidden. It just changes filesystem access rights.

✅ What actually makes `/Volumes/920NAS` not hidden

1. Remove the hidden flag (if set):
2. `sudo chflags nohidden /Volumes/920NAS`
3. Remove hidden attributes (if set):
4. `sudo xattr -d com.apple.hidden /Volumes/920NAS 2>/dev/null`
5. `sudo xattr -d com.apple.FinderInfo /Volumes/920NAS 2>/dev/null`

Now it will show in Finder if it's mounted.

⚠️ **Key point:** Finder never shows a plain folder in `/Volumes` unless it's a real mounted volume. If `/Volumes/920NAS` is just a leftover directory (not mounted), changing flags, groups, or permissions won't make it appear — it's just an empty directory.

👉 Do you want me to give you a **single script** that checks if `/Volumes/920NAS` is (1) mounted, (2) hidden, or (3) just a ghost, and then fixes visibility automatically?

You said:

yes

ChatGPT said:

Awesome — here's a **safe, self-diagnosing bash script** for macOS 15/16 that:

- Detects whether `/Volumes/920NAS` is **mounted**, **hidden**, or a **ghost** folder
- Shows a clear status report
- Optionally **fixes** visibility (removes hidden flags/attrs)
- Optionally **cleans** an empty ghost (with `rmdir` only — no `rm -rf` danger)

Save as `nas_vol_check.sh`, `chmod +x nas_vol_check.sh`, then run:

- Diagnose only: `./nas_vol_check.sh`
- Diagnose + fix visibility: `./nas_vol_check.sh --fix`
- Diagnose + fix + remove empty ghost: `./nas_vol_check.sh --fix --clean`

```
#!/usr/bin/env bash
# macOS 15/16: Diagnose & fix /Volumes/920NAS visibility & ghost issues
set -euo pipefail
```

```

VOL="/Volumes/920NAS"
DO_FIX=false
DO_CLEAN=false

# Parse flags
for arg in "$@"; do
    case "$arg" in
        --fix) DO_FIX=true ;;
        --clean) DO_CLEAN=true ;;
        -h|--help)
            echo "Usage: $0 [--fix] [--clean]"
            echo "  --fix    : clear hidden flags/attrs if present"
            echo "  --clean  : if NOT mounted and empty, remove ghost dir with
rmdir"
            exit 0
            ;;
        esac
    done

headline() { echo; echo "== $* =="; }
status()    { echo "• $*"; }
warn()      { echo "!! $*" >&2; }

is_mounted() {
    mount | awk -v p="$VOL" '$3==p{found=1} END{exit(found?0:1)}'
}

has_hidden_flag() {
    # returns 0 if 'hidden' flag present
    local out
    out=$(ls -ldO "$VOL" 2>/dev/null || true)
    [[ "$out" == *" hidden "* ]]
}

has_xattr() {
    xattr -p "$1" "$VOL" >/dev/null 2>&1
}

is_empty_dir() {
    [[ -d "$VOL" ]] && [[ -z "$(ls -A "$VOL" 2>/dev/null || true)" ]]
}

headline "Check: $VOL"

# 1) Existence
if [[ ! -e "$VOL" ]]; then
    status "Path does not exist."
    echo "Summary: NOT PRESENT"
    exit 0
fi

# 2) Mounted?
if is_mounted; then
    status "Mounted: YES"
else
    status "Mounted: NO (likely a ghost directory)"

```

```

fi

# 3) Hidden flags/attrs
if has_hidden_flag; then
    status "File flag: hidden = YES"
else
    status "File flag: hidden = NO"
fi

if has_xattr "com.apple.hidden"; then
    status "xattr: com.apple.hidden = PRESENT"
else
    status "xattr: com.apple.hidden = ABSENT"
fi

if has_xattr "com.apple.FinderInfo"; then
    status "xattr: com.apple.FinderInfo = PRESENT"
else
    status "xattr: com.apple.FinderInfo = ABSENT"
fi

# 4) If not mounted, is it empty (ghost)?
if ! is_mounted; then
    if is_empty_dir; then
        status "Directory is empty (safe ghost)."
    else
        status "Directory has contents (unexpected for a ghost)."
    fi
fi

# 5) Fixes (optional)
if $DO_FIX; then
    headline "Applying visibility fixes (--fix)"
    if has_hidden_flag; then
        sudo chflags nohidden "$VOL" || warn "Failed to clear hidden flag"
    fi
    # Remove xattrs quietly if present
    sudo xattr -d com.apple.hidden "$VOL" 2>/dev/null || true
    sudo xattr -d com.apple.FinderInfo "$VOL" 2>/dev/null || true
    status "Cleared hidden indicators (if any)."
fi

if $DO_CLEAN; then
    if is_mounted; then
        warn "Not removing: path is mounted. Unmount first."
    else
        if is_empty_dir; then
            headline "Removing empty ghost (--clean)"
            sudo rmdir "$VOL" && status "Ghost removed."
        else
            warn "Not empty - refusing to remove."
        fi
    fi
fi

echo
echo "Summary:"

```

```

echo " Mounted : $(is_mounted && echo YES || echo NO) "
echo " Hidden flag : $(has_hidden_flag && echo YES || echo NO) "
echo " xattr hidden: $(has_xattr com.apple.hidden && echo PRESENT || echo ABSENT) "
echo " FinderInfo : $(has_xattr com.apple.FinderInfo && echo PRESENT || echo ABSENT) "
if ! is_mounted; then
  echo " Empty dir : $(is_empty_dir && echo YES || echo NO) "
fi

```

What this gives you

- A clear **readout** of the state (mounted/hidden/ghost)
- **One switch** to unhide (`--fix`)
- **One switch** to remove only an **empty** ghost safely (`--clean`)
- Zero path hard-coding headaches — it never uses `rm -rf` (we like our data right where it is)

If you want, I can tailor this to accept your **share name** as an argument (so you can run it for any `/Volumes/<share>`), or wire it into your earlier auto-mount script for a full “check → fix → mount” flow.

You said:

how do i know if this volume is a ghost

ChatGPT said:

Great question. On macOS, a “**ghost volume**” in `/Volumes` is simply a leftover directory that Finder or the OS didn’t clean up after a network share or disk was unmounted. Here’s how to tell if `/Volumes/920NAS` is a ghost or a real mount:

1. Check if it’s actually mounted

```
mount | grep /Volumes/920NAS
```

- **If you see a line like:**
`smbfs on /Volumes/920NAS (server ...) → it’s a real active mount.`
- **If nothing prints:** → it’s not mounted (likely a ghost).

2. See if Finder thinks it’s mounted

In Finder → **Go** → **Computer** (⇧⌘C), look for `920NAS`.

- If it doesn't appear but the folder exists in `/Volumes`, that's another clue it's a ghost.
-

3. Check what's inside

```
ls -A /Volumes/920NAS
```

- If it's empty, it's almost certainly a ghost.
 - If it has files, double-check whether those are really your NAS files (compare with the NAS web interface). Ghosts are normally empty, but sometimes an app can dump files there by mistake.
-

4. Try removing it

```
sudo rmdir /Volumes/920NAS
```

- If it removes cleanly, it was just a ghost.
 - If you get `Resource busy`, that means it's still mounted (go back to step 1).
-

✓ Quick rule of thumb

- **Mounted in `mount` → Real**
 - **Not mounted, empty dir, only in `ls` → Ghost**
-

Do you want me to give you a **single one-liner** that runs these checks and prints: “Mounted”, “Ghost (empty)”, or “Unexpected contents”?

You said:

how do i clean up a directory that Finder or the OS didn't clean up after a network share or disk was unmounted. H

ChatGPT said:

On macOS, it's pretty common that after you eject a network share or external disk, the mount point under `/Volumes` is left behind as an **empty “ghost” directory**. Finder won't show it, but `ls /Volumes` will. Here's the safe way to clean it up:

Step-by-step: Cleaning a ghost directory

1. Verify it's really unmounted

Run:

```
mount | grep /Volumes/YourShareName
```

If you see nothing → it's not mounted (safe to clean).

If you see a line like `smbfs on /Volumes/YourShareName (server ...)` → it's still mounted.

Use:

```
diskutil unmount force /Volumes/YourShareName
```

or

```
sudo umount -f /Volumes/YourShareName
```

2. Check contents (should be empty)

```
ls -la /Volumes/YourShareName
```

- If it's empty (. and . . only), it's just a ghost folder.
 - If it has files, those are local files accidentally copied into the ghost (not on the NAS anymore). Move them out before removal.
-

3. Remove the ghost safely

```
sudo rmdir /Volumes/YourShareName
```

 Use `rmdir`, not `rm -rf`.

- `rmdir` will only succeed if the directory is empty and not mounted.
 - If it's busy, you'll get "Resource busy" (means still mounted).
 - If it's not empty, you'll get "Directory not empty" (check for stray files).
-

4. Confirm cleanup

```
ls /Volumes
```

Your ghost directory should be gone. Next time you mount the share, macOS will use the clean path (no -1).

Quick one-liner (detect & clean)

Here's a safe command that checks if `/Volumes/920NAS` is mounted, and only removes it if it's an empty ghost:

```
VOL="/Volumes/920NAS"; mount | grep -q " $VOL " && echo "❌ Still mounted" ||  
( [ -d "$VOL" ] && [ -z "$(ls -A "$VOL")" ] && sudo rmdir "$VOL" && echo "✅  
Removed ghost" || echo "ℹ️ No ghost found")
```

✅ Summary:

- Always check with `mount` first.
 - Use `rmdir` to remove only-empty ghosts (safe).
 - Never `rm -rf` unless you're 100% sure it's just a ghost, because that can delete real files accidentally copied in.
-

Do you want me to extend that one-liner so it will **scan all of /volumes**, find *every* ghost directory left behind, and clean them up in one sweep?

You said:

how do i clean up the `/Volumes/920NAS` NAS directory that Finder or the OS didn't clean up after a network share or disk was unmounted.

ChatGPT said:

Perfectly clear 👍 — you want to get rid of the **leftover /Volumes/920NAS folder** that sometimes sticks around after unmounting a NAS share. The trick is to confirm it's *not mounted anymore*, then remove the empty directory safely.

Here's the safe process:

Step-by-step cleanup

1. Check if it's still mounted

```
mount | grep /Volumes/920NAS
```

- **If you see nothing** → it's not mounted (good).
- **If you see `smbfs` on `/Volumes/920NAS`** ... → it's still live. Eject it first:
- `diskutil unmount force /Volumes/920NAS`

or

```
sudo umount -f /Volumes/920NAS
```

2. Make sure it's empty

```
ls -la /Volumes/920NAS
```

If it only shows `.` and `..`, it's an empty ghost.

⚠ If you see actual files, those are **local stray files** (not on your NAS anymore). Move them somewhere safe before deleting.

3. Remove the ghost safely

```
sudo rmdir /Volumes/920NAS
```

- Use `rmdir`, not `rm -rf`.
 - `rmdir` only removes empty directories.
 - If it's busy → you'll see `Resource busy` (still mounted).
 - If not empty → you'll see `Directory not empty` (check for stray files).
-

4. Confirm cleanup

```
ls /Volumes
```

You shouldn't see `920NAS` anymore. Next time you connect, Finder will mount cleanly at `/Volumes/920NAS` instead of making `/Volumes/920NAS-1`.

✅ Summary:

- Confirm with `mount` → unmount if necessary
- Confirm it's empty → move stray files if any
- Use `rmdir` to delete ghost

- Never `rm -rf` unless you are 100% sure
-

👉 Would you like me to give you a **ready-to-run one-liner** that checks `/Volumes/920NAS`, tells you if it's mounted, ghost, or contains stray files, and only removes it if it's a safe ghost?

You said:

yes

ChatGPT said:

Here's a **safe one-liner** you can paste into Terminal to check `/Volumes/920NAS` and clean it up only if it's an empty ghost:

```
VOL="/Volumes/920NAS"; \
if mount | grep -q " $VOL "; then \
    echo "❌ $VOL is still mounted - unmount first."; \
elif [ -d "$VOL" ]; then \
    if [ -z "$(ls -A "$VOL" 2>/dev/null)" ]; then \
        sudo rmdir "$VOL" && echo "✅ Removed empty ghost $VOL"; \
    else \
        echo "⚠️ $VOL exists but is not empty - investigate contents before deleting."; \
    fi; \
else \
    echo "ℹ️ $VOL does not exist."; \
fi
```

🔍 What it does:

- If it's still mounted → tells you to unmount first.
 - If it's just a directory:
 - **Empty** → deletes safely with `rmdir`.
 - **Not empty** → warns you so you don't delete stray files accidentally.
 - If it doesn't exist → tells you nothing to do.
-

Would you like me to expand this into a version that **scans all of /volumes**, automatically cleaning every ghost directory (but never touching mounted or non-empty ones)?

You said:

the response was `'/Volumes/920NAS exists but is not empty – investigate contents before deleting'` so how do i investigate

ChatGPT said:

That's the right next step 👍 — the script refused to delete because there's **something inside** /Volumes/920NAS. That's good protection, because sometimes macOS or an app accidentally writes local files into the ghost folder after the NAS disconnects.

Here's how to safely investigate:

1. List what's inside

```
ls -la /Volumes/920NAS
```

- -l = long listing (permissions, size, date)
- -a = show hidden files (like .DS_Store, .TemporaryItems)

This will tell you whether it's just metadata junk or actual files/folders.

2. Check file sizes

```
du -sh /Volumes/920NAS/*
```

- du -sh shows the size of each item.
 - If everything is tiny (0B, 4K, a few KB), it's likely harmless junk.
 - If you see MB/GB, then real files were written into the ghost path while the NAS wasn't connected.
-

3. If they're junk system files

Common harmless things Finder leaves behind:

- .DS_Store
- .Spotlight-V100
- .TemporaryItems
- .Trashes

You can delete those safely:

```
sudo rm -rf /Volumes/920NAS/.DS_Store /Volumes/920NAS/.Spotlight-V100  
/Volumes/920NAS/.TemporaryItems /Volumes/920NAS/.Trashes
```

Then check again:

```
ls -la /Volumes/920NAS
```

If empty now → remove the directory:

```
sudo rmdir /Volumes/920NAS
```

4. If there are real files/folders

- Compare them to what's actually on your NAS (via the NAS web interface or reconnecting cleanly).
- If they're **duplicates** of files already on the NAS → you can delete them from the ghost.
- If they're **unique** (accidentally saved locally when the NAS was offline), copy them somewhere safe first:
- `cp -R /Volumes/920NAS ~/Desktop/NAS_ghost_recovered/`

Once you're sure nothing valuable is left:

```
sudo rm -rf /Volumes/920NAS/*  
sudo rmdir /Volumes/920NAS
```

Summary:

- `ls -la` → see what's inside.
- `du -sh` → check if it's junk or real data.
- Delete junk, rescue real files, then `rmdir`.