

Lisa Image Verification Tools — Manual

Companion diagnostics for **lisafile.py**. These read Lisa DC42 floppy images on the PC and answer one question before you spend a Lisa boot cycle: *is this image safe to inject into and copy from?*

Covers: **chaincheck.py 2B** (the pre-flight verifier) and **checklen.py** (catalog length inspector, occasional use).

Written August 2026, following the XFER_v8 failure investigation. Tag-format ground truth from lisafsh-tool.c (arcanebyte/lisaem) and Tom Stepleton's LisaList2 references, verified against Lisa-created files.

1. chaincheck.py — tag chain verifier

What it checks

Every 512-byte block on a Lisa floppy carries a 12-byte tag identifying which file owns it, which page of that file it is, and where the next and previous blocks of the file live. The OS driver walks these chains when reading a file. If a chain points to a block that has been freed or reallocated, the driver stops with **Error 659 "Corrupt File System header chain found in driver"** — the error that killed the BIGFILE copy on XFER_v8.

chaincheck decodes every real file's chain and verifies:

- exactly one chain head (a block whose prev-link is an end marker)
- the chain walks from head to an end marker without hitting a block owned by another file or free space
- every next-link is answered by a matching back-link
- the chain covers every block tagged with that file's ID
- relative page numbers run 0..N-1 in chain order

What it deliberately does NOT flag

Fragmentation is legal. A file's blocks do not need to be contiguous, and on a volume with interleaved copies the spans of different files overlap freely. chaincheck reports

fragmentation as an informational note on the summary line — (fragmented: N foreign/free blocks inside span) — and only fails the file if the *chain itself* is broken. Version 2A treated every foreign block inside a span as a problem and produced hundreds of false alarms on a healthy disk; 2B fixed this.

Usage

```
python chaincheck.py IMAGE                check every real file
python chaincheck.py IMAGE 9              check one file ID (from lisafile list)
```

Run it on the working copy AND on the SD card copy — the two have diverged more than once:

```
python chaincheck.py NuTransporter.image
python chaincheck.py J:\NuTransporter.image
```

Reading the output

Healthy file:

```
file ID 13: 78 blocks, span 184..261, chain OK, pages 0..77 -- OK
```

Healthy but fragmented (still fine to use):

```
file ID 9: 228 blocks, span 49..526 (fragmented: 250 foreign/free
blocks inside span), chain OK, pages 0..227 -- OK
```

Damaged (the XFER_v8 signatures — do not use as a donor):

```
file ID 8: 21 blocks, span 112..135 (fragmented: 3 foreign/free blocks
inside span) *** 2 PROBLEM(S) ***
CHAIN BREAK: block 121 (page 9) -> block 122, which is free
(chain reached 10 of 21 blocks before breaking)

file ID 9: 219 blocks, span 299..526 *** 1 PROBLEM(S) ***
chain covers 60 of 219 tagged blocks
```

A **CHAIN BREAK** means the chain routes into a freed/foreign block. A **coverage shortfall** ("covers 60 of 219") means the chain terminates early and orphans the rest of the allocation. Either one will produce Error 659 the moment the Lisa reads past the damage. There is no host-side repair — recreate the file on the Lisa (or rebuild the

floppy) rather than attempting surgery.

Exit status

- 0 — every checked file clean (`RESULT: all chains clean`)
- 1 — problems found
- 2 — usage error

So it can gate a batch script: inject only if chaincheck succeeds.

When to run it

1. **After creating donors on the Lisa, before the first inject.** This is the mandatory pre-flight. It would have caught the XFER_v8 damage in one second instead of one day.
2. **On the card copy** whenever behavior differs between PC and Lisa.
3. **After any suspicious Lisa-side error** (659, 972, 1288, Scavenger complaints) to establish whether the image or the session is at fault.

2. checklen.py — catalog length inspector

One-purpose diagnostic: prints the two recorded-length fields (big-endian u32 at name+45 and name+49) for every occurrence of a filename in the image.

```
python checklen.py IMAGE EXAMPLE.TEXT      one file
python checklen.py IMAGE                    sweep every *.TEXT name found
```

Notes on interpretation:

- **The Lisa records the raw allocation** — an N-block file records N*512 (verified: EXAMPLE.TEXT, 24 blocks, records 12,288). lisafile writes the same convention. If these disagree after an inject, the tool has a bug.
- Expect **two hits per name**: the live catalog slot and the parallel structure copy. On a healthy file both show the same value. Extra hits with garbage values are just the filename string occurring inside some file's *content* — ignore any hit whose values don't decode to a sane block count.

Use it only when lengths are specifically in question; chaincheck is the routine tool.

3. The standard workflow (where these tools fit)

```
LISA
```

- | 1. Format fresh floppy under LOS 3.1
- | 2. File-mgr C: copy .TEXT donors at the names and sizes you need (donor type must be .TEXT)
- | 3. Power off cleanly, WAIT a few seconds, pull card

```
PC
```

- | 4. chaincheck.py IMAGE -> must exit 0
- | 5. lisafire.py inject IMAGE DONOR.TEXT source.txt
- | 6. copy image to card
- | 7. lisafire.py list J:\IMAGE -> confirm current

```
LISA
```

- | 8. Boot with floppy mounted (never hot-swap images)
- | 9. File-mgr C: -LOWER-NAME.TEXT -> -#12-NAME.TEXT
- | 10. Compile from -#12-, link, run

Step 3 exists because the probable cause of the XFER_v8 damage was pulling the SD card before the Floppy Emu finished flushing large multi-block copies — scattered single blocks reverted to their prior free state, which is invisible until something reads the file and hits 659. Step 4 makes that failure mode detectable in one second. Step 7 defends against the recurring stale-copy problem (Desktop vs. card divergence).

4. Tag format reference (400K/800K floppy, 12-byte tags)

```

bytes 0-1    version
bytes 2-3    volume id
bytes 4-5    file ID (big-endian u16)
bytes 6-7    file-relative page number
bytes 8-9    next block:  physical = (value & 0x7ff) + 28
bytes 10-11  prev block:  same encoding
(value & 0x7ff) == 0x7ff  means end of chain; high bits are flags

```

File IDs: real files are > 0x0007 and < 0x7fff. System IDs: 0 free, 1 MDDF, 2 free-bitmap, 3 s-records, 4 catalog, 0x7fff erased, plus the 0xFFxx negative IDs for boot-track structures. ProFile/Widget tags are 20 bytes with different field offsets — these tools are floppy-only.

DC42 container: 84-byte header, 800 x 512 data, 800 x 12 tags. Checksums at header

offsets 72 (data) and 76 (tags); tag checksum skips the first block's tag. Floppy Emu ignores checksums; Disk Copy 4.2 enforces them.

5. Version history

- **chaincheck 1** (unnumbered) — guessed tag layout; fwd/back read as absolute block numbers. Produced hundreds of false errors on healthy files. Retired.
- **chaincheck 2A** — correct decoding (lisafsh-tool ground truth): +28 base, 0x7ff end markers, relative pages. Found the real XFER_v8 damage, but counted legal fragmentation as problems.
- **chaincheck 2B** (current) — fragmentation demoted to informational; fails only on genuine chain pathology. Validated both ways: flags XFER_v8's known damage, passes NuTransporter's healthy fragmented volume.
- **checklen 1** — as delivered; established that recorded length = raw allocation (N*512).

Known limits: 400K DC42 only (size-checked); does not verify the free bitmap or s-records against the tag chains; no repair capability by design — the "Lisa allocates, host fills" philosophy applies to fixing as much as creating.

The version gremlin note, earned twice: the docstring on line 2 states the version. If output doesn't match this manual, check that line before anything else.