



ADIF Editor

A native macOS editor for amateur radio ADIF log files, with
purpose-built tools for Parks on the Air.



User manual

Version v26.8.18 · 22 August 2026

Tim Annable · WW8L

Contents

- 1 What this is and what it deliberately is not
 - 2 Installing including the first-launch warning
 - 3 Opening a log any file, whatever its extension
 - 4 The window grid, columns, toolbar
 - 5 Editing cells, sorting, selecting, moving QSOs
 - 6 Finding and the trick that replaces filtering
 - 7 Duplicates found, shown, then removed
 - 8 Filling from QRZ the one thing that uses the network
 - 9 The POTA command one park or ten
 - 10 Saving, and what is guaranteed byte-for-byte
 - 11 Keyboard shortcuts
 - 12 When something looks wrong
 - 13 Licence
-

1 What this is

POTA activators log with contest and digital-mode software — WSJT-X, MSHV, N1MM, FT8CN — none of which knows anything about Parks on the Air. The `.adi` files they produce are valid ADIF but lack the `MY_SIG_INFO` field carrying the park reference, which POTA's uploader needs. Every activation therefore ends in the same manual edit: add a column, put the park reference on every row, save it under a particular filename. An n-fer — one activation inside several parks at once — needs that done once per park, into separate files.

ADIF Editor makes that one command, and around it does the editing a log actually needs: correcting a mistyped callsign, trimming the contacts that were not part of the activation, removing a QSO logged twice, filling in the name and county you would otherwise type by hand.

THE PROMISE UNDERNEATH ALL OF IT

A file you open and save without editing comes back **byte-for-byte identical** to what went in. Not merely field-preserving — identical, down to field-name casing, zero-length fields, line endings and header bytes. Vendor extensions and fields the app has never heard of survive untouched, because nothing it does not understand is ever rewritten. Section 10 covers the single exception.

What it is not

It is not a logger: no radio control, no clock, no QSO entry form, no logbook of its own. It edits files that loggers produce. It never uploads anything — you upload to POTA yourself. It reads and writes one format, ADI, and no other: no ADX, Cabrillo, CSV or JSON. There is no telemetry, no analytics, no crash reporting and no update check.

It makes exactly one kind of network connection, and only when you ask for it by name: a QRZ callsign lookup (section 8). With no QRZ credentials stored, the app never opens a connection at all.

2 Installing

ADIF Editor needs **macOS 14 or later**. It is a universal build and runs natively on both Apple Silicon and Intel.

Download the `.dmg`, open it, and drag **ADIF Editor** onto the **Applications** shortcut beside it.

The first launch will be blocked

It will, and that is expected — it is not a sign that anything is wrong with the download. ADIF Editor is ad-hoc signed and **not notarized**. Notarizing requires a paid Apple Developer account, which this project does not have, so macOS treats the app as coming from an unidentified developer and refuses to open it the first time.

There are three ways past it. Pick one.

1. Open Anyway. Double-click the app and dismiss the warning. Open **System Settings** ▶ **Privacy & Security**, scroll to **Security**, and click **Open Anyway** beside the message naming ADIF Editor. Confirm, and authenticate if asked. You do this once.

2. Remove the quarantine flag, if you would rather use the Terminal:

```
xattr -dr com.apple.quarantine "/Applications/ADIF Editor.app"
```

3. Build it from source. An app you build yourself is not quarantined, so none of this applies to it. The repository's `README.md` has the two commands.

NOT THE OLD ADVICE

Control-clicking an app and choosing **Open** used to bypass this. Apple removed that route in macOS 15 for apps Gatekeeper blocks. If you find instructions saying otherwise — including in older projects of mine — they are out of date. Use **Open Anyway**.

3 Opening a log

Open a log with `⌘O`, by dragging it onto the app's icon, or by double-clicking an `.adi` file once ADIF Editor is the app macOS associates with it. Recently opened logs are under **File ▶ Open Recent**.

The extension does not matter

FT8CN writes ADIF content to files named `.txt`. Other programs use `.adif`, or no extension at all. ADIF Editor judges a file by what is inside it and opens what it is given, so there is nothing to rename first. The screenshots in this manual are of an FT8CN log named `FT8CN6469053684847039306.txt`, opened exactly as it came off the phone.

If the file has problems

Real logs are sometimes malformed — a truncated download, a hand edit that went wrong, a program that miscounts the length of a field holding accented characters. ADIF Editor recovers what it can, opens the file anyway, and tells you what it had to do in a banner under the title bar. Section 10 explains what those messages mean.

There is exactly one problem it will not open past: a file that is not valid UTF-8. Guessing at the encoding would silently change the bytes of a log you did not edit, so instead it refuses and names the byte offset of the first invalid sequence.

4 The window

One window per log. Every QSO is a row; every ADIF field in the file is a column.

#	CALL	QSL_RCVD	QSL_MANUAL	GRIDSQUARE	MODE	RST_SENT	RST_RCVD	QSO_DATE	TIME_ON	QSO_DATE_OFF	TIME
1	W9AV	N	N	EN53	FT8	-6	-8	20260209	222045	20260209	222111
2	KJ5CTX	N	N	EM10	FT8	-6	-8	20260209	222345	20260209	222411
3	K7ETC	N	N	DM38	FT8	-16	-3	20260209	222845	20260209	222911
4	N7KB	N	N	DM43	FT8	-8	-8	20260209	223600	20260209	223658
5	N5PEL	N	N	EM15	FT8	-12	-12	20260209	223700	20260209	223758
6	VA5KEN	N	N	DO70	FT8	-15	-15	20260209	223830	20260209	223900
7	KJ7EP	N	N	DN27	FT8	-6	-2	20260209	223930	20260209	224000
8	K6US	N	N	DM06	FT8	-5	-4	20260213	222730	20260213	222800
9	KF9UP	N	N		FT8	-16	-6	20260213	222930	20260213	223000
10	AA5HH	N	N	EM21	FT8	-16	-11	20260213	223230	20260213	223300
11	K3RWH	N	N	DM43	FT8	-10	-7	20260213	223330	20260213	223400
12	K4XP	N	N	EM84	FT8	-30	-4	20260213	223500	20260213	223530
13	N0AN	N	N	EN22	FT8	-8	-11	20260213	225215	20260213	225240
14	W5WLG	N	N	EM12	FT8	-27	-14	20260225	175045	20260225	175111
15	KJ7WLL	N	N	CN85	FT8	-12	-9	20260225	180000	20260225	180000
16	W7QF	N	N	CN84	FT8	-7	-6	20260225	180130	20260225	180200
17	N7KB	N	N	DM43	FT8	-18	-3	20260225	180230	20260225	180300
18	KB7SAT	N	N	DM26	FT8	-24	-10	20260225	180400	20260225	180430
19	KB0UZZ	N	N	EN34	FT8	-10	-9	20260225	180600	20260225	180630
20	KZ2T	N	N		FT8	-18	-15	20260312	005945	20260312	010040
21	N7LAN	N	N	CN84	FT8	-18	-13	20260312	010145	20260312	010211

Figure 1. A 66-QSO FT8CN log. The row numbers on the left are the grid's own count, and are what the duplicate and QRZ sheets refer to. The three toolbar buttons at the top right are the whole of the app's commands: POTA, Dedupe and QRZ.

Why there are so many columns, and why two files differ

The columns are not a fixed list ADIF Editor decided on — they are the fields your logger actually wrote. A file from FT8CN carries different fields from one written by WSJT-X, so the two open with different columns, and neither is wrong. Fields the ADIF specification has never heard of get a column too, including a logger's private `APP_*` extensions.

A field takes the place it holds in the first QSO that carries it, and once a column has been placed it does not move. That matters more than it sounds: it means clearing a cell in the first row does not shuffle its column off to the far right.

An empty cell means the field is absent from that QSO, or present and empty. Both are shown the same way, and both are preserved when you save.

5 Editing

Cells

Click a row to select it, then click the cell again to edit it. Type, and press `Return` or `Tab` to commit — or `Escape` to abandon what you typed. Every edit is undoable with `⌘Z`, all the way back to the file you opened.

CALL	QSL_RCVD	QSL_MANUAL	GRIDSQUARE	MODE	RST_SENT	RST_RCV
W9AV	N	N	EN53	FT8	-6	-8
KJ5CTX	N	N	EM10	FT8	-6	-8
K7ETC	N	N	DM38	FT8	-16	-3
N7KB	N	N	DM43	FT8	-8	-8
N5PEL	N	N	EM15	FT8	-12	-12
VA5KEN	N	N	DO70	FT8	-15	-15
KJ7EP	N	N	DN27	FT8	-6	-2
K6US	N	N	DM06	FT8	-5	-4

Figure 2. A `GRIDSQUARE` cell open for editing.

Clearing a cell removes that field from the QSO — that is what an empty field means when the file is written back. This is the one place the app treats "empty" as a deliberate act rather than something to preserve, because you did it on purpose.

Sorting

Click any column header to sort on it; click again to reverse. Sorting is a view of the log, not an edit to it — but it is the log's order that gets written when you save, so a sort you leave in place is a sort you will find in the file. Sorting on `QSO_DATE` and `TIME_ON` is the usual way to put a log that came out of several sessions back into the order it happened.

Selecting and deleting QSOs

Click a row to select it; shift-click to extend the selection, or command-click to add one row at a time.

`⌘A` selects everything.

FT8CN6469053684847039306.txt											
POTA Dedupe QRZ											
#	CALL	QSL_RCVD	QSL_MANUAL	GRIDSQUARE	MODE	RST_SENT	RST_RCVD	QSO_DATE	TIME_ON	QSO_DATE_OFF	TIME
1	AA5AH	N	N	EM12	FT8	-7	-12	20260813	144745	20260813	14481
2	AA5HH	N	N	EM21	FT8	-16	-11	20260213	223230	20260213	22330
3	AA5HH	N	N	EM21	FT8	-19	-12	20260714	155100	20260714	15513
4	AA5HH	N	N	EM21	FT8	-8	-9	20260813	145515	20260813	14561
5	AK7LV	N	N	FT8	FT8	-8	-6	20260714	153800	20260714	15383
6	K1SM	N	N	FN42	FT8	-19	-15	20260609	204515	20260609	20451
7	K2TQC	N	N	FN13	FT8	5	6	20260609	203945	20260609	20393
8	K3RWH	N	N	DM43	FT8	-10	-7	20260213	223330	20260213	22340
9	K4BTR	N	N	EM57	FT8	-18	-1	20260312	011400	20260312	01143
10	K4COM	N	N	EM65	FT8	-17	-21	20260312	010545	20260312	01061
11	K4XP	N	N	EM84	FT8	-30	-4	20260213	223500	20260213	22353
12	K6US	N	N	DM06	FT8	-5	-4	20260213	222730	20260213	22280
13	K6WDE	N	N	DM37	FT8	-1	-22	20260714	155330	20260714	15540
14	K7ETC	N	N	DM38	FT8	-16	-3	20260209	222845	20260209	22291
15	KB0UZZ	N	N	EN34	FT8	-10	-9	20260225	180600	20260225	18063
16	KB7SAT	N	N	DM26	FT8	-24	-10	20260225	180400	20260225	18043
17	KB7SAT	N	N	DM26	FT8	-16	-11	20260714	155900	20260714	15593
18	KC8YOM	N	N	EM79	FT8	-5	-13	20260409	142415	20260409	14244
19	KD0OIX	N	N	EN16	FT8	-5	6	20260609	202245	20260609	20231
20	KF9UP	N	N	FT8	FT8	-16	-6	20260213	222930	20260213	22303
21	KG1GEM	N	N	EM13	FT8	-28	-21	20260312	011230	20260312	01130

Figure 3. Sorted on **CALL** — note the sort arrow in the header — with a contiguous block of QSOs selected. Sorting first is what makes a single drag enough.

The **Delete** key removes the selected QSOs, and so does **Edit ▶ Delete**. Both are undoable. Trimming a log down to the contacts that were part of one activation is normal preparation for an upload, and it is the reason the output filename can take the earliest date in the file without having to guess anything: you have already removed the QSOs that do not belong.

Cut, copy and paste move whole QSOs

Selection in the grid is whole rows, so **⌘X**, **⌘C** and **⌘V** operate on whole QSOs rather than on a rectangle of cells. Pasted QSOs land below the selection, or at the end if nothing is selected — so pasting one log into another appends it.

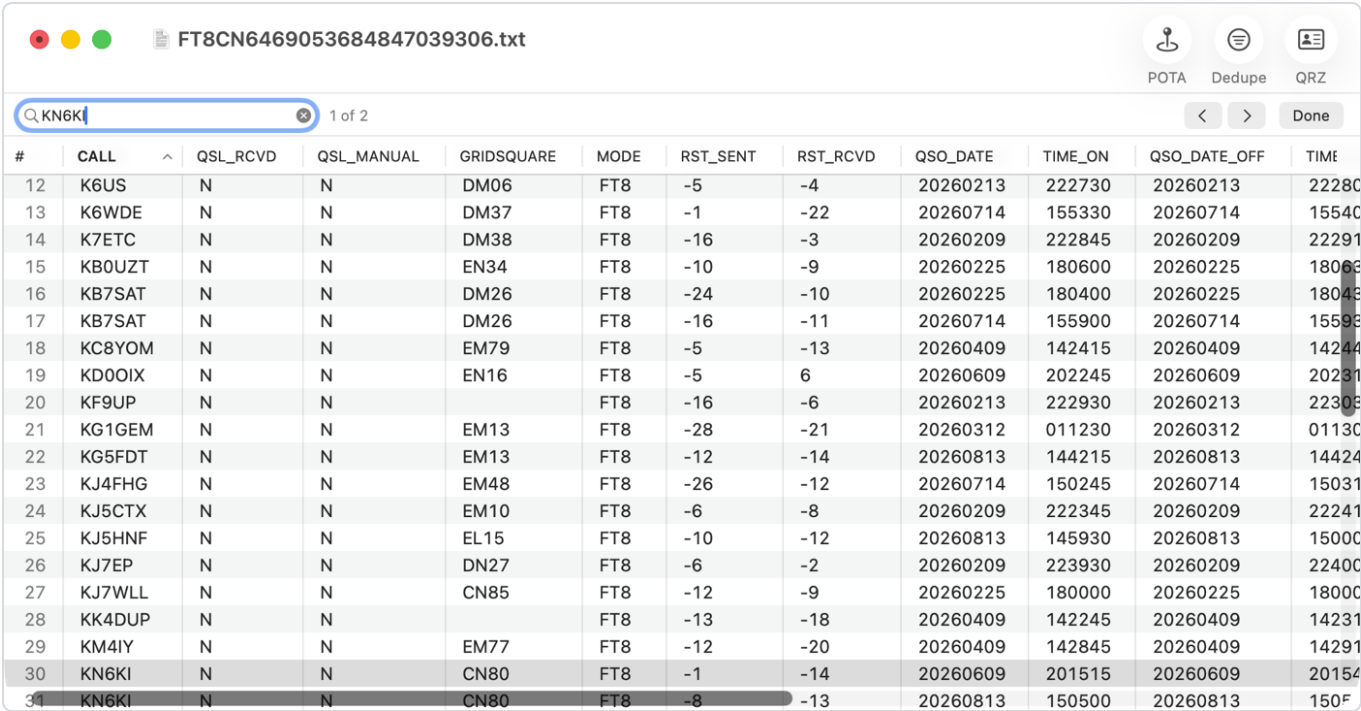
What goes on the clipboard is ADIF text. QSOs copied here paste into another log, or into a text editor, and come back in as themselves.

WHILE A CELL IS OPEN FOR EDITING

Cut, copy and paste apply to the *text* you are editing, which is what you would expect from any text field. They only move QSOs when the grid itself has focus.

6 Finding

%F opens the find bar. Type, and the grid moves its selection to the first QSO containing what you typed. **%G** steps to the next match and **↑%G** to the previous, wrapping around the ends; the arrows beside the field do the same. **Escape** or **Done** closes it.



The screenshot shows a software window titled "FT8CN6469053684847039306.txt". At the top right are icons for "POTA", "Dedupe", and "QRZ". Below these is a search bar containing "KN6KI" with a "1 of 2" match count to its right. Navigation arrows and a "Done" button are also present. Below the search bar is a table with 13 columns: #, CALL, QSL_RCVD, QSL_MANUAL, GRIDSQUARE, MODE, RST_SENT, RST_RCVD, QSO_DATE, TIME_ON, QSO_DATE_OFF, and TIME. The table contains 31 rows of QSO data. Row 30 is highlighted, showing a match for the search term "KN6KI".

#	CALL	QSL_RCVD	QSL_MANUAL	GRIDSQUARE	MODE	RST_SENT	RST_RCVD	QSO_DATE	TIME_ON	QSO_DATE_OFF	TIME
12	K6US	N	N	DM06	FT8	-5	-4	20260213	222730	20260213	222800
13	K6WDE	N	N	DM37	FT8	-1	-22	20260714	155330	20260714	155400
14	K7ETC	N	N	DM38	FT8	-16	-3	20260209	222845	20260209	222910
15	KB0UZZ	N	N	EN34	FT8	-10	-9	20260225	180600	20260225	180630
16	KB7SAT	N	N	DM26	FT8	-24	-10	20260225	180400	20260225	180430
17	KB7SAT	N	N	DM26	FT8	-16	-11	20260714	155900	20260714	155930
18	KC8YOM	N	N	EM79	FT8	-5	-13	20260409	142415	20260409	142440
19	KD0OIX	N	N	EN16	FT8	-5	6	20260609	202245	20260609	202310
20	KF9UP	N	N		FT8	-16	-6	20260213	222930	20260213	223000
21	KG1GEM	N	N	EM13	FT8	-28	-21	20260312	011230	20260312	011300
22	KG5FDT	N	N	EM13	FT8	-12	-14	20260813	144215	20260813	144240
23	KJ4FHG	N	N	EM48	FT8	-26	-12	20260714	150245	20260714	150310
24	KJ5CTX	N	N	EM10	FT8	-6	-8	20260209	222345	20260209	222410
25	KJ5HNF	N	N	EL15	FT8	-10	-12	20260813	145930	20260813	150000
26	KJ7EP	N	N	DN27	FT8	-6	-2	20260209	223930	20260209	224000
27	KJ7WLL	N	N	CN85	FT8	-12	-9	20260225	180000	20260225	180030
28	KK4DUP	N	N		FT8	-13	-18	20260409	142245	20260409	142310
29	KM4IY	N	N	EM77	FT8	-12	-20	20260409	142845	20260409	142910
30	KN6KI	N	N	CN80	FT8	-1	-14	20260609	201515	20260609	201540
31	KN6KI	N	N	CN80	FT8	-8	-13	20260813	150500	20260813	150530

Figure 4. The find bar, with the match count beside the field. Find moves the selection; it never hides a row and never changes the log.

Every column is searched at once. A callsign, a park reference, a grid square and a comment are all just text you half-remember, and having to say which column it lives in before you can look for it is a question you should not have to answer. The match is case- and accent-insensitive. Field *names* are not searched — typing **CALL** finds the QSO whose comment mentions a call, not every QSO in the log.

TO ACT ON A GROUP OF QSOS, SORT — DO NOT SEARCH

Find moves to one QSO at a time, so it cannot help you delete every contact someone else made from your callsign. Sorting can: click the **OPERATOR** column header and the QSOs that are not yours become one contiguous block, which you can select in a single drag and delete in one keystroke. The same trick works for a band, a mode, or a date. This is deliberately the only mechanism — a filter that hides rows was built, tried, and removed as far more machinery than the job needs.

7 Duplicates

Edit ▶ Find Duplicates..., `%D`, or the **Dedupe** toolbar button. A log that has been exported twice, or that has had a session appended to it by hand, ends up with the same QSO in it more than once. POTA will reject the repeat, and it is easier to remove it here.

The first sheet asks what "the same" means for this log. The fields ticked by default are `QSO_DATE`, `CALL`, `BAND` and `MODE` — what an activator almost always means by a duplicate — and only fields the file actually carries are offered.

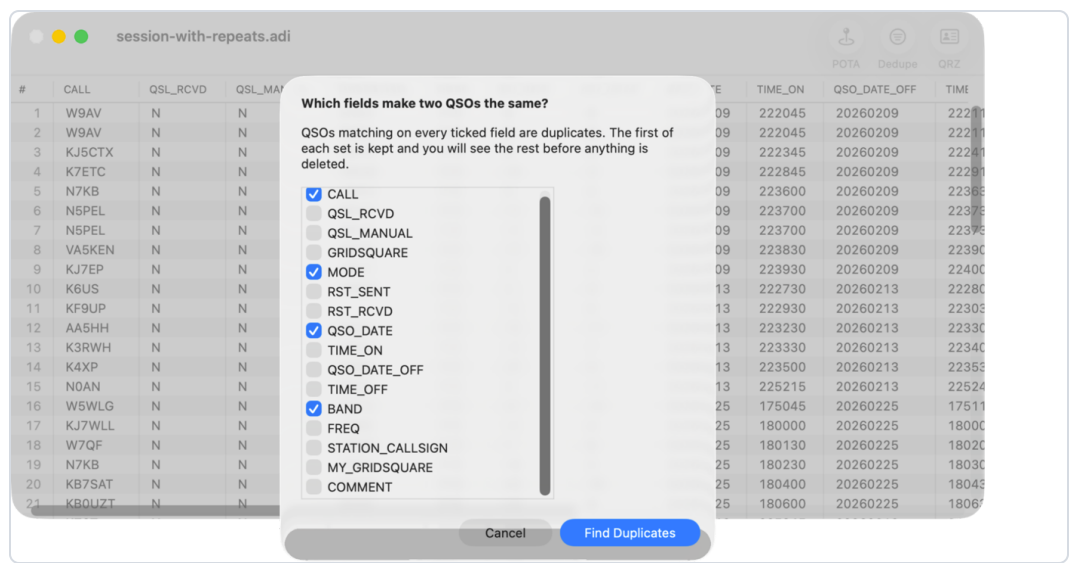


Figure 5. Choosing what counts as a duplicate. Only fields this log actually carries are offered, so the list is different for a file from a different logger.

The second sheet is the point of the whole command: **every QSO that would be deleted is listed before any of them is**, one line each, showing the row number, the values that matched, and the row it duplicates.

#12 W1ABC · 20260209 · 20m · FT8 — same as #3

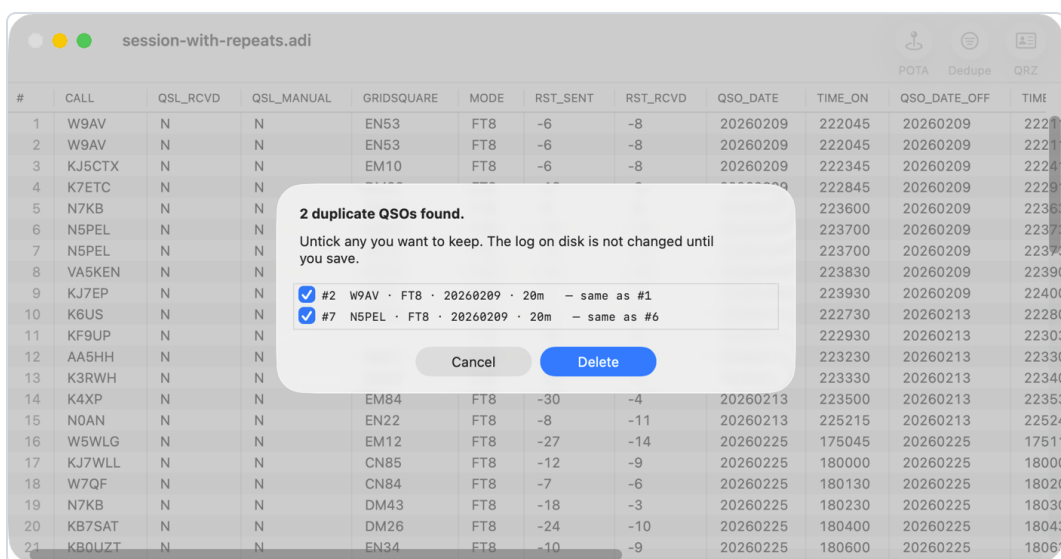


Figure 6. Nothing has been deleted at this point. Both repeats are named, with the QSO each one duplicates, and either can be unticked and kept.

They arrive all ticked, because agreeing with the whole list is the common case. Untick anything you would rather keep. The first QSO of each matching set is never offered for deletion — only the repeats. **Delete** removes what is still ticked, in one undoable step, and nothing reaches the file on disk until you save.

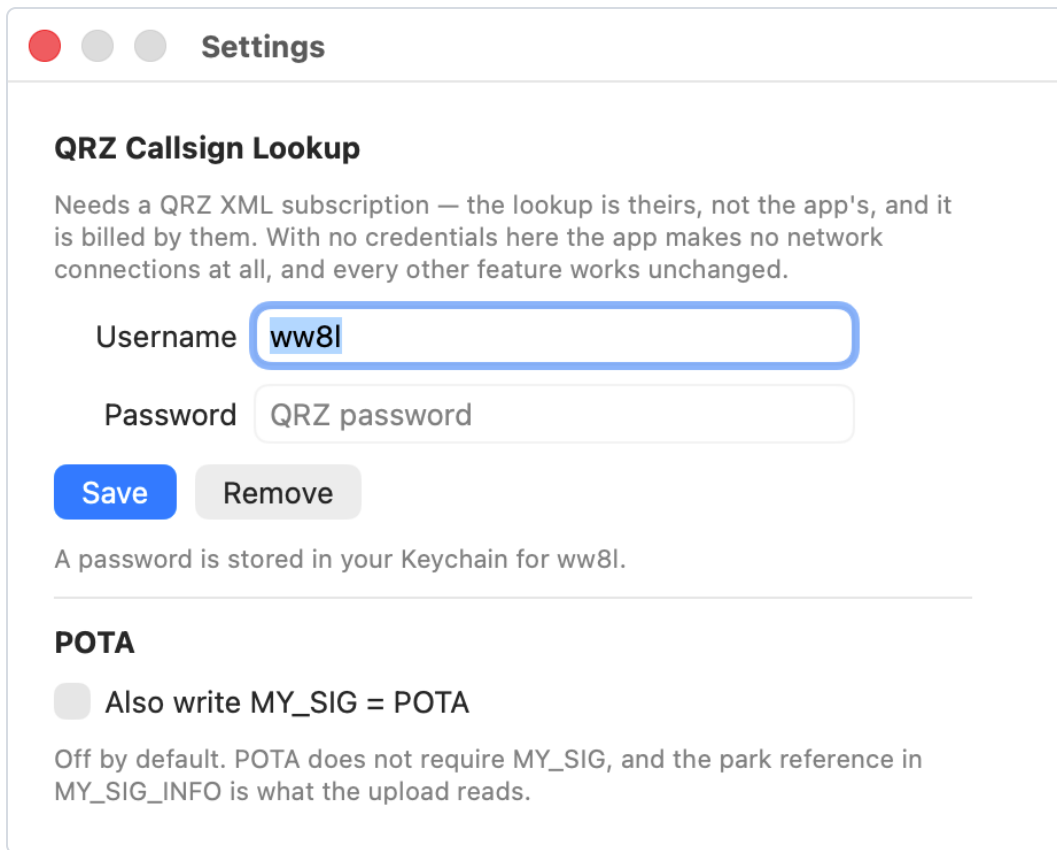
8 Filling from QRZ

Edit ▶ Fill from QRZ..., , or the **QRZ** toolbar button. It fills in the station details you would otherwise type by hand: name, town, state, county, grid square, country, DXCC, and CQ and ITU zones.

This is the only feature that uses the network, and it needs a **QRZ XML subscription** — the lookup is QRZ's service, billed by them, not something this app provides.

Credentials

Enter your QRZ username and password once, in **ADIF Editor ▶ Settings...** (,).



Settings

QRZ Callsign Lookup

Needs a QRZ XML subscription — the lookup is theirs, not the app's, and it is billed by them. With no credentials here the app makes no network connections at all, and every other feature works unchanged.

Username

Password

Save **Remove**

A password is stored in your Keychain for ww8l.

POTA

☐ Also write MY_SIG = POTA

Off by default. POTA does not require MY_SIG, and the park reference in MY_SIG_INFO is what the upload reads.

Figure 7. Settings, in its entirety. The password goes to the Keychain; the username is an ordinary preference. **Remove** takes the app offline again.

WHERE THE PASSWORD GOES

Into your Keychain, and nowhere else — never a preferences file, never a log, never the app's own storage. **Remove** deletes it, and the app is offline again the moment it is gone: with no credentials stored it does not so much as attempt a connection. One callsign at a time goes out. Your log never does.

Running a lookup

Select the QSOs you want filled, or select nothing to cover the whole log — the confirmation sheet says which of those it is about to do, and how many callsigns that means, because looking up four hundred QSOs when you meant four is the kind of surprise that costs an API quota.

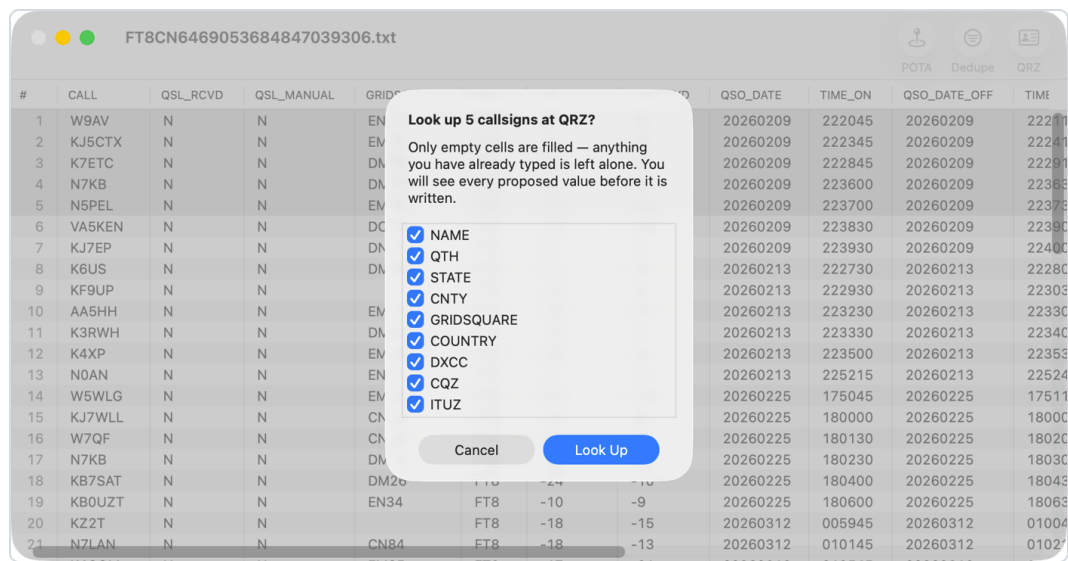


Figure 8. Five QSOs were selected, so five callsigns are looked up — with nothing selected the sheet would say so and cover the whole log. The ticked fields are remembered.

Tick the fields you want filled. The choice is remembered for next time. A progress sheet counts through the callsigns while it runs and can be cancelled part way; what has already been fetched is still offered.

Then, as with duplicates, every proposed change is shown before any of it is applied, one line per cell:

#12 W1ABC GRIDSQUARE → FN32aa

Untick anything you disagree with, and **Fill** applies the rest as a single undoable edit.

Callsigns QRZ has no record of are named in the review sheet rather than passed over in silence, as is any error that stopped the batch early. Two fields QRZ returns are deliberately not offered: latitude and longitude, because the grid square already carries the position at the precision a log needs, and the street address, because a home address is not log data and should not end up in a file you upload.

EMPTY CELLS ONLY
The lookup never overwrites a value that is already there. If you have typed a name or corrected a grid square, QRZ's version of it is not offered and cannot replace yours. Corrections are yours to make by hand — the app fills blanks and creates columns, and does not second-guess data you entered.

9 The POTA command

POTA ▶ Stamp Park Reference..., **%P**, or the POTA toolbar button. This is the reason the app exists.

Type one park reference and you get a file stamped with it. Type several, separated by commas, and you get one file per park — each holding *every* QSO, because being inside three parks at once means each contact counts for all three. It is the same command either way; the number of references you type is what decides.

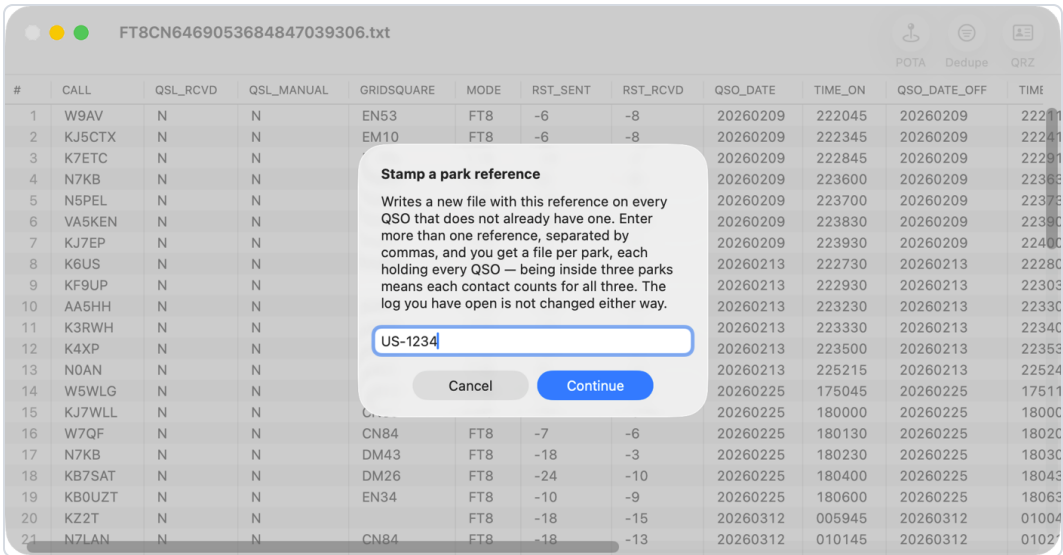


Figure 9. One reference or several, separated by commas. This is the only place the two cases differ.

References are validated leniently: anything shaped unlike **US-1234** gets a question rather than a refusal, because park numbering is not this app's business to be certain about.

Next you choose a destination folder, and then confirm the filenames. They are proposed in POTA's convention:

WW8L@US-1234 20260209.adf	
PART	WHERE IT COMES FROM
WW8L	STATION_CALLSIGN, falling back to OPERATOR — and a portable callsign's / becomes a -, since a slash is a path separator
US-1234	the park reference you typed
20260209	the earliest QSO_DATE in the log — which is why trimming the log first matters

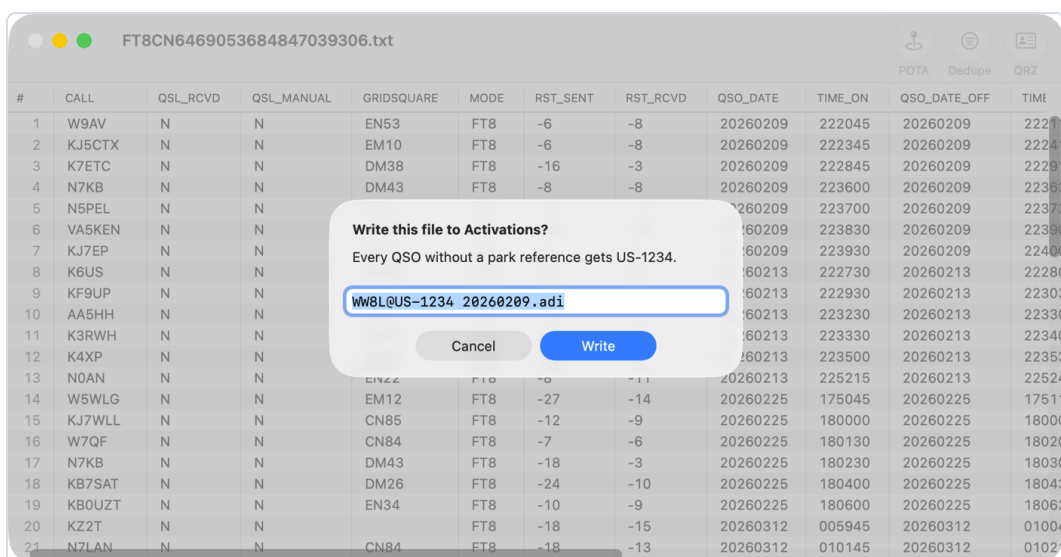


Figure 10. The proposed name, editable before anything is written, and the destination named in the question.

Every name is editable before anything is written. If the log carries no callsign at all, the name is proposed without one and the sheet says so, rather than putting a second question in front of you. If a name would collide with a file already in that folder, you are asked before it is replaced; if it would collide with the log you have open, or with another of the files about to be written, the app refuses outright and sends you back to the names.

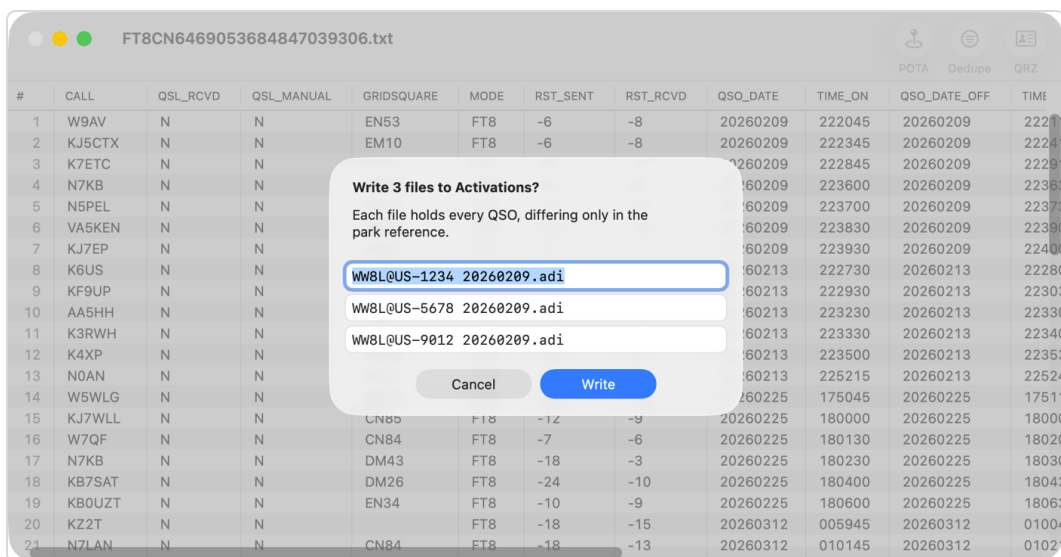


Figure 11. A three-park n-fer: three files, each holding every QSO, differing only in the park reference.

THE LOG YOU OPENED IS NEVER TOUCHED

Every one of these outputs is a new file. The log that came off the radio is not modified, not rewritten, and not even marked as changed — whatever you do here, it is still on disk exactly as it was. That is why one park and ten take the identical path.

QSOs that already have a park reference

If some QSOs already carry MY_SIG_INFO , you are asked once whether to replace those values in the output or keep them.

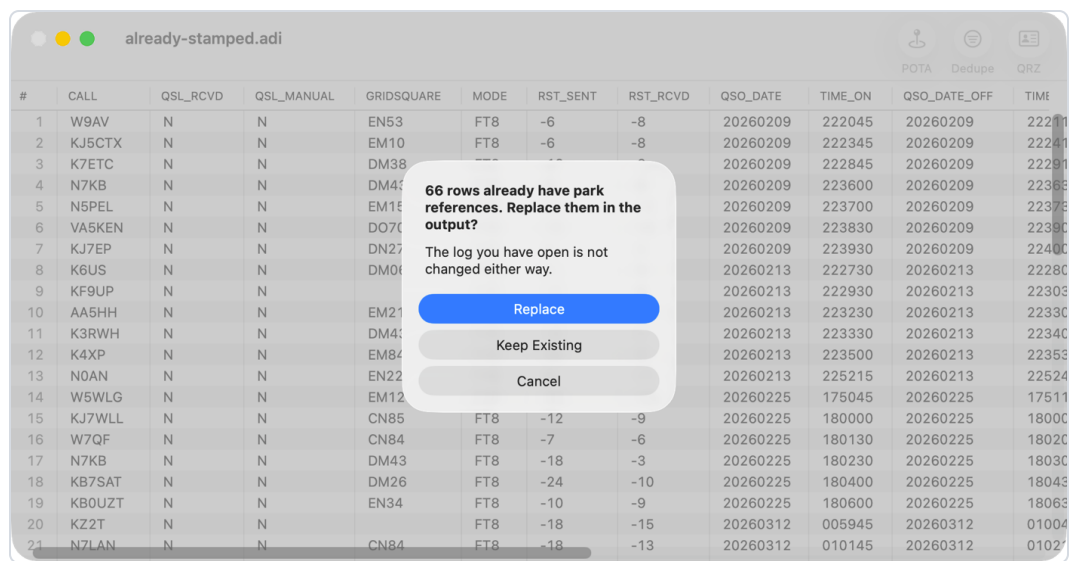


Figure 12. Asked only when there is something to overwrite — and the log you have open is not changed whichever you pick.

Empty cells are filled without comment; existing values are never replaced silently. This is the one place in the app where an automated command writes over a value you might have put there, and it is permitted only because the outputs are new files and it always asks first.

MY_SIG

Settings has one POTA option: whether to also write MY_SIG = POTA alongside the reference. It is off by default. POTA does not require it — MY_SIG_INFO is what the upload reads.

10 Saving, and what is guaranteed

`%S` saves in place; **File ▶ Save As...** writes elsewhere; **Revert to Saved** throws away everything since the last save. There is no autosave, and nothing in the app writes to the file you opened except Save. Closing a window with unsaved edits asks first.

Byte-for-byte

Open a file and save it without editing anything, and what lands on disk is identical to what came in — the same bytes, not merely the same fields. That covers things a log editor could reasonably be expected to tidy up and deliberately does not: the byte-order mark if there was one, CRLF line endings if the file used them, a missing header, the casing of every field name as it was spelled, zero-length fields, ADIF type indicators, and the order fields appear in within each QSO.

Fields the app has never heard of are covered by the same rule, because nothing it does not understand is ever rewritten. A logger's private `APP_*` extensions and `USERDEF` definitions come back out exactly as they went in.

THE ONE EXCEPTION

A field whose declared length is wrong is written back *correct*. Copying the wrong number out again would be preserving corruption rather than preserving data. Every such case is reported in the banner, so a file that changes on save always told you why first.

What the banner is telling you

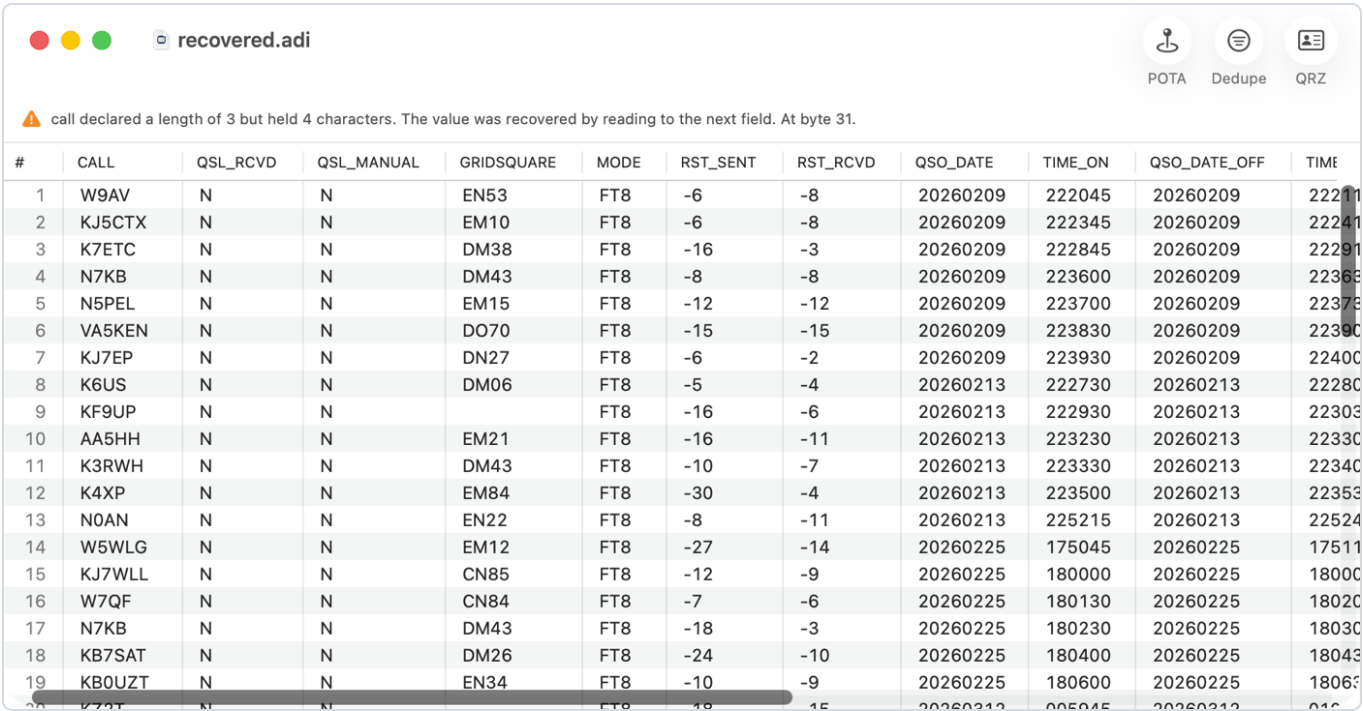


Figure 13. The banner names the first problem and the byte it was found at, and counts the rest. It is advisory — the file opened, the QSOs are all there, and nothing is waiting on you.

WHAT IT SAYS	WHAT HAPPENED
declared a length of <i>n</i> but held <i>m</i> characters	The field's length was wrong. The value was recovered by reading up to the next field, and will be written back with a corrected length.
matched only when read as bytes rather than characters	The logger counted bytes where ADIF counts characters — normal when a value holds accented characters or an emoji. The value is intact.
which is not a number	The length was not numeric at all. The field was kept and its value recovered.
Unexpected text between fields	Something that belongs to no field — a stray separator, a note someone typed in. It is left exactly where it was.
characters could not be read as ADIF and were skipped	A run the parser could not make sense of. It resynchronized at the next well-formed tag rather than giving up on the file.
ends before the last QSO's <EOR>	A truncated file — an interrupted download or copy. The final QSO is kept with the fields it had, rather than discarded.
carries <i>field</i> twice	One QSO has the same field twice. Both are kept and both will be written back, but the grid shows and edits only the first — so clearing that cell reveals the second rather than emptying it.

11 Keyboard shortcuts

KEY	DOES
⌘O	Open a log
⌘S	Save
⇧⌘S	Save As...
⌘W	Close the window
⌘Z	Undo
⇧⌘Z	Redo
⌘X ⌘C ⌘V	Cut, copy and paste whole QSOs — or text, while editing a cell
⌘A	Select every QSO
Delete	Delete the selected QSOs
⌘⌫	Delete the selected QSOs, from the Edit menu
⌘F	Find
⌘G / ⇧⌘G	Find next / previous
⌘D	Find duplicates
⌘L	Fill from QRZ
⌘P	Stamp a park reference
⌘,	Settings
Return / Tab	Commit the cell you are editing
Escape	Abandon the cell you are editing, or close the find bar

WHY DELETE IS ON ⌘⌫ IN THE MENU

A bare ⌫ as a menu shortcut would be caught by the app before it ever reached a cell you were editing, and backspace would stop deleting characters. So the menu item carries ⌘⌫, while the bare Delete key still deletes QSOs whenever the grid itself has focus — which is the only time it could mean that.

12 When something looks wrong

macOS says the app is damaged, or from an unidentified developer

Expected on first launch: the app is not notarized. Section 2 has the three ways past it. Nothing is wrong with the download.

My log is a `.txt` file

Open it anyway. FT8CN and others write ADIF content under whatever extension they like, and the app judges by content. There is no need to rename anything.

The QRZ lookup filled nothing

Most often because there was nothing to fill: the lookup only writes into empty cells, so a log that already has names and grid squares has nothing for it to do — and the sheet says exactly that. Otherwise, check that the fields you wanted are ticked, that the QSOs you meant are selected, and that Settings shows credentials as stored. QRZ having no record of a callsign is reported by name.

A QSO will not vanish when I clear its cell

If the banner mentioned a field appearing twice in one QSO, the grid is showing the first of the two. Clearing it reveals the second rather than emptying the cell. Both are in the file and both are preserved.

The window opened tiny, or in the wrong place

Windows remember the size and position you last used, shared across logs. If a remembered size no longer fits — a display that changed, or an external monitor that is no longer attached — the app notices and sizes the window to the log instead. Resize it once and the new size is what the next log gets.

The file will not open at all

The only fatal problem is a file that is not valid UTF-8. The message names the byte offset of the first bad sequence. That usually means a genuinely corrupt file or one in some older single-byte encoding; converting it with `iconv` and opening the result is the way through.

I sorted the log and now the file is in a different order

Sorting is a view, but it is the view that gets written. `⌘Z` undoes a sort like anything else, and reverting or reopening without saving leaves the file untouched.

13 Licence

ADIF Editor is free software under the **GNU General Public License, version 3 or later**. You may use, study, modify and share it freely. If you distribute a modified version you have to make your source available under the same terms, so that it stays open for whoever comes after you.

As with any licence of this kind, the program comes with **no warranty of any kind**, to the extent permitted by law; you use it at your own risk. The exact terms are in sections 15 to 17 of the licence, which ships with the source.

This is a clean-room implementation written against the public ADIF specification at adif.org. It is not derived from, affiliated with, or endorsed by any other ADIF editing tool.

ADIF Editor v26.8.18 · Copyright © 2026 Tim Annable, WW8L · GPL-3.0-or-later

This manual was generated from `docs/manual/manual.html` on 22 August 2026. Screenshots are of v26.8.18 running on macOS.