

MacHuna v2.0.0 — User Manual

Broadcast Media Format Converter

DNS Vision Limited — For Vision Mixers, TDs, and Support Engineers

Contents

1. [Overview](#)
 2. [Installation](#)
 3. [The Window](#)
 4. [Converting Files](#)
 5. [Kahuna SWS Output](#)
 6. [K-Frame EIF — Native K-Frame Format](#)
 7. [Extraction Outputs — K-Frame and Sony MVS](#)
 8. [TGA Sequences](#)
 9. [Audio](#)
 10. [The Player](#)
 11. [Menus and Keyboard](#)
 12. [Updates and Getting in Touch](#)
 13. [Large File Support \(>4GB\)](#)
 14. [SWS Format Reference](#)
 15. [Troubleshooting](#)
 16. [Known Limitations](#)
 17. [Authors](#)
-

1. Overview

MacHuna is a macOS application for translating broadcast media assets between formats. It converts video clips, TGA sequences, and still images to Grass Valley Kahuna `.SWS` format, reads and writes Grass Valley K-Frame `.EIF` native clips, and extracts `.SWS` files back to standard formats for use on K-Frame and Sony MVS desks.

v2.0 is MacHuna rewritten as a native Mac application. The interface is new throughout. **The conversion engine is not** — it is the same code, unchanged, so every file MacHuna writes is identical to the one it wrote before. If you have SWS files on a desk that came out of an earlier version, nothing about them is affected.

1.1 What MacHuna Does

Direction	From	To
Kahuna SWS	MOV, MP4, MXF, MKV, AVI, TGA sequences, PNG, BMP, JPG	.SWS (Grass Valley Kahuna)
K-Frame EIF	MOV, MP4, MXF, TGA sequences, .SWS	.EIF (Grass Valley K-Frame) (<i>UNCONFIRMED on hardware</i>)
K-Frame EIF → SWS	.EIF	.SWS (Grass Valley Kahuna) (<i>UNCONFIRMED on hardware</i>)
K-Frame EIF → TGA	.EIF	32-bit RGBA TGA sequence (<i>UNCONFIRMED on hardware</i>)
K-Frame TGA	.SWS , .EIF	32-bit RGBA TGA sequence (<i>UNCONFIRMED on hardware</i>)
Sony TGA	.SWS , .EIF , TGA sequences	32-bit RGBA TGA sequence (Sony MVS naming)
TGA Sequence	TGA sequences, video clips	32-bit RGBA TGA sequence (i?p standards conversion)
QuickTime MOV	.SWS , .EIF , TGA sequences	ProRes 4444 .mov with alpha, and audio where the source has it

1.2 Supported Standards

All seven standards below have been confirmed against K-Watch reference files and verified on a live Grass Valley Kahuna mainframe.

Standard	fps	Region
1080i/50	25	UK / Europe
1080i/59.94	29.97	USA / Japan
1080i/60	30	—
1080p/25	25	UK / Europe
1080p/50	50	UK / Europe
1080p/59.94	59.94	USA / Japan
1080p/60	60	—

1.3 What is New in v2.0

- **Drag files straight onto MacHuna** from anywhere in the Finder

- **Drag finished files straight out again**, to wherever they need to go, without opening the destination folder
- **Double-click a finished file to watch it**, without leaving the window
- **Drag any clip onto the player** to open it, converted or not
- **A timeline you can drag** in the player, and playback that stays smooth
- **Reorder a batch from the keyboard**, with $\text{⌘}↑$ and $\text{⌘}↓$
- **A full menu bar**, with a keyboard shortcut for everything that matters

NOTE MacHuna converts and extracts files on your Mac. It does not upload to a mainframe over the network and it does not synchronise projects — getting the finished files onto the desk is done however you do it now.

2. Installation

MacHuna is distributed as a self-contained `.app` bundle.

Download it from dnsvision.tv/machuna. That page always carries the current release, the release notes and this manual.

2.1 Upgrading from v1.x

v2.0 replaces the previous version in place. It has the same name and the same bundle identifier, so dragging it into Applications over the old one is all that is required. Your Dock icon keeps working.

Your settings come with you. The first time v2.0 runs it reads the settings the previous version left behind — destination folder, output format, video standard, field order and the conversion options — so nothing has to be set up again. It reads that file once and never writes to it, so an older copy of MacHuna kept elsewhere is unaffected.

2.2 First Launch

MacHuna is not signed with an Apple Developer certificate, so the first time you open it macOS says it cannot verify the developer and refuses. This is expected. Three steps, once per machine:

1. Move **MacHuna.app** to your **Applications** folder and double-click it. macOS shows a warning and will not open it. Click **Done**
2. Open **System Settings** ▸ **Privacy & Security** and scroll down. There is a line about MacHuna being blocked, with an **Open Anyway** button. Click it and confirm with Touch ID or your password
3. MacHuna opens, and opens normally from then on

NOTE Older instructions said to right-click the app and choose **Open**. That no longer works on current versions of macOS, which removed the shortcut: you have to go through System Settings as above.

Checking which version you have. In Finder, select **MacHuna.app** and press **⌘I**. **MacHuna ▸ About MacHuna** shows it too, and **MacHuna ▸ Check for Updates...** tells you whether it is the current release.

2.3 Settings

Every setting is saved as you change it, and the window keeps its size and position between sessions. There is nothing to save and nothing to lose on a crash.

3. The Window

MacHuna has one window for converting and a separate one for the player.

The conversion window is in two parts, with a bar across the bottom:

Part	What it is for
The list , on the left	One row per item to be converted. What it is, what it will be called, and how it got on
The inspector , on the right	The output format and everything that depends on it, and the destination folder
The convert bar , across the foot	What MacHuna is doing or objecting to, the progress of a batch, and the Convert button

3.1 The List

One row per item. TGA sequences are collapsed to a single row rather than one row per frame.

Column	What it shows
Name	The source file. A TGA sequence also says how many files it stands for
Source	What MacHuna found: the standard, the frame count, and for an EIF the frame rate
Number or Clip name	What this item will be called. The heading changes with the output format
Saves as	The filename that will be written. After converting, the finished file appears here as something you can drag
Status	Queued, Converting, Done, or what went wrong

Below the list is a narrow gutter holding **+** and **–**, which add files and remove the selected rows, and the **Number sequentially** tickbox. At the right of that gutter is a count of what is loaded.

Why a minus and not a trash can. **–** takes an item off the list. It does not delete anything: the source file stays exactly where it is, and you can drag it back in. A trash can would say otherwise.

3.2 The Inspector

The inspector shows only the controls that mean something for what you have loaded and what you are making. Choosing a different output format changes which controls appear.

Group	Controls
Output	Format, Video standard, Field order
Conversion	TGA source interlaced, Include audio, Ignore key
Playback	Auto play, Loop play — Kahuna SWS only, because they are flags in the SWS header
Destination	Where files are written, with Change... , Show in Finder and Clear...

Write beside the source files puts the output on the same drive as the sources, in a folder called **MacHuna**, instead of in your chosen destination. It is for the case where someone hands you a memory stick of new wipes: convert them, and the results are already on the stick ready to carry to the desk. See Section 4.5.

Clear... moves everything in the destination folder to the Trash, after asking. It is there because the new way of working means you may never open that folder yourself, and successive batches often want the same numbers. Nothing is deleted outright: it goes to the Trash where you can retrieve it.

3.3 The Convert Bar

The bar reports what is happening, or what is stopping the batch. When a message has more to it than fits on one line, the chevron at the left opens the rest: which rows collide, and what to do about it.

4. Converting Files

The whole job, in four steps:

1. **Get the files in.** Drag them onto the window from the Finder, press **+**, or choose **File ▶ Open...** (⌘O)
2. **Choose the output** in the inspector, and set anything that appears with it
3. **Give each row a number or a name**, or tick **Number sequentially**
4. **Press Convert** (⌘R)

A conversion log is written to the destination folder when the batch completes.

4.1 Getting Files In

Drag them onto the window. Files, folders, or a mixture. This is the quickest way and it works from anywhere in the Finder.

Or press +, or File ▶ Open... (⌘O). The open panel accepts files and folders. For a TGA sequence, pick its **first frame** — MacHuna takes the whole sequence, and dims the other frames so it is clear which one to choose.

Files added to a list that already has something in it are appended, not substituted. Something already on the list is not added twice, and MacHuna says so rather than doing nothing.

4.2 What Can Be Converted Together

MacHuna works out what you have given it and offers only the outputs that make sense for it.

What is loaded	Outputs offered
.SWS files	Kahuna SWS, K-Frame TGA, K-Frame EIF, Sony TGA, QuickTime MOV
.EIF files, or a mix of EIF and SWS	Kahuna SWS, K-Frame TGA, Sony TGA, QuickTime MOV
Video clips	Kahuna SWS, K-Frame TGA, K-Frame EIF, Sony TGA, TGA Sequence
TGA sequences and stills	Kahuna SWS, K-Frame EIF, Sony TGA, TGA Sequence, QuickTime MOV

One batch has to be one kind of job. Material being written **to** a desk format, or desk files being read **back out** — not both at once. Trying to add the wrong kind is refused with an explanation, before anything joins the list, so you are never left unpicking a batch that cannot run.

NOTE — stills go to Kahuna SWS only. A still image is a single frame, not a clip, so MacHuna does not convert stills to K-Frame EIF, Sony TGA or TGA Sequence output. Those outputs need a clip or a TGA sequence.

4.3 Naming and Numbering

Every output is named, and the **Saves as** column shows exactly what will be written before anything is.

Output	What each row takes
Kahuna SWS	A number, 1 to 9999, written as 12.SWS
K-Frame EIF	A slot number, written as 0012.eif
Sony TGA	A four-character clip name, written as a folder of frames
QuickTime MOV	A name, or nothing at all — leave it blank and the MOV is named after its source
K-Frame TGA, TGA Sequence	Nothing. These name their output from the source file

Number sequentially fills the column from the first row down. Change the first row and everything below follows. Untick it and every row is yours to fill in: filling a scattered set of empty slots on a desk is at least as common as numbering a batch straight through.

The setting is remembered between sessions. The **numbers are not** — every batch starts from 1 unless you say otherwise.

Checks before anything is written

Three checks run as you type, and any of them stops the batch:

1. **Nothing may be blank**, or outside the range the format allows
2. **No two rows may share a number or name**
3. **Nothing may collide with the destination folder** — an existing `12.SWS`, whether a file or a split-file folder, an existing `0012.eif`, or a Sony clip-name folder

There is no overwrite option. Offending rows are marked, the convert bar says what is wrong, and Convert stays unavailable until it is fixed. Correcting one row clears its partner's mark too.

MacHuna watches the destination folder while you work, so deleting the offending file in the Finder clears the warning immediately.

4.4 Reordering a Batch

Select one or more rows and use **Edit ▸ Move Up** (⇧⌘↑) or **Move Down** (⇧⌘↓).

With **Number sequentially** on, reordering decides **which file gets which number** — the numbers stay where they are and the files move between them. With it off, each row keeps the number you typed and only the order changes.

4.5 Writing Beside the Sources

Tick **Write beside the source files** in the inspector and the output goes onto the same drive as the sources, in a folder called `MacHuna`, rather than to your chosen destination. Convert a stick of wipes and the results are already on the stick.

The folder it has chosen is shown under the tickbox as soon as files are loaded, so it is never a guess. It is the deepest folder that contains every source: a stick with everything at the top level resolves to the stick itself; sources in `Wipes/A` and `Wipes/B` resolve to `Wipes`.

It is off by default, it is never remembered, and it switches itself off after every conversion. That is deliberate. A destination that outlives the drive it pointed at is a trap, so turning this on is a decision you take once per batch.

MacHuna checks three things before letting the batch run, and says which is wrong in the bar at the foot of the window:

Different drives	The sources are not all on one drive, so there is no single folder beside them. Turn it off and choose a destination
Not writable	The drive is locked, read-only, or otherwise refuses. Turn it off and choose a destination
Not enough room	The bar says how much is free and roughly how much the batch needs. Remove some rows, or turn it off and choose a destination

The space check is worth trusting. MacHuna works out the size from the frame count, the output format and the rate change the conversion will apply, and errs on the generous side. Finding out a stick is full halfway through, with someone waiting to take it to the desk, is exactly what it exists to prevent.

A note on speed. Writing direct to the stick is **not** slower than converting to the internal disk and copying afterwards. The bytes have to cross to the stick once either way, at the stick's speed, and the conversion itself is far quicker than any stick: measured at 734 MiB a second of output, against perhaps 20 to 150 MiB a second for USB. So the stick is the limit in both cases, and writing direct simply saves you the copy.

4.6 After Converting

Converted rows show **Done**, and the finished file appears in the **Saves as** column.

That file can be dragged straight out of MacHuna to a Finder window, a USB drive, or anywhere else. Select several and drag them together, using ⌘-click and ⌘-click as you would anywhere on a Mac. Dragging a file out **copies** it — the file stays in the destination folder as well.

Double-click it to watch it in the player, without leaving the window.

Convert only ever runs rows that have not been done, so nothing is converted twice. Add more files to a finished batch and press Convert again: only the new ones run.

If a converted file is deleted or moved in the Finder, MacHuna notices and the row stops offering it.

4.7 Stopping

Convert ▸ Stop (⌘.) stops after the file in progress. Anything already written stays written. The conversion log is not produced for a batch that was stopped part way through.

5. Kahuna SWS Output

When outputting to Kahuna SWS, the following options are available:

Option	Description
Video standard	Target video standard. 1080p/50 is most common for UK/European broadcast.
Ignore key	No key plane is written. Output is fill-only, matching K-Watch no-alpha behaviour. Use for fill-only content or when the Kahuna output does not use a downstream keyer.
Auto play	Sets the Auto Play flag in the SWS header. The Kahuna begins playback when the clip is loaded.
Loop play	Sets the Loop Play flag in the SWS header. The Kahuna loops the clip continuously.
TGA source interlaced	See Section 8.2.

Option	Description
Include audio	Embeds audio from the source into the .SWS file. Shown only when audio is detected in the source. See Section 9.

SWS → SWS standards conversion: when the source is itself an SWS file, the output header's clip name follows the source SWS's name, and the output key state follows the source (a keyless source produces keyless output). **Audio is not carried through an SWS → SWS conversion** — if the source SWS contains embedded audio, it is dropped and a warning is written to the log.

5.1 Progressive to Interlaced (P→I)

When converting a progressive source to an interlaced standard, MacHuna field-weaves pairs of progressive frames into genuine interlaced frames. The frame count halves — a 50fps progressive source becomes 25fps interlaced output. This is the correct behaviour for the Kahuna.

Example: 100 frames of 1080p/50 source → 50 frames of 1080i/50 output.

The source must run at the interlaced field rate (double the frame rate). Weaving pairs of frames only keeps the right duration when the source is a genuine field-rate stream:

Progressive source	Interlaced target	Result
50p	1080i/50	✓ correct — weaves to 25 interlaced frames
59.94p	1080i/59.94	✓ correct
60p	1080i/60	✓ correct
25p	1080i/50	✗ blocked — would play at 2× speed
29.97p	1080i/59.94	✗ blocked
30p	1080i/60	✗ blocked

If you give MacHuna a **same-rate** progressive source (e.g. a 25p file destined for 1080i/50) or any other incompatible rate, it will **stop with a clear error and write no file**, rather than silently produce a clip that plays at the wrong speed. Convert the source to the correct field rate first (e.g. 25p → 50p), or choose a progressive output standard. *(This rate check applies to video-clip, SWS→SWS and clip→TGA conversions, which carry a known frame rate. A loose TGA image sequence has no frame rate to check and is still assumed to be a double-rate field stream.)*

5.2 Interlaced to Progressive (I→P)

When converting an interlaced source to a progressive standard, MacHuna bob-deinterlaces to produce the correct number of progressive frames. Playback speed on the Kahuna is preserved.

Example: 50 frames of 1080i/50 source → 100 frames of 1080p/50 output.

Cross-rate conversions (corrected in v1.6.12). Deinterlacing doubles the frame count, which lands on the target rate exactly only when the progressive standard is double the interlaced source's frame rate — 1080i/50 → 1080p/50, or 1080i/59.94 → 1080p/59.94. Any other pairing needs a rate change on top, and before v1.6.12 MacHuna did not apply one, so the clip played fast or slow:

Source	Target	Before v1.6.12	Now
1080i/50	1080p/50	correct	correct
1080i/50	1080p/60	ran fast	correct
1080i/59.94	1080p/50	ran fast	correct
1080i/59.94	1080p/25	ran 20% slow	correct

This applies to video clips, SWS→SWS, and TGA outputs. **A loose TGA image sequence is the exception:** it carries no frame rate anywhere, so MacHuna assumes the source standard matches the output family you pick (an interlaced pile aimed at 1080p/50 is treated as 1080i/50 material). That assumption is right for normal use. If you are converting a TGA sequence across families — interlaced 50Hz frames aimed at a 60Hz progressive standard — convert via a video clip or SWS instead, so MacHuna has a real frame rate to work from.

6. K-Frame EIF — Native K-Frame Format

EIF clips have no audio (v1.7.1). K-Frame stores clip audio in a companion `.eaf` file. That format has now been worked out (8-channel 16-bit big-endian, 48 kHz), but the code to read and write it is not built yet. `has_audio` is always false, so **any audio on your source is dropped when converting to EIF**. MacHuna warns you in the log when the selection contains audio. This is a missing feature rather than a fault, and it is the one thing in this manual that is *known* not to work rather than merely untested. One question is still open: which of the eight channels a K-Frame treats as programme audio. If you know, or can find out on a desk, that is the last piece: machuna@dnsvision.tv.

IMPORTANT — Hardware Status EIF write and conversion paths have been verified correct by analysis against real Kayenne-produced reference files, but **none have been tested on a live K-Frame desk**. MacHuna will warn you before converting to or from EIF. Verify the first import carefully.

MacHuna can read and write Grass Valley K-Frame `.EIF` files — the native clip format used by the K-Frame Image Store and ClipStore. The Image Store lives in the **frame**, not the control panel, so `.EIF` applies equally whether your surface is a **Kayenne**, a **Karrera** or a **GV Korona**. The format was fully reverse-engineered from real Kayenne-produced files.

6.1 What is EIF?

.EIF is the K-Frame's native clip container. Each file holds a complete clip — fill and key combined in a single proprietary pixel format. The format stores 1920×1080 progressive video only; frame rates are either 25fps or 50fps. Files are named by slot number: **0001.eif**, **0002.eif**, and so on.

6.2 Converting TO EIF

Any of the following inputs can be converted to K-Frame EIF:

Input	Notes
MOV, MP4, MXF, MKV, AVI	Alpha channel (fill + key) preserved if present
TGA sequence	Progressive or interlaced source — see TGA source interlaced option
Kahuna SWS	Lossless direct YCbCr repack — no RGB conversion

Output options:

Option	Description
Slot (per row)	Each row in the item list carries its own slot number, written as 0001.eif , 0002.eif and so on. Tick Number sequentially to fill them in from the first row. See Section 4.4.
TGA source interlaced	Shown when source is a TGA sequence. Tick when TGA frames are from an interlaced source. MacHuna deinterlaces each frame using yadif (field separation, TFF) to produce 50fps progressive EIF output.

EIF output is always 1920×1080. Sources of other sizes are scaled. Frame rate is rounded to the nearest EIF-supported rate (25fps or 50fps), and the video is resampled to that rate so the clip keeps its original duration and plays at the correct speed — for example a 60fps source becomes a 50fps EIF clip of the same length.

6.3 Converting FROM EIF

EIF files can be converted to the following outputs:

Output	Notes
Kahuna SWS	Lossless direct YCbCr repack. Output standard auto-derived from EIF frame rate (25fps → 1080p/25, 50fps → 1080p/50).
K-Frame TGA	Full-resolution 1920×1080 32-bit RGBA TGA sequence. Frames numbered 0001.tga onwards.
Sony TGA	32-bit RGBA TGA sequence with 4-character clip name prefix.

For TGA outputs, select the **Video standard** to control whether MacHuna field-weaves frames (interlaced standard) or extracts them as-is (progressive standard). The same field order and clip name options as regular extraction apply — see Section 7.

6.4 EIF in the Video Player

The built-in Video Player opens **.EIF** files directly. Frame rate is detected automatically from the file header — no prompt required. Fill, key, and composite panels are all populated. See Section 10.

7. Extraction Outputs — K-Frame and Sony MVS

IMPORTANT — Hardware Status The extraction output paths have been confirmed correct by code analysis. However, K-Frame TGA output has **never been loaded on a live K-Frame desk**, and Sony TGA clip naming has **never been verified on a live Sony MVS**. MacHuna will warn you before converting to these targets. Use with that caveat in mind and verify the first import on your desk carefully.

*(MOV output returned in v1.10.0 as **QuickTime MOV**, and is listed separately below. It is not a desk format, so nothing about it is awaiting hardware confirmation.)*

7.1 Output Targets

Output	Format	Destination desk
K-Frame TGA	32-bit RGBA TGA sequence. Frames numbered 0001.tga onwards. One subfolder per SWS.	Grass Valley K-Frame Image Store
Sony TGA	32-bit RGBA TGA sequence. Frames numbered XXXX0000.tga (4-character clip name + frame number). One subfolder per SWS.	Sony MVS Image Store
QuickTime MOV	ProRes 4444 .mov , key carried as a real alpha channel, audio included where the source has it. Named after the source file.	None — an ordinary video file for an edit suite

7.2 Workflow

1. Add your **.SWS** files to the list
2. Choose the output from the **Format** menu
3. Set any options shown (Video standard, Field order, Include audio) and give each row a clip name
4. Press **Convert**

7.3 QuickTime MOV — getting material back out

Added in v1.10.0. Converts a Kahuna **.SWS**, a K-Frame **.EIF** or a TGA sequence into a **ProRes 4444 QuickTime**: the fill as picture, the key as a real alpha channel, and sound where the source has any. The file is named after its source, so **51.SWS** becomes **51.mov**.

This is the way to get a graphic out of a desk format and into an edit suite, grade or archive. It is **not** a desk format, so unlike K-Frame TGA or Sony TGA there is no hardware warning on it and nothing awaiting confirmation — if it opens in QuickTime or your NLE, it is right.

Where the audio comes from:

Source	Audio
.SWS	Read from the file itself, which carries audio inline
.EIF	Read from the clip's companion .eaf file, if it has one. A K-Frame keeps clip audio in a separate file of the same name
TGA sequence	None — a sequence of images has no audio to carry

Untick **Include audio** to produce a silent MOV from a source that has sound.

Naming the file. Each item in the list has a name field. Leave it blank and the MOV is named after its source, which is usually what you want: **1.SWS** becomes **1.mov**, a sequence called **MYSEQ** becomes **MYSEQ.mov**. Type a name and it uses that instead, so a TGA render can come out as **Barcelona Wipe.mov** rather than inheriting whatever the render was called.

Names may contain spaces, letters, digits and ordinary punctuation, up to 64 characters. Anything a filesystem would refuse is blocked as you type. Two items cannot be given the same name, and a name that already exists in the destination is refused rather than overwritten — the same protection the numbered outputs have. Typing **.mov** on the end is harmless; it will not be doubled.

Frame rate. An **.SWS** and an **.EIF** both declare their rate in the header, so MacHuna uses it and no control appears. A TGA sequence declares nothing, so for TGA input a **Frame rate** control appears: 23.976, 24, 25, 29.97, 30, 50, 59.94 or 60. Set it to the rate the sequence was made at.

Fixed in v1.10.2. This used to be the **Video standard** menu, which offers only the seven verified Kahuna broadcast standards. Its only 30fps entries are *interlaced*, so a 30fps sequence had nowhere correct to go — picking a progressive standard played it at the wrong speed, and picking an interlaced one wove fields into a QuickTime. A MOV is an ordinary video file and now gets ordinary video rates. **The "TGA source interlaced" tickbox would not have helped either** — ticking it left the clip running at double speed and put progressive frames through a deinterlacer.

If your TGA frames are themselves interlaced, tick **TGA source interlaced** and choose the field order. MacHuna deinterlaces them into full progressive frames — one frame per field, so a 25fps interlaced sequence becomes a 50fps progressive MOV of the same length, with smooth motion and no combing. Leave it unticked for ordinary progressive renders, which is the normal case.

A QuickTime is never *woven* into interlace — ProRes carries no field-order flag, so there would be nothing to set and nothing downstream to read it. But frames that already contain fields still have to be *deinterlaced*, or the comb stays in the picture. Those are two different things, and v1.10.2 briefly dropped both. Restored in v1.10.3 after David spotted it.

Stills are not accepted, in line with the rest of MacHuna: a single frame is not a clip. Stills convert to Kahuna SWS only.

Interlaced sources. ProRes carries no field-order flag, so an interlaced source produces a MOV with correctly decoded interlaced frames that a downstream NLE may not automatically identify as interlaced. Set the field order in your NLE if it matters.

7.4 Standard (TGA outputs)

For K-Frame TGA and Sony TGA, select the **Video standard** matching your target desk's video standard. This determines whether MacHuna applies field-weaving (for interlaced output standards) or extracts frames as-is (for progressive standards).

- **Progressive standard selected:** frames extracted directly from the SWS
- **Interlaced standard selected:** if the source SWS is already interlaced, frames are passed through. If the source is progressive, pairs of frames are field-woven into interlaced output.

MacHuna logs a note describing what it did for each file.

7.5 Field Order (TGA outputs)

A **TFF / BFF** toggle appears for interlaced standards, and always for Sony TGA. **TFF (Top Field First) is the default** — it is the SMPTE standard for HD (including 1080i) and is correct for all known 1080i HD workflows (default since v1.6.3). If you see motion artefacts or comb effects on the desk after import, switch to BFF and reconvert.

Fixed in v1.6.12. For Sony TGA output from a TGA sequence or a video clip, this toggle was displayed but had no effect — the conversion was hardcoded to TFF, so switching to BFF changed nothing in the output. It now works in both conversion directions, and the conversion log states which field order was applied, so you can confirm what was used. If you tested Sony TGA output before v1.6.12 and concluded BFF did not help, that test was not valid and is worth repeating.

7.6 Sony TGA — Clip Name

Enter a **4-character alphanumeric clip name** (e.g. **WIPE**). All TGA frames in the batch share this name — on the Sony MVS, files with the same 4-character prefix are grouped into a single clip on import.

Each clip takes its own 4-character name from the item list (see Section 4.4), so each becomes its own output folder and several Sony clips convert together. The shared Clip name field that once forced one clip at a time is gone as of v1.8.0.

7.7 Output Structure

Output	File naming
K-Frame TGA	Subfolder per SWS, named after the SWS stem. Frames 0001.tga ... inside.

Output	File naming
Sony TGA	Subfolder per SWS, named after the 4-character clip name. Frames XXXX0000.tga ... inside.

8. TGA Sequences

TGA sequences are multi-frame clips stored as individually numbered still images. MacHuna handles them through the folder browser — you do not need to select individual frames.

MacHuna accepts any numbered TGA naming convention — K-Watch names, After Effects exports, third-party renders, or anything else:

```
TNTS201_30_0001.tga    (K-Watch naming)
FEDX0000.tga          (letters + digits, no separator)
shot_0001.tga         (underscore separator)
render.0001.tga       (dot separator)
```

The only requirement is that frames are numbered sequentially.

8.1 In the Folder Browser

When you open a folder containing a TGA sequence, MacHuna collapses the entire sequence to a single entry:

```
TNTS201  (30 frames)
```

Select this entry and convert as normal. MacHuna handles the frame ordering automatically.

8.2 TGA Source Already Interlaced

If you are re-wrapping TGA frames that were previously extracted from an interlaced SWS (e.g. via MacHuna's extraction outputs), tick **TGA source interlaced**. This tells MacHuna to pass frames through directly without applying field-weaving. Without this, MacHuna would incorrectly treat the already-interlaced frames as progressive and weave them again.

When **TGA source interlaced** is ticked and you select a **progressive** target standard (e.g. 1080p/50), MacHuna deinterlaces the frames using yadif (**send_field** , TFF per SMPTE 274M) rather than duplicating them — each interlaced frame is separated into two progressive fields, doubling the frame count with correct, smooth motion. Any alpha/key channel is deinterlaced with the identical filter so fill and key stay aligned.

8.3 TGA Sequence Output

When the input is a TGA sequence or a video clip, **TGA Sequence** appears as an output option. This re-writes the frames as a new numbered TGA sequence in a subfolder, applying an interlaced↔progressive conversion according to the selected **Standard** — useful for converting a sequence between field standards without going through SWS. Progressive→interlaced uses field-weaving (**tinterlace**); interlaced→progressive uses yadif deinterlacing. Both follow the **Field order** toggle where it is shown, and default to TFF otherwise. The alpha/key channel is preserved throughout.

8.3 Alpha Channel

- If TGA files have an alpha channel, the key plane is extracted automatically
- If no alpha is present and Ignore key is off, a solid white key plane is generated
- If Ignore key is ticked, no key plane is written regardless

8.4 Audio

Audio is not supported in TGA sequence conversions.

9. Audio

MacHuna extracts audio from source files and embeds it in the **.SWS** file in K-Watch native format.

9.1 Audio Format

Encoding	16-bit signed little-endian PCM
Sample rate	48,000 Hz
Channels	16 channels interleaved
Channel mapping	Ch1 = Left, Ch3 = Right, all others silent
Samples per frame	48000 ÷ fps (e.g. 960 samples at 50fps, 1920 at 25fps)

The channel mapping (L=Ch1, R=Ch3) matches the K-Watch convention, confirmed by hex analysis of K-Watch reference files. MacHuna uses an explicit ffmpeg pan filter — a straight **-ac 16** upmix does not produce the correct layout.

9.2 Notes

- Audio is appended after the key plane in the **.SWS** file
- If the source has no audio and Include audio is ticked, the option is simply ignored — no error

- Audio detection in third-party **.SWS** files uses the audio offset and format flag fields (0x1E8 and 0x1EC). The audio frame size field at 0x1C2 is unreliable across workflows and is not used for detection

10. The Player

The player checks a file without a Kahuna or K-Frame desk in front of you. It reads the file's own bytes at full resolution: nothing is transcoded, so what you are looking at is the file itself and not a copy of it.

Open it with the **Player** button in the toolbar, or by double-clicking a converted file in the list.

10.1 Getting a Clip In

- **Drag a clip onto the player window.** Converted or not, from anywhere
- **Double-click a converted file** in the conversion window's list
- **Press Open...** in the player's transport bar

Input	Fill / Key / Composite	Audio
.SWS	Yes	Yes, where the file has it
.SWS split files (a folder of 01_OF_03._XX parts)	Yes	Split files carry no audio
.EIF (K-Frame native)	Yes	Yes, from the clip's companion .eaf
.TGA sequence (pick any frame)	Yes, where the TGAs have alpha	No
.MOV , .MP4 , .MXF	Yes, alpha preserved for ProRes 4444 and similar	Yes

A TGA sequence carries no frame rate anywhere in the files, so the player assumes 25fps and puts a frame-rate menu in the transport bar. Change it and the timecode and playback speed follow. Nothing is re-read: the rate drives the clock, not the decoding.

10.2 The Four Panels

Panel	Content
Fill	The picture, as the file has it, with the key ignored
Key	The key plane, shown as greyscale
Composite	The fill through its own key, over a chequerboard
Audio	dBFS metering, one bar per channel

The composite sits on a chequerboard, not on black. Over black, a black fill with a full key looks identical to complete transparency — which is precisely the fault the composite panel exists to reveal.

The meter runs from -60 to 0 dBFS: green to -18, amber to -6, red above that, with a peak-hold line that looks back about a second so a transient lasting one frame is still visible.

10.3 Transport

Control	What it does
Recue	Back to the first frame, stopped
Play / Pause	Separate buttons, as on a desk, so the state is never in doubt. The space bar toggles them
Previous / Next frame	One frame at a time
Loop	Plays it again at the end, with no pause at the join
Sound / Muted	Silences the speaker and leaves the meter running
The timeline	Drag it to any frame. There is no wait: an SWS is uncompressed with a fixed frame size, so frame 400 costs the same as frame 1

Reaching the end stops and returns to the top, ready to play again.

10.4 Loading

A clip is decoded before it plays. The timeline is replaced by a load bar and the status line counts frames. A wipe takes about a second; a long clip takes longer.

This is deliberate. Decoding while playing gives every frame a deadline of a few milliseconds, and any frame that misses is visible. Decoding first means playback has nothing left to do but put finished frames on screen.

You can scrub through the part that has already loaded while the rest arrives.

On smoothness. A 50fps clip on a 60Hz screen cannot be shown perfectly evenly by any player, because 60 does not divide by 50: each frame is held for one refresh or two, in a repeating pattern. MacHuna presents frames on the screen's own clock so that cadence is regular rather than random. If motion looks very slightly uneven at 50fps, that is the screen, not the file.

11. Menus and Keyboard

Everything that matters is in the menu bar with a keyboard shortcut. The menus follow the frontmost window: Convert greys out when the player is in front, and the Player menu greys out when it is not.

11.1 MacHuna

Item	
About MacHuna	
Check for Updates...	Checks now, and always tells you the result

11.6 Help

Item	
MacHuna User Manual	Opens this document from inside the app, so it works with no internet and always matches the version you are running
Report a Problem...	See Section 12.3
Suggest a Feature...	The same panel, different prompts

11.2 File

Item	Shortcut
Open...	⌘O
Show Destination in Finder	⇧ ⌘R
Clear Destination...	

11.3 Edit

Item	Shortcut
Remove from List	⌘X
Move Up	⇧ ⌘↑
Move Down	⇧ ⌘↓

Select All (⌘A) and the usual pasteboard items are there as well. **⌘A followed by ⌘X empties the list.**

11.4 Convert

Item	Shortcut
Convert	⌘R
Stop	⌘.

Item	Shortcut
Play Converted File	⌘Y
Output Format ▸	
Video standard ▸	
Field Order ▸	Only for the outputs that have one
Include Audio, Ignore Key, Auto Play, Loop Play, Number Sequentially	Ticks

The format, the standard and the options are here **as well as** in the inspector. That is not duplication for its own sake: it is what lets the app be driven from the keyboard, and it is how someone finds out what it can do. A button on screen teaches nobody its shortcut.

Kahuna SWS has no Field Order. Its weave is confirmed TFF on real hardware, so offering a toggle would only invite breaking a verified path.

11.5 Player

Item	Shortcut
Play / Pause	Space
Recue	⌘←
Next Frame	→
Previous Frame	←
Loop, Mute	Ticks

12. Updates and Getting in Touch

12.1 Update Notifications

A few seconds after MacHuna opens, it reads a small file at dnsvision.tv/machuna/version.json in the background to see whether a newer version exists. This never delays the window opening.

If a newer version is available, a bar appears across the top of the window with the version number, a line about what changed, a **Download...** button and **Later**. Download opens the download page in your browser; MacHuna never installs anything itself.

A bar and not a dialog, deliberately. An update is news, not an interruption, and it should never stand between you and the job you opened the app to do.

If you are up to date, offline, or the check fails for any reason, nothing at all appears. No error, no delay, no message. MacHuna is used in OB trucks and galleries with no route to the internet, and an app that hangs on startup or nags about a network it cannot reach would be worse than useless. The request gives up after five seconds and is forgotten.

Later hides the bar for that session. **MacHuna ▸ Check for Updates...** shows it again, because asking is a request to be told.

Check for Updates... always reports back: a newer version, "You are on the latest version", or "Could not check for updates just now". Only the automatic check on launch is silent.

What is actually sent: nothing about you. MacHuna reads one small public file and nothing else. It sends no information about you, your machine, your files or your conversions, and there is no usage tracking of any kind. The request is an ordinary web fetch, the same as opening a page in a browser.

12.2 Hardware-Status Notices

Some of MacHuna's outputs have been confirmed on the desk they are for, and some have not. The log says which, at the head of every batch.

Output	What the log says
Kahuna SWS	Nothing. It is confirmed on a live Kahuna mainframe across all seven standards, so a warning would be false caution
K-Frame EIF, K-Frame TGA, Sony MVS TGA	A note that the output has never been loaded on that desk, and an invitation to get in touch if you can test it
K-Frame EIF, when the source has audio	An additional note that the audio is being dropped — see Section 6

These are notes, not errors. Nothing is blocked and no dialog appears. The conversion runs exactly as it always did; you are simply told where it stands so you can decide whether to check the result before trusting it on air.

The full status of every output is also published at dnsvision.tv/machuna.

12.3 Getting in Touch

Help ▸ Report a Problem... opens a panel showing exactly what is about to be shared, then hands it to your own mail app as a draft.

The draft arrives already carrying the things that otherwise take three emails to establish: the MacHuna version, your macOS version, your Mac model, the output format and video standard you had selected, your destination folder, and the files that were loaded. You write what happened in your own words in the space at the top.

- **Open Email** opens the draft in your mail app. **Nothing is sent until you send it.** You can edit or delete any part of it first, and on a truck with no signal it simply waits in your outbox

- **Copy Details** copies the same text to the clipboard, for anyone whose Mac has no mail app set up
- **Show Log in Finder** appears when a conversion log exists in your destination folder. That log usually answers the question on its own, so attaching it helps

Suggest a Feature... is the same panel with different prompts, for a format, a standard or a naming convention your facility needs.

Both go to machuna@dnsvision.tv.

13. Large File Support (>4GB)

Files larger than 4GB are automatically split into 2GB chunks. There is no setting: it always happens, because a Kahuna cannot read a file over 4GB from a FAT32 drive and there is no reason to write one. The split format exactly matches K-Watch output and has been confirmed working on a live Kahuna mainframe.

13.1 Split File Structure

```
201.SWS/
  01_OF_03._XX  (512-byte header + video data, exactly 2GB)
  02_OF_03._XX  (video data, exactly 2GB)
  03_OF_03._XX  (video data, remainder)
```

- Chunk 1 contains the SWS header followed by video data
- Subsequent chunks contain raw video data only
- The header in chunk 1 carries the total frame count across all chunks
- Audio is not supported in split files

IMPORTANT When transferring to a Kahuna via USB, copy the entire **.SWS** folder (e.g. **201.SWS/**), not the individual chunk files inside it.

14. SWS Format Reference

This section is for support engineers and developers. It documents the SWS binary format as reverse-engineered from K-Watch reference files and verified against a live Grass Valley Kahuna mainframe.

14.1 File Layout

Range	Content
0x000 – 0x1FF	512-byte header (big-endian)
0x200 – N	Fill plane: v210 big-endian, plane_size × frame_count bytes

Range	Content
N — M	Key plane: v210 big-endian, same size as fill plane (absent if <code>play_count == 0</code>)
M — EOF	Audio data: 16-bit LE PCM, 16ch, 48kHz (absent if audio offset == 0)

14.2 Key Header Fields

Offset	Type	Description
0x188	uint32 BE	Video standard code (OR'd with playback flags — see 14.4)
0x18C	uint32 BE	Format variant code (unambiguous fps lookup — all nine values are unique)
0x190	uint32 BE	Width in pixels
0x194	uint32 BE	Height in pixels (fill plane)
0x1A0	uint32 BE	Plane size (bytes per frame, fill or key)
0x1A4	uint32 BE	Frame count
0x1A8	uint32 BE	Play count (= frame count; 0 if no key plane)
0x1B4	uint32 BE	$(\text{plane_size} \times \text{frame_count} + 512) / 32$; 0 if no key
0x1C2	uint16 BE	Audio frame size — unreliable, do not use for detection
0x1CC	uint32 BE	Total file size in bytes
0x1E8	uint32 BE	Audio data offset ÷ 32 (0 if no audio)
0x1EC	uint32 BE	Audio format flag: 0x03000000 if audio present

14.3 Audio Detection

Reliable method: `aud_offset (0x1E8) > 0 AND aud_fmt (0x1EC) == 0x03000000`.

Do not rely on the audio frame size field at 0x1C2. K-Watch writes 0x1680, but third-party tools may write different values. MacHuna uses the offset and format flag fields for all audio detection.

14.4 Playback Flags

Bits 2 (0x04) and 3 (0x08) of the low byte at 0x188 are OR'd into the video standard code:

Flag	Bit	Value
Auto Play	2	0x04
Loop Play	3	0x08

14.5 Interlaced Flag

Bit 15 (`0x8000`) of `0x188` is set for all interlaced standards. The standard code for all interlaced formats is `0xC923` .

15. Troubleshooting

Kahuna showing black key / no key

- If **Ignore key** is ticked, no key plane is written — correct behaviour
- If the source has no alpha channel, no key plane is written and the desk shows the fill alone. This is deliberate as of v1.11.0 and confirmed on a live Kahuna: the flat white key that earlier versions wrote loaded as a black key and never keyed
- If you are expecting a real key and getting none, check your source file has a valid alpha channel

Audio not playing on Kahuna

- Confirm **Include audio** is ticked. It appears only when MacHuna has found audio in the source
- Confirm the source file has an audio track
- Audio is resampled to 48kHz by ffmpeg if the source is at a different rate — this is automatic

File >4GB not loading on Kahuna

- Copy the entire `.SWS` folder to the USB drive, not the individual chunk files inside it
- The USB drive must be FAT32-formatted

Convert is greyed out

Something in the list is stopping the batch. The convert bar says what; if the message has a chevron beside it, click it for the rest — it names the rows involved.

The usual causes are a blank or duplicate number, or a name that already exists in the destination folder. Nothing is ever overwritten.

Conversion fails with an ffmpeg error

- The convert bar carries the message; the conversion log in the destination folder carries the whole thing
- Confirm the source file is not corrupted — try opening it in another application
- Confirm the destination folder still exists and is writable

A converted file will not drag out

The file has to exist. If it has been moved or deleted in the Finder since it was written, MacHuna notices and the row stops offering it.

The player will not open a file

The player reads **.SWS** (including split folders), **.EIF**, TGA sequences and video files. For a TGA sequence, drop or open **any frame** and the whole sequence loads.

Extraction output not loading on K-Frame or Sony MVS

- Note that K-Frame TGA output has not been confirmed on a live K-Frame desk
- Sony TGA clip naming has not been confirmed on a live Sony MVS
- Check that the correct **Video standard** is selected for the target desk's video format
- For Sony TGA: confirm the four-character clip name matches your expected import workflow
- For interlaced output: if motion artefacts appear, try switching **Field order** (TFF ☐ BFF; TFF is the default)

EIF file not loading on K-Frame desk

- EIF write output has not yet been tested on a live K-Frame desk — proceed with caution
- Confirm the file is named with the correct four-digit slot number (e.g. **0001.eif**)
- Confirm the destination folder or USB drive is formatted correctly for K-Frame
- Check the conversion log for any warnings

TGA sequence not appearing as a single row

- MacHuna detects TGA sequences by parsing numbered filenames — frames must be numbered sequentially. Any naming convention works, as long as the base name is consistent
- When opening from the panel rather than dragging, pick the **first** frame

File plays at double speed on Kahuna

This can happen if:

- A progressive source was converted to a Kahuna SWS without using the progressive standard, and there is a mismatch between the frame count and the interlaced header
- A TGA sequence from an interlaced source was converted without ticking **TGA source interlaced** — MacHuna field-weaved the already-interlaced frames, halving the frame count again

Check the conversion log for any warnings about P→I or interlaced detection.

A clip plays at the wrong speed in the player

A TGA sequence carries no frame rate, so the player assumes 25fps until told otherwise. Use the frame-rate menu in the transport bar.

For every other format the rate comes from the file itself and is shown in the **Source** column, so a clip playing at the wrong speed there is worth reporting.

16. Known Limitations

- **Apple Silicon only** — Intel Mac builds are not supported
 - **macOS 26 or later is required.** The floor comes from the libraries MacHuna is built against, not from the code
 - **Splitting files over 4GB is automatic and has no setting.** It always happens; there is no way to write an unsplit file over 4GB
 - **Drag to reorder is not supported.** Reordering is by keyboard, `⌘↑` and `⌘↓`. Drag reorder was built and removed: its feedback could not be made reliable in the time available, and a reorder you cannot trust is worse than one you type
 - **TGA sequence audio** is not supported
 - **HLG Rec.2020** colour space is not implemented (requires a reference HLG `.SWS` file to reverse-engineer the header values)
 - **EIF write and conversion** — coded and working by analysis, but not yet confirmed on a live K-Frame desk. Verify your first import carefully
 - **EIF audio is read, not written** — MacHuna reads a clip's companion `.eaf` when converting to QuickTime MOV and when playing an EIF, but cannot yet write one, so EIF *output* is still silent
 - **Extraction outputs unconfirmed on hardware** — K-Frame TGA and Sony MVS TGA naming have not been verified on live desks
-

17. Authors

David Steer / DNS Vision Limited & Claude (Anthropic)

MacHuna was built collaboratively using AI-assisted development. The SWS format was reverse-engineered from K-Watch reference files and verified against a live Grass Valley Kahuna mainframe. The K-Frame EIF format was reverse-engineered from real Kayenne-produced clips.