MacHuna 2.12.0 for Windows: 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 for Windows is a Windows 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.

It is the Windows edition of MacHuna 2.6.0 for Mac. It does the same jobs, with the same window, the same words and the same conversion engine, so the files it writes match the Mac's, except in the one case in Section 16. Where Windows does something differently, this manual says what the PC does.

1.1 What MacHuna Does

Direction	From	To
Kahuna SWS	MOV, MP4, MXF, MKV, AVI, TGA and PNG sequences, PNG, BMP, JPG and TGA stills	`.SWS` (Grass Valley Kahuna)

Direction	From	To
K-Frame EIF	MOV, MP4, MXF, TGA and PNG sequences, `.SWS`	`.EIF` (Grass Valley K-Frame), with its `.eaf` audio
K-Frame EIF → SWS	`.EIF`	`.SWS` (Grass Valley Kahuna), with the `.eaf` audio
K-Frame EIF → TGA	`.EIF`	32-bit RGBA TGA sequence
K-Frame TGA	`.SWS`, `.EIF`, video clips	32-bit RGBA TGA sequence, 1920×1080
Sony TGA	`.SWS`, `.EIF`, video clips, TGA and PNG sequences	32-bit RGBA TGA sequence, 1920×1080 (Sony MVS naming)
TGA Sequence	TGA and PNG sequences, video clips	32-bit RGBA TGA sequence (i↔p standards conversion)
QuickTime MOV	`.SWS`, `.EIF`, TGA and PNG 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 This Edition

This Windows edition, 2.12.0, has everything MacHuna 2.12.0 for Mac has. From this version it carries the same number as the Mac release it matches: 2.10 has the same features as MacHuna 2.10 for Mac, and the last number is each edition's own fixes. It has:

- **Drag files straight onto MacHuna** from anywhere in File Explorer, and **drag finished files straight out again**, without opening the destination folder
- **Double-click a finished file to watch it** in the player, or **drag any clip onto the player**, converted or not

- **Sound for any clip**: give any clip, TGA sequence, SWS or EIF a separate sound file, as Kahuna Manager's Add Audio does (Section 9.3)
- **Clear frame**, **Write beside the source files**, and **Eject** for memory sticks that tidies away the hidden files a Mac leaves on them (Section 4.7)
- **A Quick Start Guide** in the Help menu, offered once with a strip across the top of the window
- **Light or Dark, just for MacHuna** (Section 2.5)
- **Convert runs only what is left to do**: finished rows are left alone, and a file the chosen output cannot take is marked and skipped while the rest convert (Section 4.6)
- **Ignore key works for QuickTime MOV**: ticked, the MOV has no alpha channel at all, the fill alone. The switch appears only for Kahuna SWS and QuickTime MOV, the outputs it affects (Sections 5 and 7.3)
- **A typed MOV name is kept exactly as typed**, capitals and all
- **A full menu bar**, with a keyboard shortcut for everything that matters
- **Open-source licences**: **Help ▸ Open-Source Licences** lists every part inside MacHuna, and the source code of FFmpeg is on the download page (Section 17)

New in this version, as in MacHuna 2.12 for Mac:

- **Presets.** Save the options on the right under a name and put them back in one go, from the **Preset** menu at the top of the options, **Convert ▸ Preset**, or Ctrl+Shift+1 to Ctrl+Shift+9 (Section 4.12).
- **Key slip.** Move a key up to 10 frames earlier or later against its fill, in the Key window or for several rows at once. The fill never moves (Section 4.9).
- **HDR back to SDR.** Turn HDR into SDR with the matching LUT, or with **MacHuna's own** calculation when you have none (Section 4.11). A clip to QuickTime MOV now takes HDR too.
- **The player's HDR view** shows HLG or PQ material as it would look in SDR (Section 10.5).
- **The same format, when something changes.** A K-Frame EIF to EIF, or a clip to QuickTime MOV, to change its sound, key, HDR, clear frame or Description. If nothing changes, the row says **nothing to change** (Section 4.13).
- **1080i field order confirmed** on a 1080i/50 Kahuna: top field first, as MacHuna has always written it.

New in 2.11.2, as in MacHuna 2.11 for Mac:

- **HDR.** Choose one of your HDR LUTs and every file in the batch goes through it on its way to any format: clips, TGA and PNG sequences, stills, SWS and EIF, to Kahuna SWS, K-Frame EIF, K-Frame TGA, Sony TGA, TGA Sequence and QuickTime MOV. MacHuna ships no LUT: add your broadcaster's .cube file in **Settings ▸ HDR**. The key is never changed (Section 4.11).
- **TGA and PNG sequences to Kahuna SWS have the right colours.** They were converted with SD colour maths, so saturated colours were slightly off on the desk: greens darker, reds and yellows a touch orange. Whites, greys and black were always right. Convert again any sequence you made into SWS with an earlier version.
- Changing **HDR** lets a finished row convert again, as changing the format does.
- If a file MacHuna made last time is open in another program when you convert again, the row now says so in plain words, and the old file is kept as it was.

New in 2.10.1, as in MacHuna 2.10.1 for Mac:

- **HDR clips keep their colour.** A clip labelled HDR (HLG or PQ) made into Kahuna SWS or K-Frame EIF had its strongest colours shifted, because MacHuna converted it to ordinary HD colour. Its colour values now go through exactly as they are, and the log says so. Whites and greys were never affected. If you have made HDR clips into SWS or EIF before, convert them again.
- **Sony TGA is confirmed** on a Sony XVS at 1080p/50. Interlaced has not yet been tried on a Sony desk.
- The player says "1 frame" for a still, not "1 frames".

New in 2.10.0, as in MacHuna 2.10.0 for Mac:

- **K-Frame TGA asks what format the show is.** A K-Frame shows one TGA a frame, so a 25p clip made into K-Frame TGA played at double speed on a 50p desk (found on a K-Frame). Now K-Frame TGA asks **"What format is the show?"**: 50p, 50i or 25p. Files already in that format go through untouched, and anything delivered in another, such as 25p supplied for a 50i show, is converted to play correctly on the desk. It covers clips, SWS and EIF (Section 7.4)
- **Sony TGA from an SWS or EIF keeps its length**, as Sony TGA from a clip has since 0.2.4: a 25p SWS made for 1080p/50 played at double speed
- **Stills to K-Frame EIF.** A still becomes a one-frame EIF, which a K-Frame lists as a still, just as when it imports a PNG itself, with its Description and its key. Tried on a K-Frame: the picture is the same as the desk's own import of the same PNG (Section 6.2)
- **An SWS or EIF can be the key.** Choose one in the Key window and MacHuna uses its key. Until now both were refused (Section 4.9)
- K-Frame TGA made from a 25p clip, or Sony TGA from a 25p SWS or EIF, with an earlier version plays at double speed: convert it again
- **Fixed:** on some PCs a sequence's row showed an empty square where its symbol should be. The symbol is now drawn by MacHuna itself, so it no longer depends on which Windows icon font the PC has

New in 0.2.4, as in MacHuna 2.9.0, 2.9.1 and 2.9.2 for Mac:

- **The Key window: a separate key for any file.** A fill and its key often arrive apart: a sequence and its key folder, a clip and a matte. Right-click a row and choose **Key...** (or **Convert ▸ Key...**), and give it a key: a folder of frames, a clip or a single picture, white solid and black clear. It works for every source, SWS and EIF included, and every output. The window shows the fill, the key and the two together before you commit, and checks that the key is the same size and length as the fill, or a single frame used for every frame (Section 4.9)
- **Remove a file's own key, or put it back. Remove Key** in the Key window converts a row as if its file had no key; **Use as Supplied** puts the file's own key back. The source file is never changed
- **Save Key As.** Right-click a row to save the key it will use on its own, as black and white pictures: to rework it elsewhere and bring it back, or to give a file that lost its key its key again. Or drag the Key panel out of the Key window
- **A keyed file chosen as a key gives its own key.** Choose a MOV, TGA or PNG that has a key of its own as the key, and MacHuna uses that key, not its picture
- **One key for several rows.** Select the rows, right-click and choose **Key from... for N Rows**. Any row the key does not fit is listed straight away, and selected, ready for a key of its own

- **A key line under the Source.** A plain grey **key** says the row will come out with a key, the file's own or one you added. Hover over it to see which
- **A Desk name and a Description for Kahuna SWS, as the desk shows them.** A Kahuna shows two fields for a clip: its **Name**, up to 11 characters, which the clip list shows, and its **Description**, of which it shows 32. Until now the desk's Description was always the source's file name. Now an SWS row has two boxes, one over the other: **Desk name** and **Description**. Both start filled from the row's own name, the Desk name cut to 11 at a whole word, and you can type over either, capitals, small letters and spaces as you type them; changing one leaves the other as it is (Section 5)
- **The Desk name is the row's own name, whatever the source.** A sequence used to write its first frame's name (`STING0001`, listed as `STING000`) and a still a name made from its number; both now write the name the list shows
- **No file endings on the desk.** The Description no longer ends in `.mov`, `.png` or the like
- **An SWS converted again keeps its own Desk name and Description**, unless you type others. It used to be named by its number
- **The first column is called Source name**, as it is the source file's name
- **Pictures that are not 16:9 are placed, not stretched.** Kahuna SWS, K-Frame EIF, K-Frame TGA and Sony TGA stretched every picture to fill the screen. Now a still or sequence of another shape is centred at its own size, and a clip is fitted with bars. 16:9 pictures, anamorphic ones included, fill the screen as before (Section 4.10)
- **Sony TGA and TGA Sequence keep a clip's length at any video standard.** Every frame of a clip became one TGA, so a 25p clip made for 1080p/50 played at double speed. Frames are now repeated or dropped evenly, as Kahuna SWS has always done, and what SWS refuses is refused. Convert again any TGAs made from a clip at a different rate to the standard (Section 7.4)
- **The player's title says what the desk will list**, as `3.SWS · Lower Third` or `0001.eif · LOWER THIRD`, and the player now opens a single still
- **Fixed:** a row that failed during a conversion showed FFmpeg's whole command. It now says one plain line
- **Fixed:** a file dragged over MacHuna's window and away again, quickly, or ended with Esc, could leave its picture stuck on the screen

New in 0.2.2, as in MacHuna 2.8.0 for Mac:

- **TGA rate, in place of TGA source interlaced.** A folder of frames has no frame rate of its own, so MacHuna asks how many of them make one second: **50 | 25i | 25p** (the numbers follow the video standard, so 59.94 | 29.97i | 29.97p for a 59.94 desk). **50** is the old switch off and **25i** the old switch on; **25p** is new, for renders made at 25 frames a second: each frame is kept whole (PsF) for an interlaced desk and shown twice for a 50p one. Your old setting carries over. For QuickTime MOV, which has its own Frame rate, the choice is **TGA frames: Interlaced | Progressive** (Section 8.2)
- **K-Frame EIF weaves a 50p render for an interlaced desk**, as every other format does. It used to take one frame per TGA, so a 50p render played at half speed on a 50i K-Frame (Section 8.2)
- **A 25p clip goes to 1080i/50 frame for frame (PsF)** instead of being refused, and likewise 29.97p and 30p to 1080i/59.94 and 1080i/60: each frame becomes one interlaced frame with both fields from the same picture, so speed and length are the clip's own. Other rates are still refused (Section 5.1)

- **Numbered PNGs are a sequence**, exactly as numbered TGAs are, for every output. PNG is read only; MacHuna does not write PNG (Section 8)
- **Greyscale and palette PNG stills convert.** They used to fail with an FFmpeg error, as did 16-bit greyscale PNGs
- **A folder of frames named only with numbers is one sequence**, named after its folder. `000.tga` to `229.tga` used to come in as three sequences, split at every hundred, and a MacHuna TGA output of 1000 frames or more split at frame 1000 when added back (Section 8)
- **A Kahuna lists SWS clip names in full.** MacHuna wrote the name one byte early, so the desk listed every clip without its first letter. Convert again any SWS whose name matters (Section 5)
- **Change a setting and convert again.** A Done row now converts again when anything its output depends on changes, such as Ignore key, Clear frame or Auto play, and it **replaces its own earlier file**: only a file it made in this session and nothing has changed since. Anything else of that name is still **already there** (Section 4.6)
- **MacHuna itself refuses to write an SWS or EIF over anything already there**, as it already did for every other output. It used to rely on the check made as you typed
- **Remove Done and Remove All.** The ⋯ button at the end of **+** and **−** under the list, the Edit menu and a row's right-click menu take every finished row, or every row, off the list. Files stay on disk (Section 4.6)
- **Number sequentially leaves finished rows alone.** A file added above a Done row used to renumber it, so Convert wrote it again under the new number. A Done row now keeps its number, and the numbering flows around it (Section 4.3)

New in 0.2.1, as in MacHuna 2.7.0 for Mac:

- **Your own description for an EIF.** The description a K-Frame shows for a clip used to be the source's file name, every time. Converting to K-Frame EIF, the list now has a **Description** column: type a shorter or clearer one, up to 31 characters, or leave it empty to keep the file name, which the empty field shows in grey (Section 6.2)
- **Files go in in name order.** Several files added at once now join the list sorted by name, as File Explorer sorts them, numbers by their value, whatever order they were picked or dragged in. Untick **Edit ▸ Sort Added Files by Name** to have them go in as they arrive instead (Section 4.1)
- **Its own icon.** MacHuna now has its own icon, the same as the Mac's, where it used to show Windows' plain program icon. The installer carries it too

New in 0.2.0, as in MacHuna 2.6.0 for Mac:

- **An installer.** MacHuna now installs for your Windows account alone, adds itself to the Start menu, and upgrades in place (Section 2)
- **EIFs of any length.** The 10-second limit came from FAT32 sticks, which hold no file over 4 GB. Writing to an exFAT stick, or a folder on your PC, an EIF can now be as long as you need: a K-Frame reads exFAT and plays a long clip from it to the end. On a FAT32 stick the limit stays, and MacHuna says so before converting (Section 6.2)
- **Nothing is ever written over, and you are told before Convert.** A row left without a name takes its source's name, and that name was not checked: converting it again, or converting something else of the same name, replaced the earlier file, and a second TGA sequence of the same name mixed its

frames into the first one's folder. Now a row whose output is already in the destination says **already there** in orange before you press Convert, and is skipped while the rest of the batch converts; of two rows that would write the same file, the second says **duplicate** and is skipped (Section 4.3). Underneath, the conversion itself refuses to write over anything

- **Right-click a row to remove it** (**Remove from List**), as well as Edit ▸ Remove from List and Delete
- **After Write beside, the files are one click away.** The "Done" line names the folder in blue: click it, or **Show in File Explorer** beside it, and File Explorer opens on the new files (Section 4.5)
- **A row that fails says why.** The Status column gives the short form (**already there**), and the line at the foot of the window gives the reason, in full when you click it
- **Play With Chosen Sound** is the new name for Play With This Sound, in a row's Sound menu
- **Stop stops.** Until now it waited for the file in progress, which on a long clip could mean most of a minute, or simply finished it. Now it stops within about half a second, leaves nothing cut short behind, and the row says **Stopped**. Quitting mid-batch asks first (Section 4.8)
- **Nothing you do while a batch converts can change it.** The batch is fixed when you press Convert. Remove from List and Write beside wait until it finishes; anything else you change applies to the next batch

1.4 How MacHuna Converts

Confirmed on a live K-Frame. K-Frame EIF (at 50p and 50i, with and without audio) and K-Frame TGA were loaded on a K-Frame on 7 October 2026 and played correctly.

Audio:

- **EIF output carries audio** in the companion `.eaf` file a K-Frame expects, at 24-bit. **Include audio** appears for K-Frame EIF too, on by default when the source has sound
- **An EIF and its `.eaf` are one output.** The finished file shows a waveform mark and the pair's combined size, and dragging it carries both files (Section 6)
- **SWS to SWS and EIF to SWS keep their audio**
- **Mono** plays on both left and right; **two mono tracks** become left and right

Frame rates:

- **A different frame rate keeps the clip's length.** 25p going to 1080p/50, or 59.94p going to a 50fps EIF, has frames repeated or dropped so it plays at the right speed and its sound still fits. Every route picks the same frames, and 29.97 and 59.94 are worked out exactly
- **59.94 and 60 count as one family.** Material at one can go to a standard at the other; it plays 0.1% fast or slow, which nobody can see, and the log says so
- **EIF to SWS honours the standard you choose.** A 25fps EIF can become 1080i/50, which is what it usually is

Pictures:

- **SWS to SWS stays 10-bit.** A straight copy or a frame-rate change copies the picture exactly
- **Every SWS is 1920×1080**, and so are K-Frame TGA and Sony TGA made from a video clip
- **Colour:** an HD file with no colour label is read as HD colour (BT.709), not SD. A clip labelled HDR keeps its colour values exactly in Kahuna SWS and K-Frame EIF

K-Frame TGA asks about the desk only when it matters. A progressive clip at 50fps or more is the only case where the desk's format changes the result, so only then does MacHuna ask "Is the desk 50p or 50i?" Everything else goes through frame for frame (Section 7.4).

Safer and clearer:

- **A stopped or failed EIF conversion leaves nothing half-written**, and any earlier file of the same name is left as it was. The row says **Stopped**
- **Sony clip names must be exactly four letters or digits**; anything else is refused before anything is written
- **Stills convert to Kahuna SWS only**, and a video clip in a QuickTime MOV batch is not made into a MOV. Since 2.5.5 such a row is marked before you convert and skipped, and the rest of the batch converts

> **NOTE** MacHuna converts and extracts files on your PC. 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.

1.5 Files Made by Earlier Versions of MacHuna for Mac

Files made on a Mac by an earlier MacHuna may need converting again. MacHuna for Windows reads them all, and plays them correctly in its own player.

- **An SWS with sound made before Mac version 2.2.1** plays only its first eighth of a second of sound on a Kahuna, at reduced quality. The picture and key are fine. Convert it again (Section 9.1)
- **A 25fps (50i) EIF made before Mac version 2.1** plays too fast on a K-Frame. Convert it again
- **A keyed clip over 4GB made before Mac version 2.3.0** loads on a Kahuna with no key. Convert it again (Section 13.1)

---

2. Installation

MacHuna for Windows comes as an installer. It installs for your Windows account only, so it never asks for an administrator's password, and adds MacHuna to the Start menu.

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

2.1 Which Download

There are two downloads, and they do the same thing:

Download	For
Windows (Intel or AMD)	Almost every PC: one with an Intel or AMD processor
Windows on ARM	A Windows on ARM PC only, such as one with a Snapdragon processor

Not sure? Open **Settings** ▸ **System** ▸ **About**. Under **System type**, "x64-based processor" means the Windows (Intel or AMD) download, and "ARM-based processor" the Windows on ARM one.

MacHuna needs Windows 11.

2.2 Installing It

1. **Double-click the installer** you downloaded, such as `MacHuna-2.12.0-windows-x64-setup.exe`. If Windows says **Windows protected your PC**, see Section 2.3
2. Click **Install**
3. When Setup has finished, leave **Launch MacHuna** ticked and click **Finish**

MacHuna is then in the **Start menu**, and in **Settings** ▸ **Apps** ▸ **Installed apps**, where it can be uninstalled like any other app. **Uninstalling removes MacHuna and nothing else**: your settings are kept, and so is every file you converted.

If MacHuna is open when you install or uninstall, Setup says so and asks you to close it: quit MacHuna, then click **OK**.

2.3 "Windows protected your PC"

This copy of MacHuna is not yet signed, so when you open the installer, Windows may show a blue box headed **Windows protected your PC**, saying that Microsoft Defender SmartScreen prevented an unrecognised app from starting. This is expected:

1. Click **More info**. The box shows the installer's name and "Unknown publisher"
2. Click **Run anyway**
3. Setup opens

Windows may instead show a smaller box, **Open File - Security Warning**, saying the publisher could not be verified, with **Publisher: Unknown Publisher**. Click **Run**.

If the blue box has no **Run anyway** button, your PC is managed by your organisation and does not allow it: ask whoever looks after your PC.

Checking which version you have. Help ▸ **About MacHuna** shows it, and **Help** ▸ **Check for Updates...** tells you whether it is the current release.

2.4 Upgrading

To upgrade, run the new version's installer. It replaces the old version in place: there is nothing to uninstall first.

If you had the zip of an earlier version (0.1), its MacHuna folder can be deleted once the installed MacHuna runs. Or keep it: the two do not interfere.

Your settings come with you. MacHuna keeps them in your Windows user profile, not in its folder, so a new version finds them where the last one left them: destination folder, output format, video standard, field order and the conversion options.

2.5 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.

File ▸ Settings... (Ctrl+,) has three tabs: **Presets** (Section 4.12), **HDR** (Section 4.11) and **Appearance**.

Appearance. MacHuna follows Windows' Light or Dark mode. To choose one for MacHuna alone, open **File ▸ Settings...** (Ctrl+,) and pick **Match System**, **Light** or **Dark**. It changes the whole app, the player included, straight away, and MacHuna remembers it. Match System is the default.

---

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
Source 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. A grey **key** underneath means the row will come out with a key (Section 4.9)
Number, **Clip name** or **Name**	What this item will be called. The heading changes with the output format: a number for Kahuna SWS and K-Frame EIF, a four-character clip name for Sony TGA, and a name for QuickTime MOV (leave it blank to use the source's name)
Desk name and **Description**	Kahuna SWS: two boxes, one over the other, for the Name and the Description a Kahuna shows, both filled from the row's own name. K-Frame EIF: one **Description** box, for what the K-Frame shows, empty for the file name. See Sections 5 and 6.2
Saves as	The filename that will be written. After converting, the finished file appears here as something you can drag
Sound	Where the item's sound comes from - its own, a sound file you chose, or none - and how long the sound and the picture each run. See Section 9.3

Column	What it shows
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, ⋯, which offers **Remove Done** and **Remove All** (Section 4.6), and the **Number sequentially** tickbox. At the right of that gutter is a count of what is loaded.

> **Why a minus and not a bin. −** 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 bin 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
Preset	The preset in force, or None, with **· edited** when a setting it sets has changed (Section 4.12)
Output	Format, Video standard, Field order
Conversion	TGA rate (TGA frames for QuickTime MOV), Include audio, Ignore key, Clear frame, HDR
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...** (File Explorer) and **Clear...**

Clear frame adds one transparent frame, black with the key fully off, at the **Start**, the **End** or **Both** of every clip in the batch. It is for wipes supplied starting on the first frame of vision or ending frozen on the last: without it the desk shows the graphic while the clip is cued, and holds it after the clip ends. With a clear frame the desk sits invisibly on it instead. A clip with no key has no key to turn off, so it gets a black frame. A frame at the start moves the sound one frame later, so it stays in sync. It works for every output (SWS, EIF, K-Frame TGA, Sony TGA, TGA Sequence and QuickTime MOV, where the frame is fully transparent), and nothing else about the clip changes. It starts at **Off** each time MacHuna opens, so it never carries over to the next day's work unnoticed.

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... empties the destination folder, 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. Usually everything goes to the Recycle Bin, where you can get it back. Where the folder has no Recycle Bin, such as a USB stick or a network or shared folder, MacHuna says so first: the files will be deleted permanently and cannot be recovered, and the button reads Delete Permanently. If a file is too big for the Recycle Bin, Windows asks before deleting it permanently.

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 File Explorer, press **+**, or choose **File ▸ Open...** (Ctrl+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** (Ctrl+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 File Explorer.

Or press + and choose **Add...** for files or **Add Folder...** for a folder, or use **File ▸ Open...** (Ctrl+O) and **File ▸ Open Folder...** (Ctrl+Shift+O). Windows' file dialog picks files or a folder, not both at once, so there are two. For a TGA sequence, pick **any frame**: MacHuna takes the whole sequence.

Several files added at once go in **in name order**, as File Explorer sorts them (Clip9 before Clip10, capitals and small letters alike), whatever order they were picked or dragged in. Rows already in the list never move. To have files go in exactly as they arrive, untick **Edit ▸ Sort Added Files by Name**: added one at a time, they then keep the order you add them in. Files added to a list that already has something in it are added, not substituted: dropped between two rows they go in there, dropped below the list or added with + they go at the end. 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 or PNG 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 and K-Frame EIF.** A still image is a single frame, not a clip. Kahuna SWS stores it as a still, and K-Frame EIF writes a one-frame EIF, which a K-Frame lists as a still. Sony TGA, K-Frame TGA, TGA Sequence and QuickTime MOV need a clip or a TGA sequence: in a batch with other files, the still's row says **not for** that format before you convert, and it is skipped while the rest convert.

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, 1 to 9999, 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

A number field takes digits only, up to four, and a Sony clip name stops at four characters, so there is nothing to type that the desk could not use.

Number sequentially fills the column from the first row down. Change the first row and everything below follows. A row that is already **Done** keeps its number, because its file has that number, and the numbering flows around it without using it again. 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. The one exception is a row's own earlier file, when it is converted again (Section 4.6)

Rows **named after their source** - a QuickTime MOV row left blank, and every TGA Sequence and K-Frame TGA row - are checked too, but they do not stop the batch. What the **Saves as** column shows is what will be written, so if that name is already in the destination the row says **already there** in orange,

Saves as says **skipped**, and the rest of the batch converts without it. Of two rows that would write the same name, the first converts and the second says **duplicate** and is skipped. Type a name for a MOV row, or move the earlier file, and it runs next time.

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. Underneath, the conversion itself refuses to write over anything, so the protection does not depend on these checks alone.

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

4.4 Reordering a Batch

Select one or more rows and use **Edit ▸ Move Up** (Alt+Up) or **Move Down** (Alt+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.

Finding the files afterwards. When the batch ends, the line at the foot of the window says *Done. 3 written to MacHuna.* with the folder's name in blue. Click it, or **Show in File Explorer** beside it, and File Explorer opens on the new files. The new files are in the `MacHuna` folder, not loose beside the sources.

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 on a Mac 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 File Explorer window, a USB drive, or anywhere else. Select several and drag them together, using Ctrl-click and Shift-click as you would anywhere in Windows. 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. When every row is done, Convert waits.

Change a setting and a Done row converts again. A finished row runs again when anything its output depends on changes: the output format, the destination, its name or number, its desk name or description, its key or key slip, the video standard, Ignore key, Include audio, Clear frame, Auto play and Loop play (Kahuna SWS), the field order, the MOV frame rate, the TGA rate, or the row's sound. The row says **Queued** and Saves as shows the name it will write. A setting that cannot affect it, such as Auto play for an EIF, leaves it Done.

It replaces its own earlier file, and nothing else. Converted again, the row writes over the file it made itself, so you can convert, notice a setting was wrong, and simply convert again. That is allowed only for a file the row made **in this session** that **nothing has changed since**: anything else of that name says **already there**, as always, and so does the row's own file if it has been edited, replaced or moved in, or once MacHuna has been closed and opened again. The old file is kept safely until the new one is written, and put back if the conversion fails or is stopped.

Remove Done and Remove All take rows off the list, never files off the disk: Remove Done every row that says Done, Remove All every row. They are under ⋯ at the end of **+** and **−**, in the Edit menu, and on a row's right-click menu, and they wait while a batch converts.

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

4.7 Ejecting a Memory Stick

Eject your memory stick from MacHuna, however the files got onto it. A Mac writes a hidden companion file beside every file it puts on a PC-format stick (FAT32 or exFAT, which is almost all of them): `0001.tga` arrives with a `._0001.tga`. Windows writes none, but a stick that has been through a

Mac carries them. On a Kahuna the extra entries are easy to spot. On a desk that lists a TGA sequence frame by frame, such as a Sony mixer, a 60-frame wipe arrives as 120 entries.

While a PC-format stick is plugged in, it appears at the foot of the **Destination** panel with its free space and how many extra Mac files it holds right now (or "clean"). Click **Eject...**, or press **Ctrl+E**, and MacHuna asks **"Tidy up USB Drive (E:) before ejecting?"**, explaining that the desk would show the hidden files as extra files you don't need. Click **Tidy Up and Eject** and it removes them and ejects the stick, as Windows' **Safely Remove Hardware** does. A stick that is already tidy simply ejects. A stick with no name of its own is shown as Windows names it, such as **USB Drive (E:)**.

If Windows cannot eject the stick, because a file on it is still open, MacHuna says so: close whatever is using it and try again.

It removes only the Mac's `._` companion files (recognised by their contents, so a real file whose name happens to start with `._` is left alone) and the Mac's `.DS_Store` files. It never touches your own files, and leaves the stick's own system folders, including a Mac's Trash and Windows' System Volume Information, as they are.

If you eject in File Explorer instead, the hidden files stay on the stick.

4.8 Stopping

Convert ▸ Stop (Ctrl+.) stops at once, within about half a second, the file in progress included. That file is not written, or is removed if it had been started, so nothing cut short is ever left under a real name; its row says **Stopped** and converts on the next run. Files already finished stay written. **Quitting** while a batch converts asks first: **Stop and Quit**, or **Keep Converting**. The conversion log is not produced for a batch that was stopped part way through.

4.9 Keys: the Key window

A grey **key** under a row's Source means it will come out with a key: the file's own (the alpha in a clip, TGA or PNG, the key in an SWS; every EIF has one), or one you have added. No key, no line. With **Ignore key** ticked the line fades, because no key is used. Hover over it to see where the key comes from.

To add, change or remove a key, right-click the row and choose **Key...**, or select it and choose **Convert ▸ Key...**.

Part of the Key window	
Fill	The row's file
Key	The key that will be used: **As supplied** (the file's own), another key you have chosen, or **No key**. Choose one with **Key from...** (for a sequence, pick any one of its frames), or drag a folder, clip or picture onto the box
The line under the boxes	Whether the key fits, or why not
Fill, **Key** and **Composite** panels	As in the player, with a transport to step through it, first and last frame included

What a key can be. A TGA or PNG sequence, a clip or a single picture, read as **white solid, black clear**. It must be the same size as the fill, and the same length, or a single frame, which is then used for every frame. Anything else is refused, saying both sizes or lengths, and the key you had stays. A file with a key of its own, such as a ProRes 4444 with alpha or a 32-bit TGA or PNG, chosen as a key gives **its key**, not its picture: the Key box says *using the clip's key*. An SWS (including a split one) or an EIF chosen as a key gives its own key: the Key box says *using the SWS's key* or *using the EIF's key*. An SWS with no key of its own is refused.

A key you add replaces the file's own. Remove Key converts the row as a file with no key: an SWS with no key plane, an EIF solid, TGAs and MOVs with no clear areas. On a Kahuna, an SWS with no key shows nothing when keyed; use it as a full-frame source. **Use as Supplied** puts the file's own key back. Nothing reaches the row until you press **OK**; **Cancel** leaves it as it was.

The source file is never changed. Your choice is a note against the row. At Convert, MacHuna writes a working copy with that key into a hidden folder in the destination, converts it and removes it. Adding or removing a key makes a Done row convert again.

Save Key As... on a row's right-click menu saves the key the row will use on its own, white solid and black clear, in the form **Key from...** reads, so it can be reworked elsewhere and brought back:

Row	Saved as
A TGA or PNG sequence	A folder `NAME_KEY` of frames numbered as the fill's
A still	One picture, `NAME_KEY.tga` or `.png`
A clip, SWS or EIF	`NAME_KEY.mov`, a ProRes 422 HQ matte at the clip's rate and length

For a sequence, the Save dialog asks for the folder's name: what you type is the folder MacHuna makes, and one already there is never written over.

You can also drag the **Key** panel out of the Key window into File Explorer. Nothing is written over without asking: if the folder already has one of that name, Windows asks whether to replace it (for a sequence's folder, whether to merge the two), as it does for any file you drag there. Save Key As... is greyed for a row with no key.

One key for several rows. Select the rows, right-click and choose **Key from... for N Rows**. The key is checked against each row: the rows it fits take it, the rest keep what they had. If it does not fit every row, MacHuna says which and why straight away, and **OK** selects those rows, ready for a key of their own. **Remove Key for N Rows** and **Use as Supplied for N Rows** work the same way.

Key slip: a key out of step with its fill. In the Key window, the strip above the Key panel moves the key whole frames against the fill, up to 10 either way: **+** makes it earlier, **−** later, and **Reset** puts it back in sync. The strip says **In sync**, **2 frames earlier** or **3 frames later**, and the preview shows the result frame by frame. The fill never moves: where the key has moved away, at the start or the end, there is no key. It works on any key of more than one frame, the file's own included; with no key, or a one-frame key, the strip is there but dimmed, saying why. Press **OK** and the row's grey line says **key +2** (2 frames earlier) or **key -3** (3 frames later). The slip goes with the row into Convert, **Save Key As...** and the player. Choosing a different key puts it back in sync; **Use as Supplied** keeps it when you come back to the file's own key after **Remove Key**.

One slip for several rows. Select the rows, right-click and choose **Slip Key for N Rows...**: the same − / + and Reset, for all of them at once. It starts at their slip if they share one, otherwise in sync, and says so. Each row is checked as the Key window would: a row with no key, or a one-frame key, is left as it was, and MacHuna names those rows. A slip is never part of a preset: it mends one batch, and must not carry into the next.

4.10 Pictures That Are Not 16:9

Kahuna SWS, K-Frame EIF, K-Frame TGA and Sony TGA are always 1920×1080. A 16:9 picture of another size is scaled to fill the frame, as always. That includes anamorphic video, such as 1440×1080 HDV or 720×576 widescreen, which is 16:9 on screen. From 0.2.3 any other shape is **placed**, not stretched:

Source	Placed
A still, or a TGA or PNG sequence	Centred at its own size, as Kahuna Manager places a graphic; shrunk to fit only if it is bigger than the frame
A video clip	Fitted: as large as it will go, its shape kept, with bars

Around a picture with a key is clear, so the desk shows nothing there. Around a picture without a key is black, and no key is added. TGA Sequence and QuickTime MOV keep the source's own size, as before.

Before 0.2.3 a 608×32 strip filled the whole screen and 4:3 video came out wide.

4.11 HDR: LUTs

HDR takes a batch from SDR to HDR, or back from HDR to SDR, with a LUT: the file a production uses to convert pictures, usually a `.cube`. MacHuna ships no LUT. Productions tune theirs, and some are set by the rights holder, so you add each one you were given, once, and MacHuna keeps it as a preset. For HDR back to SDR with no matching LUT, MacHuna can also work the picture out itself (below).

Adding a LUT. Choose **File ▸ Settings... ▸ HDR ▸ Add LUT...**, or drag a `.cube` file onto the list there. MacHuna checks the file (a damaged or incomplete LUT is refused, saying why) and reads what it says about itself:

Name	Suggested from the file. Call it what you like, such as "BBC LUT"
What it does	Such as SDR to HLG, when the file says
Levels	**Full range** or **Video range**: how the LUT reads the picture. The file usually says; if it does not, ask whoever gave you the LUT. Get this wrong and colours shift
Written in the file	The LUT's own copyright and licence lines, so you can see its terms

MacHuna keeps its own copy of the file, so the preset still works if the original moves. Your LUTs never leave your PC. **Rename** and **Remove** are beside **Add LUT...**, and each preset's levels can be changed there.

Using a LUT. The **HDR** line under Clear frame says **Off**, or the LUT in use in amber with what it does underneath. Choose a LUT from its menu, or from **Convert ▸ HDR**. It applies to every file in the batch, to every output format, and changes the picture (fill) only: the key is used exactly as it is, as broadcast HDR practice does (ITU-R BT.2408). The log names the LUT on every file.

HDR starts at Off each time MacHuna opens, so it can never carry into an SDR job unnoticed. If you work in HDR all the time, tick **Keep HDR On When MacHuna Reopens** in **Convert ▸ HDR** or in **Settings ▸ HDR**; the HDR line then stays amber and the status line says **HDR is on** from the start.

Output	With HDR
Kahuna SWS, K-Frame EIF	The LUT's result, 10-bit, ready for a desk set up for HDR. SWS and EIF carry no colour label: the desk's store must be set up for the same HDR (HLG or PQ)
QuickTime MOV	Labelled with what the LUT makes (for SDR to HLG: HLG, BT.2020), so editing software and players see it as HDR
K-Frame TGA, Sony TGA, TGA Sequence	The LUT's result in the TGA's RGB, 8-bit. How a desk turns an HDR TGA into its own colour has not yet been checked on a desk: the log says so

For the closest match, convert the original. From a clip, sequence or still the LUT sees each pixel's own colour. An SWS or EIF has already halved its colour (4:2:2), so fine colour detail can differ slightly from an HDR version made from the original; the log says so.

A file that is already HDR (a clip labelled BT.2020) given a LUT that expects SDR is refused, before anything is written: converting it twice would ruin it. The same goes the other way: a clip labelled SDR given a LUT that expects HDR is refused as **already SDR**, and a clip labelled PQ given a LUT that expects HLG (or the reverse) is refused too. A file with no label, such as any SWS or EIF, is taken as you say.

HDR back to SDR. HDR LUTs usually come in pairs, one each way: SDR to HDR, and a matching HDR to SDR. To turn HDR back into SDR, add the matching one, the partner of the LUT the picture was made with, and choose it as any other: Settings ▸ HDR shows it as **HLG → SDR** (or PQ → SDR) when the file says so. Its MOVs are labelled BT.709. The status line says **Ready. HLG to SDR with** *its name*, not "HDR is on".

MacHuna's own · HLG → SDR and **MacHuna's own · PQ → SDR** are always at the foot of the HDR menu, for when there is no matching LUT. "MacHuna's own" means no LUT is used: MacHuna works out the SDR picture itself. It undoes the HLG curve (or, for PQ, the PQ curve as seen on a 1,000 cd/m² display), puts HDR's reference white, the level graphics and faces are set to, at SDR's white, converts the colours from BT.2020 to BT.709, clips anything brighter or more saturated than SDR can hold, and puts the SDR curve back on. It is close, but it is not the conversion the picture was made for, so when it matters, ask for the matching LUT. The HDR line says so under it: *MacHuna is working out the SDR picture itself, as closely as it can. If you have a matching HDR to SDR LUT, use that instead.* The status line says **Ready. HLG to SDR, MacHuna's own calculation.**

4.12 Presets

A preset is a set of the options on the right, saved under a name, so a job you do often is one choice away.

Making one. Set the options as you want them, then choose **New Preset…** from the **Preset** menu at the top of the options (or **Convert ▸ Preset**). The window that opens is filled from what you have now, every setting ticked. Name it and untick anything you would rather it left alone: a preset changes only what it has ticked. Settings that do not apply to the format ticked, such as Field order for Kahuna SWS, are greyed with the reason. **Use Current Settings** fills it from the main window again, **Clear All** unticks everything, and **Save** waits for a new name and at least one tick.

A preset can hold	
Format, Video standard, show format (K-Frame TGA), Field order, MOV frame rate	
HDR (Off, a LUT, or MacHuna's own), Clear frame, Ignore key, Include audio	
Auto play, Loop play, TGA rate	
The destination: a folder, or Write beside the source files	
The player's HDR view (Off, HLG or PQ)	Not ticked unless you tick it

A preset never holds anything that belongs to one row: numbers, names, Descriptions, sounds or keys.

Using one. Choose it from the Preset menu, from **Convert ▸ Preset**, or with **Ctrl+Shift+1** to **Ctrl+Shift+9** for the first nine. Only what it sets changes. The menu then shows its name, and **· edited** if you change one of those settings afterwards: the same menu offers **Save Changes to "**name**"** or **Revert to "**name**"**. **None** leaves everything as it is. MacHuna starts with no preset in force each time it opens.

When part of a preset cannot be used, the rest still is, and the status line says what was left: a LUT that has since been removed (HDR goes Off), a folder that is not there (such as a stick not plugged in: the destination is left as it was), or a format the files loaded cannot be converted to. That part does not count as an edit, and **Save Changes** keeps the preset's own value for it.

The player's HDR view in a preset opens each clip in that view, unless the clip is labelled SDR.

Managing them. File ▸ Settings… ▸ Presets lists them with what each sets. **+** makes a new one, **–** deletes the one selected (Ctrl+Z, with Settings still open, brings it back), and ⋯ or a right-click offers **Edit…**, **Duplicate**, **Rename** and **Delete**. Double-click to edit; drag to change the order, which is also the order of Ctrl+Shift+1 to Ctrl+Shift+9.

4.13 The Same Format, When Something Changes

A K-Frame EIF can be converted to K-Frame EIF, and a clip to QuickTime MOV, to change something about it while keeping its format: its sound (a new one, or none), its key (another, removed, or slipped), HDR, a clear frame, Ignore key, or for an EIF its Description.

If nothing changes, there is nothing to convert: the row says **nothing to change**, it is skipped, and nothing is written. Hover over it to see what would make it convert.

Only what changes is changed. An EIF keeps its own picture and header, and only the fields that changed are rewritten; its `.eaf` sound is copied, replaced or dropped. A MOV whose sound alone changes keeps its picture exactly, untouched. A clip that has to be encoded again keeps its field order, pixel shape and timecode. Kahuna SWS to SWS always converts, as it always has.

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.
Desk name and **Description (per row)**	The Name and the Description a Kahuna shows for the clip. See below.
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. Also offered for QuickTime MOV (Section 7.3); not shown for the other outputs, where it would have no effect.
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 rate	How many TGA or PNG files make one second. See Section 8.2.
Include audio	Embeds audio from the source into the .SWS file. Shown only when audio is detected in the source. See Section 9.

The Desk name and Description. A Kahuna shows two fields for a clip: its **Name**, which the clip list shows, up to 11 characters, and its **Description**, of which it shows 32. An SWS row has a box for each, Desk name over Description, with capitals, small letters and spaces kept as you type them, as K-Watch and Kahuna Manager write them. Both start filled from the row's own name, without a file ending: a clip's or still's file name, a sequence's name as the list shows it, an SWS's own Name and Description, an EIF's description. A Desk name taken from a longer name is cut to 11 characters at a whole word (`Home team line-up` gives `Home team`). Changing the Description leaves the Desk name as it is: type in the Desk name to change it. A box you empty shows a grey hint and, at Convert, falls back to the row's own name, so the desk never shows a blank. The file name (`12.SWS`) does not change. Before 0.2.4 the desk's Description was the source's file name with its ending, an empty name for a sequence was its first frame's name, and for a still a name made from its number. Before 0.2.2 MacHuna wrote the Name one byte early, so a Kahuna listed every name without its first letter (`LOWER_THIRD` as `OWER_THIRD`); convert again any SWS whose name matters.

> **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 carried through**; untick **Include audio** to leave it out. **The picture stays 10-bit:** a straight copy, a frame-rate change and a weave copy the source's own picture exactly. An interlaced SWS going to the same interlaced standard passes straight through; to a different interlaced rate it is refused, except between 59.94 and 60, which pass through and play 0.1% fast or slow.

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	✅ kept frame for frame (PsF), 0.2.2
29.97p	1080i/59.94	✅ kept frame for frame (PsF), 0.2.2
30p	1080i/60	✅ kept frame for frame (PsF), 0.2.2

A source at the interlaced standard's frame rate (a 25p clip for 1080i/50) is not woven: weaving pairs would halve it and play it at double speed. Each frame becomes one interlaced frame with **both fields from the same picture**, which is PsF, the usual way 25p travels on a 50i system. Speed and length are the clip's own. Before 0.2.2 this was refused. It applies to video clips and to SWS→SWS (a 1080p/25 SWS to 1080i/50 passes its pictures through unchanged).

Any other rate (a 24p clip for 1080i/50, say) still **stops with a clear error and writes no file**, rather than silently producing a clip at the wrong speed: MacHuna does not convert frame rates. Convert the source to 50p or 25p first, or choose a progressive output standard. *(A TGA or PNG sequence has no frame rate of its own, so for those the **TGA rate** says what the frames are: Section 8.2.)*

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 MacHuna for Mac 1.6.12). 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 MacHuna for Mac 1.6.12 it did not apply one, so the clip played fast or slow:

Source	Target	Before MacHuna for Mac 1.6.12	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 TGA or PNG sequence is the exception:** it carries no frame rate anywhere, so MacHuna takes its rate from the **TGA rate** (Section 8.2) in the family of the output you pick (25i frames aimed at 1080p/50 are 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

> **Confirmed on a live K-Frame (7 October 2026).** MacHuna's EIF files were loaded on a K-Frame at 1080p50 and at 1080i 25Hz, with and without audio, and played correctly. **25fps (50i) EIF files made by MacHuna for Mac before 2.1 carry the wrong frame-rate code and play too fast: convert them again.**

Audio. A K-Frame keeps a clip's sound in a separate `.eaf` file of the same name (`0005.eif` and `0005.eaf`). When the source has sound and **Include audio** is ticked, MacHuna writes both: left and right on the K-Frame's channels 1 and 2, at 24-bit, padded with silence or trimmed to the clip's length. Untick it and no `.eaf` is written. If an earlier `.eaf` of the same name is in the destination and the new clip has no sound, MacHuna removes it, so the desk can never pair old sound with a new picture.

The pair travels as one. A finished EIF with audio shows a **waveform mark** beside the file, and the size is the two files together. Dragging it out carries both files. If you delete the `.eaf` in File Explorer, the row stops claiming the clip is done. A drag that starts in File Explorer is up to you: the two files simply sit side by side with the same number.

If you stop or a conversion fails, nothing is left half-written and any earlier clip of the same name is left exactly as it was. The row says **Stopped** and converts on the next run.

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 or PNG sequence	Progressive or interlaced, as the TGA rate says (Section 8.2)
Kahuna SWS	Lossless direct YCbCr repack - no RGB conversion
K-Frame EIF	Only when something changes: its sound, key, HDR, a clear frame or its Description. Otherwise **nothing to change** (Section 4.13)
Still (PNG, TGA, TIFF, BMP, JPEG)	A one-frame EIF, which a K-Frame lists as a still, just as when it imports a PNG itself: with its Description and its key

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.
Description (per row)	What the K-Frame shows for the clip in its clip list. Left empty, it is the source's file name, shown in grey in the empty field; type over it for something shorter or clearer. Up to 31 characters, in capitals as the K-Frame shows them. The file name (`0001.eif`) does not change.
TGA rate	Shown when the source is a TGA or PNG sequence. For a 50i K-Frame: **50** weaves the frames in pairs into 25 interlaced frames a second (before 0.2.2 an EIF took one frame per TGA, so a 50p render played at half speed), **25i** keeps each interlaced frame as it is, **25p** keeps each frame whole (PsF). For a 50p K-Frame: **50** one frame per TGA, **25p** each frame twice. See Section 8.2.

Long clips, and which stick. An EIF has no length limit of its own, but a FAT32 stick holds no file over 4 GB, which is 517 frames of EIF: about 10.3 seconds at 50fps, 20.7 at 25. Writing to a FAT32 stick, MacHuna says so before converting anything. For longer clips use an **exFAT** stick: a K-Frame reads

exFAT, and a 28.8-second, 11.9 GB clip was exported to one, loaded back and played to the end on a K-Frame (2026-10-20). A Kahuna reads FAT32 sticks only, so keep exFAT sticks for the K-Frame. For long clips with no key, such as looping backgrounds for monitors, QuickTime MOV is another route: a K-Frame loads a MacHuna MOV directly, and tick Ignore key if it needs none.

EIF output is always 1920×1080. A 16:9 source of another size is scaled; any other shape is placed in the frame (Section 4.10). 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, to the **Video standard** you choose: a 25fps EIF can be 1080p/25 or 1080i/50 (choose 1080i/50 for 50i material, which is what a 25fps EIF usually is); a 50fps EIF can be 1080p/50, or 1080i/50 by weaving pairs of frames; other progressive standards are converted to the new rate. The key is always kept, and the `.eaf` audio is carried into the SWS (left on channel 1, right on 3).
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 a 25fps EIF is treated as 50i, which is what it usually holds: it goes through untouched for a 50i show, and is deinterlaced for a progressive one. See Section 7.4.

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

> **K-Frame TGA is confirmed on a live K-Frame (7 October 2026)**, from video clips, `.EIF` and `.SWS` . On the desk, import it through **Image Store > Library > From Disk/Folders** and **select "Sequence"**: without that, the desk lists every frame as a separate still. Sony TGA has not been loaded on a Sony desk in testing; its warning was lifted on the developer's judgement.
>
> *(**QuickTime MOV** 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

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`.

A clip can go to QuickTime MOV too, when something changes: its sound, its key, HDR, a clear frame or Ignore key. A clip with nothing to change says **nothing to change** (Section 4.13). It keeps its field order, pixel shape and timecode.

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: 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 of its own. Add a sound file to its row to give it one (Section 9.3)

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

Ignore key makes a MOV with **no alpha channel at all**: the fill alone, for an edit or a screen that has no use for the key. It is not a solid white alpha. Leave it off and the key is carried as the alpha channel, as before.

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, exactly as typed, capitals and all, so a TGA render can come out as `Opening Wipe.mov` rather than inheriting whatever the render was called. If a file of the source's name is already in the destination, a row left blank says **already there** before you convert and is skipped, and the earlier file is never written over. Type a name, or move the earlier one.

Names may contain spaces, letters, digits and ordinary punctuation, up to 64 characters. Anything a filesystem would refuse is blocked as you type, including names Windows cannot hold (such as `CON` or `NUL`, or a name ending in a dot), since the desks are Windows machines. 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 MacHuna for Mac 1.10.2.** 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 of the time would not have helped either**: ticking it left the clip running at double speed and put progressive frames through a deinterlacer.

TGA frames: Interlaced | Progressive. A MOV has its own Frame rate, so for QuickTime MOV the TGA rate becomes a simpler choice. **Progressive**, the normal case for renders: each frame is one frame of the MOV, at its Frame rate. **Interlaced**, for frames that are themselves interlaced: choose the field order too, and MacHuna deinterlaces them into full progressive frames, one per field, so a 25fps interlaced sequence becomes a 50fps progressive MOV of the same length, with smooth motion and no combing.

> 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 MacHuna for Mac 1.10.2 briefly dropped both. Restored in 1.10.3 after David spotted it.

Stills are not accepted: a single frame is not a clip. Stills convert to Kahuna SWS and K-Frame EIF 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)

A TGA sequence has no frame rate of its own: the desk shows one TGA a frame at whatever format it is set to. So for **K-Frame TGA**, MacHuna asks **"What format is the show?"** (50p, 50i or 25p) and fits every file to it. Files already in that format go through untouched; others are converted to play correctly on the desk:

File	50p show	50i show	25p show
50p	Untouched	Pairs woven into interlaced frames	Every other frame
50i (also a 1080i SWS, or a 25fps EIF)	Deinterlaced: one frame per field	Untouched	Deinterlaced: one frame per frame
25p	Each frame twice	Untouched (each frame kept whole)	Untouched

Other rates, such as 29.97 or 59.94, go through frame for frame. A K-Frame has no 1080p 25 setting: a 25p show runs the desk at 1080sf 25Hz (PsF) or 1080i 25Hz.

K-Frame TGA has no Video standard control: the show's format takes its place.

Sony TGA keeps its Video standard control. **A video clip, SWS or EIF keeps its length** at that standard's rate, by the same rule as Kahuna SWS (clips since 0.2.4, SWS and EIF since 2.10.0):

Source	Standard	Sony TGA
At the standard's own rate	-	One TGA per frame
25p	1080p/50	Each frame twice
50i (also a 1080i SWS, or a 25fps EIF)	1080p/50	Deinterlaced: one frame per field
50p	1080p/25	Every other frame
50p	1080i/50	Pairs woven into interlaced frames
25p	1080i/50	Each frame kept whole (PsF)
Another progressive rate	A progressive standard	Frames repeated or dropped evenly
A rate Kahuna SWS refuses, such as 29.97p for 1080i/50, or 50i for 1080i/59.94		Refused, with the reason

From an SWS or EIF, only 25 and 50 are fitted; other rates, such as 59.94, go through frame for frame. Until 0.2.4 every frame of a clip became one TGA, and until 2.10.0 every frame of an SWS or EIF, so a 25p source at 1080p/50 played at double speed. Convert again any Sony TGAs made from a clip, SWS or EIF at a different rate to the standard. Sony TGA from a TGA sequence is unchanged. Every K-Frame TGA

and Sony TGA frame is **1920×1080, 32-bit RGBA**, whatever the source. A video clip of another size is scaled (an interlaced one a field at a time, so the fields never blend), or placed if it is not 16:9 (Section 4.10).

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. If you see motion artefacts or comb effects on the desk after import, switch to BFF and reconvert.

> **Fixed in MacHuna for Mac 1.6.12.** 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 MacHuna for Mac 1.6.12 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.

7.7 Output Structure

Output	File naming
K-Frame TGA	Subfolder per SWS, named after the SWS stem. Frames `0001.tga` … inside.
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, so you do not need to select individual frames. **Numbered PNGs are handled exactly the same way**, for every output: everything in this section about TGA sequences applies to PNG sequences too. A PNG on its own, or one without a number, is a still. MacHuna reads PNG but does not write it.

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

```
PROMO201_30_0001.tga  (K-Watch naming)
LOGO0000.tga          (letters + digits, no separator)
shot_0001.tga         (underscore separator)
render.0001.tga       (dot separator)
0001.tga              (numbers only)
WIPE_001.png          (PNG)
```

The only requirement is that frames are numbered sequentially. **Frames named only with numbers** (`000.tga`, `001.tga` ...) are one sequence, named after the folder they are in. A PNG sequence and a TGA sequence of the same name in one folder stay apart.

8.1 In the Folder Browser

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

```
PROMO201  (30 frames)
```

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

8.2 TGA Rate

A TGA or PNG sequence carries no frame rate, so MacHuna asks **how many of its files make one second**. The **TGA rate** appears under Conversion whenever a sequence is loaded, as **50 | 25i | 25p**. The numbers follow the video standard: **59.94 | 29.97i | 29.97p** for a 59.94 desk, **60 | 30i | 30p** for a 60 desk.

TGA rate	What the frames are	Interlaced desk (1080i/50)	Progressive desk (1080p/50)	1080p/25
50	Renders at 50 frames a second	Woven in pairs (TFF) into 25 interlaced frames a second	One frame per file	One frame per file
25i	Already interlaced: each file holds two fields	One interlaced frame per file, as it is	Deinterlaced into two frames per file (yadif, TFF)	Deinterlaced, one frame per file
25p	Renders at 25 whole frames a second	One frame per file, both fields from it (PsF)	Each file shown twice	One frame per file

Each choice keeps the render's own speed: 25 files are one second at 25i and 25p, and 50 files are one second at 50.

- **25i is for frames extracted from an interlaced desk file** (an interlaced SWS, a 25fps EIF), or rendered interlaced. They pass straight through to an interlaced desk without being woven a second

time, and are deinterlaced for a progressive one. Any key is treated with the identical filter, so fill and key stay aligned.

- **25p** used to be possible only by ticking **TGA source interlaced**, which kept the frames whole for an interlaced desk but, for a 50p desk, played them at double speed or softened them by deinterlacing. Now each frame is simply shown twice.
- **For QuickTime MOV**, which has its own Frame rate, the choice is **TGA frames: Interlaced | Progressive** (Section 7.3).

Your setting carries over from earlier versions: TGA source interlaced ticked became **25i**, unticked became **50**. The TGA rate is remembered between sessions.

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.

From a video clip, the clip keeps its length at the selected standard's rate, as for Sony TGA (Section 7.4): frames are repeated or dropped evenly, and a rate Kahuna SWS refuses is refused. Before 0.2.3 every frame of the clip became one TGA, so a 25p clip at 1080p/50 came out half its length.

8.4 Alpha Channel

- If TGA files have an alpha channel, the key plane is extracted automatically
- If no alpha is present, no key plane is written: the clip is fill-only (since MacHuna for Mac 1.11, after a flat key plane was found to load as a black key on a Kahuna)
- If Ignore key is ticked, no key plane is written regardless

8.5 Audio

A TGA sequence has no sound of its own, but you can give it one: drop a WAV or AIFF onto its row, or choose one from its Sound column (Section 9.3). It goes into Kahuna SWS, K-Frame EIF and QuickTime MOV output. TGA output cannot carry sound.

Check the two lengths in the Sound column. A TGA sequence has no frame rate, so how long its picture runs depends on the output and on the **TGA rate**: the same 250 frames are 10 seconds at 25i or 25p, and 5 seconds at 50. If the sound came with the frames, a picture half or twice the length you expect usually means the TGA rate is wrong.

---

9. Audio

MacHuna extracts audio from source files and embeds it in the `.SWS` file in K-Watch native format, or writes it to the `.eaf` file beside a K-Frame `.EIF` (Section 6). Whatever the source, the audio comes out in the form the target desk expects: a 24-bit source going to a 16-bit SWS keeps its top 16 bits, and a 16-bit source going to a 24-bit `.eaf` is carried exactly. Mono goes to both left and right; two mono tracks become left and right.

9.1 Audio Format

Encoding	16-bit signed PCM, big-endian, one 4-byte slot per channel
Sample rate	48,000 Hz
Channels	8 slots per sample (32 bytes)
Channel mapping	Ch1 = Left, Ch2 = Right, all others silent
Length	The header says exactly how many samples there are, and a Kahuna plays that many

Confirmed on a live Kahuna (9 October 2026), by ear. This is the layout K-Watch writes, settled by comparing K-Watch conversions that play correctly on a Kahuna with the same sources, sample by sample. Before MacHuna for Mac 2.2.1, MacHuna wrote the sound differently, and a Kahuna played only its first eighth of a second: **convert any SWS with sound made by an earlier version again.** MacHuna still reads its older files correctly.

The sound keeps its own length, as Kahuna Manager writes it. If the sound runs longer than the picture, all of it goes into the SWS, and on the desk the picture holds its last frame while the sound plays on; if it is shorter, the picture carries on in silence. A picture that should disappear before the sound ends does so through its key. Before MacHuna for Mac 2.3 the sound was cut or padded to the picture.

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 uses the audio offset and format flag fields (0x1E8 and 0x1EC). The four bytes at 0x1C0 are the number of samples, which a Kahuna plays exactly

9.3 Adding a Sound File

Any clip, TGA sequence, SWS or EIF can carry a separate sound file instead of its own sound, the way Kahuna Manager's Add Audio does. Graphics are often delivered as frames in one file and sound in another.

To add one:

- **Drop a sound file onto the row it belongs to.** WAV, AIFF, BWF and other sound files all work. Dropped anywhere else in the list, it is not added as a clip; MacHuna says where it goes

- Or press **+ Add sound...** in the row's Sound column, or open the **Sound menu** (the arrow at the right of the Sound column, a right-click on the row, or **Edit ▸ Sound**) and choose **Choose Sound File...**. From there you can also pick a video clip whose sound you want

With several rows selected, a right-click or Edit ▸ Sound sets them all at once.

The Sound column shows where each row's sound comes from and how long the sound and the picture each run, for example *Sound 1:54 · picture 0:10*. The Sound menu also offers **Use the Clip's Own Sound**, **No Sound** (for one row, even though it has sound of its own; the column then reads *Sound off*), **Play With Chosen Sound** (the source in the player with the chosen sound, before you convert) and **Show Sound File in File Explorer**.

Saving a row's sound. Drag the sound from the Sound column, where it says *From the clip* or shows a sound file's name, to File Explorer, and it arrives there as a WAV named after the source (`OPENER.SWS` gives `OPENER.wav`, a split `201.SWS` folder `201.wav`). Or choose **Save Sound As...** from the row's Sound menu. It is the reverse of adding a sound file: take the sound off an old wipe and put it on a new one, or open it in an audio editor.

- **All of it.** An SWS's sound keeps its own length, so a sound that runs on past the picture comes out whole. An EIF's sound is the length of its picture, as the K-Frame stores it.
- **Nothing lost.** The WAV is as good as the sound in the file: 16-bit from an SWS (all an SWS holds), 24-bit from an EIF or a clip that holds 24. Always 48 kHz stereo, left and right as channels 1 and 2.
- **A copy.** The source is never changed. If the folder already has a WAV of that name, Windows asks whether to replace it, as it does for any file you drag there; the line at the foot of the window then says **Saved the sound as OPENER.wav.** or **Nothing was saved. Save Sound As...** asks the same before replacing a file. Where a sound file has been chosen for the row, that is the sound saved, as it is the one that would go into the output.
- **Written as you drop it**, not before, so a long list stays quick.

What each output does with it:

Output	The sound
Kahuna SWS	Kept at its own length, as Kahuna Manager does (Section 9.1)
K-Frame EIF	Cut or padded to the picture's length, because a K-Frame plays sound with the picture. The Sound column says so in orange before you convert
QuickTime MOV	Kept at its own length
K-Frame TGA, Sony TGA, TGA Sequence	None: a TGA cannot carry sound. The Sound column is dimmed, and your choice is kept for when you switch back

The sound is treated like a clip's own: the first two channels become left and right, a mono sound goes to both, and anything not at 48 kHz is converted to it.

Include audio in the inspector is the master switch. Untick it and nothing gets sound, chosen or not; the Sound column dims to show it.

---

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. Windows' file dialog picks files, so open a split SWS from any of its parts, and a TGA sequence from any of its frames

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	Yes (splits made by MacHuna for Mac before 2.3.0 were written without sound)
`.EIF` (K-Frame native)	Yes	Yes, from the clip's companion `.eaf`
`.TGA` sequence (pick any frame)	Yes, where the TGAs have alpha	Only a sound file chosen for it (Play With Chosen Sound)
`.MOV`, `.MP4`, `.MXF`	Yes, alpha preserved for ProRes 4444 and similar	Yes
A still (PNG, TGA, BMP, JPG)	Yes, where it has alpha	No

The title says what the desk will list. An SWS or EIF opens as its file name and then the name the desk shows for it, read from the file itself: `3.SWS · Lower Third`, `0001.eif · LOWER THIRD`.

A TGA sequence carries no frame rate anywhere in the files, so the player is told one 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.

Where that rate comes from depends on how you opened the sequence:

Opened from	Rate used
A converted row in the conversion window	The rate of the output standard you converted to. MacHuna wrote the files, so it knows - 1080p/50 opens at 50fps, 1080i/50 at 25fps, because an interlaced standard weaves field pairs
File Explorer, or the player's Open…	**50fps**, which is overwhelmingly what a modern wipe runs at. There is nothing in the files to go on

Either way the menu overrides it.

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. When the sound runs longer than the picture, the whole of the sound is played each time
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.

When the sound runs longer than the picture, the player carries on to the end of the sound, as a Kahuna does: the picture holds its last frame (the Fill panel says so), a yellow mark on the timeline shows where the picture ends, and the status line says how far the sound runs on.

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.

10.5 HDR View

HDR view, beside **Open...** and in the **Player** menu, shows HLG or PQ material as it would look in SDR, so you can check it on an ordinary screen. Choose **HLG** or **PQ** for what the clip is; **Off** shows the file's own values. It is for viewing only: nothing is converted.

It uses your own HDR to SDR LUT for that kind of HDR if you have added one (the newest, if several), and MacHuna's own calculation if not (Section 4.11). Hover over it to see which. It changes the Fill and Composite panels as each frame is shown, playing or stopped; the Key panel is never changed.

Each clip opens with the view **Off**, unless a preset in force sets it (Section 4.12) and the clip is not labelled SDR.

---

11. Menus and Keyboard

Everything that matters is in a menu bar with a keyboard shortcut. On Windows each window has its own menu bar: the conversion window has **File**, **Edit**, **Convert** and **Help**, and the player has **Player**.

11.1 File

Item	Shortcut
Open...	Ctrl+O
Open Folder...	Ctrl+Shift+O
Show Destination in File Explorer	Ctrl+Shift+R
Clear Destination...	
Eject *stick name*... (only while a PC-format stick is plugged in)	Ctrl+E
Settings...	Ctrl+, (Presets, HDR and Appearance; Section 2.5)
Exit	Alt+F4

11.2 Edit

Item	Shortcut
Select All	Ctrl+A

Item	Shortcut
Remove from List (also on a row's right-click menu)	Delete
Remove Done (also under ⋯ below the list, and on a row's right-click menu)	
Remove All (likewise)	
Move Up	Alt+Up
Move Down	Alt+Down
Sort Added Files by Name (ticked: files join the list in name order; unticked: in the order they arrive)	
Sound ▸ Choose Sound File..., Use the Clip's Own Sound, No Sound (for the selected rows)	

A row's right-click menu has the Sound items, **Key...** and **Save Key As...** (for one row), or **Key from... for N Rows**, **Remove Key for N Rows**, **Slip Key for N Rows...** and **Use as Supplied for N Rows** (for several), then **Remove from List**, **Remove Done** and **Remove All**.

Ctrl+A followed by Delete empties the list.

11.3 Convert

Item	Shortcut
Convert	Ctrl+R
Stop	Ctrl+.
Play Converted File	Ctrl+Y
Key...	Opens the Key window for the selected row (Section 4.9)
Output Format ▸	
Video Standard ▸	
Field Order ▸	Only for the outputs that have one
Include Audio, Ignore Key, Auto Play, Loop Play, Number Sequentially	Ticks
Preset ▸	None, your presets (Ctrl+Shift+1 to Ctrl+Shift+9), **Save Changes** and **Revert** when edited, **New Preset...** and **Manage Presets...** (Section 4.12)
HDR ▸	Off, your LUTs, **MacHuna's own · HLG → SDR** and **· PQ → SDR**, **Keep HDR On When MacHuna Reopens**, and **Manage HDR LUTs...** (Section 4.11)

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.

11.4 Player

Item	Shortcut
Play / Pause	Space
Recue	Ctrl+Left
Next Frame	Right arrow
Previous Frame	Left arrow
Loop, Mute	Ticks
HDR View ▸	Off, HLG, PQ (Section 10.5)

11.5 Help

Item	
MacHuna Quick Start Guide	A short tour of the things that are easy to miss, from inside the app
MacHuna User Manual	Opens this document from inside the app, so it works with no internet and always matches the version you are running
Open-Source Licences	The open-source software inside MacHuna, its licences, and where to get FFmpeg's source (Section 17)
Report a Problem…	See Section 12.3
Suggest a Feature…	The same panel, different prompts
Check for Updates…	Checks now, and always tells you the result
Engine Check…	Asks MacHuna's conversion engine what it is and what it can do: its version, the standards and conversions it offers, and whether FFmpeg is there
About MacHuna	The version, and the open-source credit

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-pc.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 for your PC, Windows (Intel or AMD) or Windows on ARM, 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. **Help ▸ 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.

A test version, numbered 0.something, does not check by itself; **Help ▸ Check for Updates…** works as usual.

> **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

None remain. Kahuna SWS is confirmed on a live Kahuna across all seven standards; K-Frame EIF and K-Frame TGA were confirmed on a live K-Frame on 7 October 2026; Sony TGA was confirmed on a Sony XVS at 1080p/50 on 11 November 2026 (interlaced has not yet been tried on a Sony desk). If anything you load on a desk does not behave, please get in touch (Section 12.3).

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 and which download it is (Intel or AMD, or ARM), your Windows version, your PC's maker and 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 PC has no mail app set up
- **Show Log in File Explorer** 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)
```

- The parts are one ordinary SWS file cut into 2GB pieces: header, fill, key, then sound
- Chunk 1 starts with the SWS header; the sound, if any, is at the end of the last part
- The header in chunk 1 carries the total frame count across all chunks

> **Convert keyed clips over 4GB again if they were made by MacHuna for Mac before 2.3.0.** Those versions wrote every split file with a header saying it had no key, and left its sound out. Confirmed on a Kahuna on 2026-10-12: such a clip loads with **no key at all**. A split MacHuna writes now, of the same clip as K-Watch's own, played identically on the desk: key, sound and length. Clips under 4GB, and clips with no key, are not affected by the key fault.

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

13.2 Converting a split file

A split file can be **converted**, not only played: drag its `.SWS` folder into the list, or choose any part inside it with **Add...**, and it becomes one SWS row, ready to go to EIF, K-Frame TGA, Sony TGA, QuickTime MOV or another SWS. A folder dropped on the list that holds split files brings each one in.

A split file to EIF needs an exFAT stick. Every split file is longer than 517 frames, the most an EIF can be on a FAT32 stick (Section 6.2), so to make an EIF of one, write it to an exFAT stick or a folder on your PC. MacHuna says so before converting anything if the destination is FAT32. SWS, TGA and QuickTime MOV have no such limit.

Rescuing an old keyed split. A keyed split made by MacHuna for Mac before 2.3.0 loads on a Kahuna with no key, because its header wrongly says it has none. MacHuna reads the key that is really there, so converting such a file **SWS to SWS** at the same standard writes a new one with a correct header, and the key comes back on the desk. You do not need the original source.

---

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
`N – M`	Key plane: v210 big-endian, same size as fill plane (absent if `play_count == 0`)
`M – EOF`	Audio data: 16-bit LE PCM, 16ch, 48kHz (absent if audio offset == 0)

14.2 Key Header Fields

Offset	Type	Description
`0x0FC`	char[12]	Clip name: what a Kahuna lists, 11 characters at most, then a null. MacHuna wrote it at `0x0FB` before 0.2.2, so the desk dropped its first letter
`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)

Offset	Type	Description
`0x1A4`	uint32 BE	Frame count
`0x1A8`	uint32 BE	Play count (= frame count; 0 if no key plane)
`0x1B4`	uint32 BE	`(plane_size × frame_count + 512) / 32`; 0 if no key
`0x1C0`	uint32 BE	Number of audio samples (0 if no audio); a Kahuna plays exactly this many
`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`.

The four bytes at `0x1C0` are the **number of audio samples**, and a Kahuna plays that many and no more (confirmed 9 October 2026). Earlier versions of the Mac's manual called the last two of them an unreliable "audio frame size"; the value `0x1680` it quoted was one reference file's sample count, mistaken for a constant.

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

A LUT was refused

Settings ▸ HDR says why under the list: the file may have been cut short when it was copied, be a 1D LUT (MacHuna needs a 3D one), or hold something a .cube cannot. Ask whoever gave you the LUT for a fresh copy.

A file was refused as "already HDR"

The clip is labelled HDR (BT.2020) and the chosen LUT expects SDR. It is already HDR: set HDR to **Off** and convert it as it is (its colour values go through unchanged, Section 1.3, New in 2.10.1), or choose a LUT made for HDR material.

A file was refused as "already SDR"

The clip is labelled SDR (BT.709) and the chosen LUT, or MacHuna's own, turns HDR into SDR. There is nothing to undo: set HDR to **Off**. If the clip really is HDR but wrongly labelled, convert it to an SWS first (which carries no label) and use that.

A row says "nothing to change"

It is already in the format you chose (a K-Frame EIF to EIF, or a clip to QuickTime MOV) and no setting changes it, so there is nothing to write. Choose another format, or change what you meant to change: its sound, key, HDR, a clear frame or its Description (Section 4.13).

Colours from a TGA or PNG sequence look different from before

From 2.11.2 a sequence to Kahuna SWS has the right colours. Earlier versions used SD colour maths for it, so its greens were darker and its reds and yellows a touch orange. The new file is the right one; convert again any you made before.

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, 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, and that the row's key was not removed in the Key window: the grey **key** under its Source says whether it will have one (Section 4.9)

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 File Explorer 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

- 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

- 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 and PNG sequences by parsing numbered filenames, so frames must be numbered sequentially. Any naming convention works, as long as the base name is consistent; frames named only with numbers are one sequence, named after their folder

- When adding with **Add...** rather than dragging, pick any frame of the sequence

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 with the **TGA rate** at **50**: MacHuna wove the already-interlaced frames in pairs, halving the frame count again. Choose **25i** (Section 8.2)
- A 25p render was converted at **50**: choose **25p**
- A 25p or 50i clip was made into **Sony TGA** or **TGA Sequence** at 1080p/50 before 0.2.3: convert it again (Section 7.4)
- A 25p clip, SWS or EIF was made into **K-Frame TGA** for a 50p K-Frame, or a 25p SWS or EIF into **Sony TGA** at 1080p/50, before 2.10: convert it again (Section 7.4)

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 is told one: the output standard's rate when you open a converted row, and 50fps when you open one from File Explorer. Use the frame-rate menu in the transport bar to change it.

If a sequence you converted plays at the wrong speed, check the **frame count** rather than the rate. Extracting an EIF to TGA writes one file per EIF frame and does not resample, so a 25fps EIF converted to a 1080p/50 sequence has the original number of frames and runs in half the time at 50fps. That is the extraction behaviour, not a player fault.

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

- **Windows 11, 64-bit, is required.** MacHuna for Windows has been tested on Windows 11; Windows 10 has not been tried
- **The conversion engine is an Intel/AMD (x64) program in both downloads.** On a Windows on ARM PC, Windows runs it through its own emulation; the window itself is native ARM
- **This copy is not yet signed**, so Windows asks before the installer opens (Section 2.3)
- **Paths are limited to 259 characters**, as Windows limits them unless a system setting is changed, which MacHuna does not touch. MacHuna says so before converting: choose a destination with a shorter path
- **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, Alt+Up and Alt+Down. 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

- **HDR to TGA (K-Frame and Sony) has not been checked on a desk, and TGA is 8-bit.** For HDR on a desk, Kahuna SWS and K-Frame EIF are the proven route
- **Sony TGA is confirmed on a Sony XVS at 1080p/50.** Interlaced (25i) has not yet been tried on a Sony desk; its field order follows the documented convention
- **EIF to EIF, and SDR made by MacHuna's own calculation, have not yet been loaded on a desk.** EIF to EIF keeps the EIF's own header and picture; MacHuna's own SDR is an ordinary BT.709 picture
- **59.94 audio** is written at 1,602 samples a frame (801 at 59.94p), which keeps sync for the whole clip; there is no K-Watch 59.94 reference file to compare it with

---

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.

17.1 Open-Source Software

MacHuna includes open-source software, each part under its own licence: **FFmpeg** (which does the video decoding, scaling and encoding), with the zlib, bzip2 and XZ libraries built into it; **Python**, **NumPy** and **Pillow**; **Tcl/Tk**; **OpenSSL**; **.NET**; and a number of smaller libraries. **Help ▸ Open-Source Licences** lists every one with its version and licence, and the full licence texts are in the **Licences** folder beside MacHuna.exe.

FFmpeg is distributed under the GNU General Public License. Its complete source code, exactly as built into each version of MacHuna for Windows, is available from the download page as **MacHuna-*version*-windows-open-source.zip**, and on request from machuna@dnsvision.tv.

MacHuna runs FFmpeg as a separate program. MacHuna's own code is not part of FFmpeg and is not covered by its licence.