

Understanding Hasselblad Phocus on macOS - The Hidden Rules Behind 3FR, FFF, HNCS, and HNNR

- [Understanding Hasselblad Phocus on macOS - The Hidden Rules Behind 3FR, FFF, HNCS, and HNNR](#)
 - [Change Log](#)
 - [Why You Still Need Phocus Even If You Don't Want to Edit in It](#)
 - [Where Phocus Fits Into the Hasselblad Workflow](#)
 - [Getting Files Off the X2D II and Into Phocus Without the Confusion](#)
 - [Importing From a CFexpress Card and What Usually Happens](#)
 - [Importing From the X2D II Over USB-C and How It Differs From Using a CFexpress Card](#)
 - [Understanding the Mass Storage Prompt When Phocus Is Already Open](#)
 - [Importing When the Camera Is Already Connected and Why Files May Appear as FFFs](#)
 - [How to Ensure Phocus Imports Your Files as 3FR Instead of FFF](#)
 - [Why Choosing Mass Storage vs Media Transfer Protocol \(MTP\) Changes What Phocus Imports](#)
 - [Why Getting the Import Right Matters for HNCS, HNNR, and the Rest of the Workflow](#)
 - [What Actually Gets Applied at Import \(and What Doesn't\)](#)
 - [Choosing an HNCS Preset at Import and Which One Makes Sense for Most Images](#)
 - [Why Phocus Still Shows "3FR" After Choosing an HNCS Preset \(and What It Really Means\)](#)
 - [What Phocus Applies in Memory but Doesn't Save Right Away](#)
 - [What Actually Triggers Phocus to Write a Sidecar File](#)
 - [Why the Image You See in Phocus May Not Match What's Actually Saved](#)
 - [How Phocus Bakes In Your Import Preset Even If Nothing Has Been Saved Yet](#)
 - [How Phocus Separates What You See From What It Actually Saves](#)
 - [How the Adjustments Dropdown in Phocus Changes Meaning at Each Stage](#)
 - [How to Apply an HNCS Preset to Multiple Images in Phocus \(and Why It Doesn't](#)

[Work Like You Expect](#)

- [How Phocus Creates and Uses the .phos Sidecar File](#)
 - [Understanding 'Save Changes' vs Auto-Save in Phocus](#)
 - [Understanding 'Save Changes' vs Auto-Save in Phocus](#)
 - [When Phocus Creates or Updates the .phos Sidecar File](#)
 - [Why Only User Edits \(Not Preset Changes\) Trigger a .phos Sidecar Update in Phocus](#)
 - [What's Actually Inside a .phos Sidecar File \(and What Isn't\)](#)
 - [How Phocus Rewrites the .phos Sidecar Over Time \(and Why It Never Stores a History\)](#)
 - [Why RAW + Sidecar Files Can Look Different on Another Machine in Phocus](#)
 - [How HNNR Creates a New RAW and Sidecar Pair in Phocus](#)
 - [What All This Means for Understanding Sidecars in Phocus](#)
 - [Understanding Hasselblad Natural Noise Reduction \(HNNR\)](#)
 - [Where HNNR Lives in Phocus and How You Actually Run It](#)
 - [What Phocus Creates When You Run HNNR \(New RAW and New Sidecar\)](#)
 - [What's Actually Inside the HNNR RAW File \(and Why It Looks Like a Normal 3FR\)](#)
 - [How HNNR Changes Your Workflow, Storage, and File Management](#)
 - [Some things worth noting:](#)
 - [Two independent RAWs](#)
 - [File count doubles](#)
 - [Choose when to apply it](#)
 - [How Fast HNNR Runs on Different Macs \(and Why Batch Processing Is Slow\)](#)
 - [Key Takeaways for Using HNNR in Your Workflow](#)
-

Change Log

v1.4 – 2025-11-27

- Corrected explanation of "Save Changes" button – it creates reusable adjustment snapshots, not sidecar persistence
- Clarified that "Always Save" preference (enabled by default) handles automatic sidecar writes
- Added terminology table explaining Save Changes vs Auto-Save behaviour
- Removed incorrect guidance suggesting manual saves are required for edits to persist

Why You Still Need Phocus Even If You Don't Want to Edit in It

When I bought my Hasselblad X2D II in October 2025 - my first Hassy, I knew I was signing up for a different kind of workflow. I've been shooting Leica for years, and I love that system deeply, but the moment I started pulling images off the Hassy, there was no going back - the files have a depth and subtlety that just grabbed me. The quirks? Sure. The learning curve? Definitely. But the results are worth every bit of it.

While figuring out how to fit the X2D II into my existing workflow, I kept bumping into behaviours in Phocus - Hasselblad's RAW processing software - that weren't obvious from the user manual. None of it is complicated once you understand it, but very little of it is spelled out. So I started taking notes. And testing things. And digging through obscure forum posts. And testing some more.

This post is the result of that curiosity. It's not meant to be definitive or authoritative - I've only been a Hasselblad owner for a short time. What you'll find here is a collection of observations, experiments, and practical explanations that would have saved me a lot of head-scratching in the first couple of weeks. Some of what I describe is based on documented behaviour, some of it's based on what the software actually does, and some of it is educated guesswork informed by testing.

If you're new to the X2D II or to Phocus, or if you're coming from Leica (like me) and want to understand how things differ under the hood, you might find this helpful. And if you spot something I've misunderstood, please reach out - I'm always up for learning more about this system.

Where Phocus Fits Into the Hasselblad Workflow

Phocus generally makes importing straightforward, but the X2D II seems to have a couple of modes that behave differently depending on how and when the camera is connected. None of it is complicated once you know what to expect, but a few details aren't explicitly documented by Hasselblad, so the descriptions here are, for the most part, based on observed behavior rather than official specifications.

Note that in Phocus v4 3FR images can be edited directly, without conversion to FFF

The goal in this section is simply to set the stage:

How do the files get into Phocus, and in what form?

Getting Files Off the X2D II and Into Phocus Without the Confusion

Phocus can work with image files from the X2D II in a few different ways, and the method used affects what appears in macOS, which file format Phocus works with, and how things behave later in terms of adjustments and sidecar usage. In everyday use, the camera's files typically reach the Mac in one of the following ways:

- **CFexpress card mounted directly in macOS using a Cfe Card reader**
- **The camera connected via USB in Mass Storage mode**
- **The camera connected while Phocus is already open, resulting in a tethered-style connection**

When a CFexpress card is inserted and the camera is connected in **Mass Storage** mode, macOS usually mounts **two separate volumes** - one for the internal SSD and one for the CFexpress card - and Phocus can import 3FR files from either source. In contrast, when the camera is connected while Phocus is open and the camera enters a tethered communication state, macOS does not mount any volumes, and Phocus instead offers FFF files for import.

The following sections simply describe what to expect when using each method, based on observed behaviour in Phocus 4.0.1 on macOS.

Importing From a CFexpress Card and What Usually Happens

This is the most predictable path, and it behaves exactly as you'd expect. It is also potentially the fastest way to import files, dependent on the speed of the card reader you use, as well as the speed of the ports on your Mac.

What usually happens:

- macOS mounts the CFexpress card as a normal drive.
- Phocus sees and imports the original .3FR files.
- It reads:
 - the embedded JPEG preview
 - EXIF data
 - rating/metadata
- Then applies your chosen **import preset** (Standard, Nature, etc.) as soon as the file is displayed inside Phocus.

At this stage, Phocus hasn't written anything to a .phos sidecar yet.

The preset is applied inside the application, not stored in metadata: I'll get deeper into that in Section Two.

CFexpress is the most straightforward import source, which is why a lot of people use it even when the camera has internal storage available.

Importing From the X2D II Over USB-C and How It Differs From Using a CFexpress Card

The X2D II's internal SSD is very fast, and Phocus can import directly from it, albeit at USB 3.2 Gen 2 speeds, and assuming you're using the Hasselblad-supplied USB-C cable. And again the caveat that it also depends on the speed of the ports on your Mac that you connect it to.

The behavior depends a little on how the camera connects and how Phocus is launched. I couldn't find this explicitly documented by Hasselblad, so the explanation here is based on what I observe to be consistently happening in normal use.

A workflow that reliably exposes the SSD and inserted CFexpress card as drives:

1. Quit Phocus completely.
2. Connect the X2D II to the Mac via USB-C.
3. The camera typically shows a “**Mass Storage?**” prompt.
4. After tapping it, macOS mounts the internal SSD and inserted CFe card as two separate drives.
5. Launch Phocus **after** the volume appears in Finder.
6. Phocus then imports **true 3FR files**, just like a CFexpress card.

Understanding the Mass Storage Prompt When Phocus Is Already Open

In some cases, the **Mass Storage** prompt still appears even when Phocus is open.

Tapping it in that state doesn't seem to change how Phocus interacts with the camera on the machines I've tested: it still connects to it in tethered mode, not mass storage mode.

The one pattern that does hold up well is:

When the camera mounts as a drive before Phocus launches, Phocus reliably imports the original 3FRs.

That sequence is simple enough to use consistently if 3FR import is the goal.

Importing When the Camera Is Already Connected and Why Files May Appear as FFFs

If the camera is connected while Phocus is already open, the behavior shifts in a noticeable way:

- Neither the internal SSD or inserted CFe card mount in Finder.
- Phocus presents the camera like a tethering source than a storage device.
- The import dialog offers **FFF files** instead of 3FRs.

What the FFFs appear to be:

Hasselblad hasn't publicly documented the exact mechanics, that I can find, but based on some testing:

- These FFFs are **not** the camera's original files.
- Instead, they appear to be containers Phocus generates from the image data it receives over the tethering/communication protocol.
- They don't appear to include the embedded JPEG preview or the exact metadata structure of a native 3FR.

Nothing about this is necessarily "wrong", it's just different from importing the original files straight off the SSD, and some folks actually prefer working with FFF files.

Why this matters

If you're expecting original 3FR files and instead end up with FFFs, it helps to understand that Phocus seems to be interacting with the camera in a different mode. The only visible indication in the UI that you're in tethered mode is that it won't display previews directly from the RAW files on the card/SSD, referring to them, somewhat confusingly, as 'legacy' files.

How to Ensure Phocus Imports Your Files as 3FR Instead of FFF

If the intention is to work directly with the original 3FR files recorded on the camera's SSD, the following order seems to be the most reliable:

1. Quit Phocus.

2. Connect the X2D II.
3. Tap **Mass Storage**.
4. Wait for the SSD and inserted CFe card to mount as drives in Finder.
5. Launch Phocus and import.

This sequence avoids the tethered-import behavior and uses the SSD as a normal storage device.

Why Choosing Mass Storage vs Media Transfer Protocol (MTP) Changes What Phocus Imports

Mass Storage mode (SSD & inserted CFe mounted):

- Camera typically shows “Mass Storage?” prompt
- Drives appear in Finder
- Phocus imports the original .3FR files
- Behavior matches CFexpress import

Tethering-style mode (SSD not mounted):

- No drives appear in Finder
 - Phocus sees a connected camera rather than a storage device and does not display previews
 - Imports appear as .FFF files
 - File contents differ from the original 3FRs
 - Useful for tethered workflows, but different from a file-based import
-

Why Getting the Import Right Matters for HNCS, HNNR, and the Rest of the Workflow

Everything that comes later — how presets get applied, what gets stored in sidecars, how consistent edits are across machines — depends on whether Phocus imported a 3FR or generated an FFF.

Section Two will attempt to build on this by looking at what actually gets applied at import time (and what doesn't), and how much of that is written to the sidecar file versus kept inside Phocus itself.

What Actually Gets Applied at Import (and What Doesn't)

Once a 3FR file appears inside Phocus, regardless of whether it came from a CFexpress card or the internal SSD, Phocus immediately applies some interpretation of the image. Some of this is visible and obvious, while other parts happen quietly behind the scenes. It's also not always clear what Phocus is **saving** versus what it's simply **showing** you, so this section focuses on separating those two ideas.

The short version is that Phocus does a lot at import time, but it doesn't write much of it anywhere permanent.

Choosing an HNCS Preset at Import and Which One Makes Sense for Most Images

When you select a preset in the import dialog, Phocus applies that preset the moment the image first appears in the viewer. This includes:

- the HNCS tone curve
- the HNCS colour rendering
- any preset-specific bias in contrast or saturation

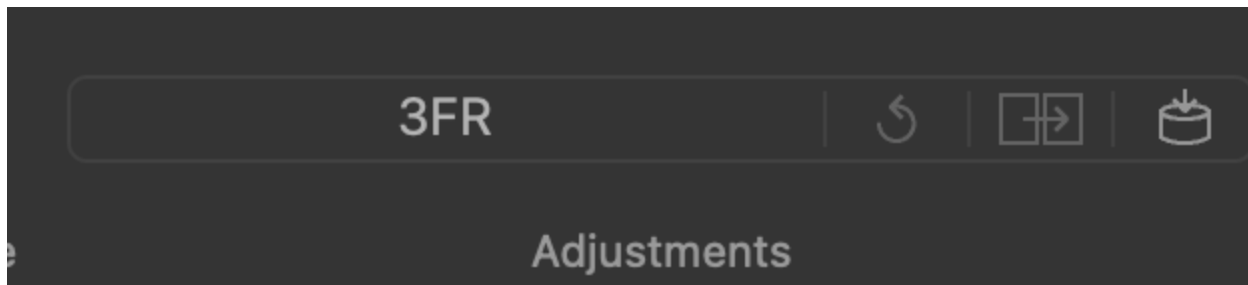
These changes are real in the sense that they affect what you see and how the file will export later.

However:

- none of this gets written to a .phos file
- there is no metadata tag indicating "Standard was applied at import"
- switching to a different preset later also doesn't write anything into a sidecar

Phocus appears to treat HNCS presets as the **initial interpretation** of the RAW rather than an edit that should be preserved.

This makes the import preset more of a rendering-time decision than a setting that becomes part of the file's history.



Why Phocus Still Shows “3FR” After Choosing an HNCS Preset (and What It Really Means)

After import, the adjustments dropdown at the top of the Phocus window typically reads:

3FR

This doesn’t mean “no preset was applied.” It simply reflects that the adjustment set is tied to a 3FR file. The label does not update when a preset is chosen or changed.

From observing the behaviour, it seems this dropdown is closer to a technical identifier than a user-facing label. It doesn’t appear to track presets at all, and it isn’t attempting to communicate the current HNCS profile.

All in all it feels very counter-intuitive.

What Phocus Applies in Memory but Doesn’t Save Right Away

This is one of the more unintuitive parts of the workflow:

Changing the import preset does not create or modify a sidecar file.

The .phos sidecar remains a pure metadata container until you make an explicit edit — meaning an actual slider change, crop, rotation, or similar action.

What **doesn’t** get saved at this point:

- tone curve from the preset
- HNCS profile choice
- any preset-defined rendering parameters
- anything applied automatically at import
- anything that affects only the internal rendering engine

None of these appear in the sidecar until an actual adjustment is made.

What Actually Triggers Phocus to Write a Sidecar File

The moment you change anything Phocus considers an 'adjustment,' Phocus marks the image as having unsaved changes. If 'Always Save' is enabled in Preferences (the default), these changes are written to the .phos sidecar automatically when you navigate to another image. The types of changes that trigger this include:

- exposure
- white balance or tint
- dynamic range sliders (highlights, shadows)
- clarity and structure
- noise reduction
- cropping or straightening
- perspective corrections

These values are written as explicit deltas from Phocus' internal baseline, which is why the .phos file remains empty of adjustments until you move at least one control.

Phocus has an 'Always Save' preference (enabled by default) that automatically writes your adjustments to the sidecar when you navigate away from an image. You don't need to manually save for your edits to persist. However, if you want to create a reusable adjustment set - one that appears in the Embedded list and can be applied to other images via 'Use Last Saved Adjustments' - you need to explicitly click 'Save Changes' in the toolbar (or use Image > Save Adjustments from the menu).

Why the Image You See in Phocus May Not Match What's Actually Saved

Phocus' viewer always reflects the **current** rendering engine state. That state includes:

- the chosen import preset
- embedded JPEG preview (initial thumbnail only)
- all default RAW interpretation parameters
- any slider changes you've made

Even though the import preset isn't written anywhere, it still affects this internal rendering. That's why the scene looks different if you swap between Standard, Nature, or any other HNCS profile.

However, until a sidecar-worthy adjustment is made, none of these rendering choices exist outside Phocus' internal state.

How Phocus Bakes In Your Import Preset Even If Nothing Has Been Saved Yet

If you export a TIFF without touching any sliders:

- the chosen import preset **is** reflected in the export
- the TIFF contains the look you see on screen
- the exported file retains Hasselblad's colour profile if Output Profile = "Source" is used
- the preset is effectively "baked in" despite not existing in the sidecar

This creates a situation where:

- The preset isn't recorded anywhere in metadata
- But the exported file visually reflects it perfectly

Phocus seems to treat export as the moment when the rendering becomes permanent.

How Phocus Separates What You See From What It Actually Saves

Applied immediately (but only in memory):

- HNCS profile from the chosen preset
- Colour and tone interpretation
- The embedded preview
- The baseline Hasselblad rendering pipeline

Not saved at import — and never saved as metadata:

- The HNCS preset you chose
- Any preset-derived rendering decisions
- Tone curve selections
- Anything relating to the internal render state

Only saved after you adjust something (via sidecar):

- Exposure
- White balance
- Crop
- Clarity
- Dynamic range tools
- Noise reduction
- Perspective tools

With the default 'Always Save' preference enabled, these adjustments are written to the sidecar automatically when you navigate away from the image. No manual save is required for persistence. The 'Save Changes' button serves a different purpose: it creates a named entry in the Embedded adjustment list, which you can later revert to or apply to other images.

The core idea:

Phocus treats HNCS presets as part of its internal rendering pipeline, not as saved edits.

They shape what you **see** and they're baked into exported TIFFs, but they never exist in any metadata or sidecar, before or after export.

Their only “trace” is the visual rendering itself.

How the Adjustments Dropdown in Phocus Changes Meaning at Each Stage

One part of the interface that can be confusing at first is the **Adjustments** dropdown at the top of the Phocus window. It doesn't behave like a simple preset selector. Instead, it changes meaning depending on what stage the image is in.

Right after import

- The dropdown shows “**3FR**”
- A preset may have been applied during import
- The preview reflects that preset
- But there is **no sidecar yet**
- And there is **no record** of which preset was used

At this point, “3FR” refers to the source file type, not the rendering state.

If you change the preset before saving

- The dropdown shows the preset name
- The rendering updates to match it
- But the preset is still **not written** to a sidecar

This is the only moment where the dropdown acts like a preset selector.

After saving or making the first adjustment

- The dropdown changes to “**Edit...**”
- The .phos file is created or updated (automatically, if "Always Save" is enabled)
- Only explicit slider values are recorded
- The preset itself is **not stored anywhere**

Note that clicking "Save Changes" doesn't change what's written to the sidecar - it creates an entry in the Embedded list, which is a separate concept from sidecar persistence.

So the dropdown has now shifted to showing the **current saved adjustment state**.

Switching back to “3FR” after saving

This now truly reverts the rendering to the RAW baseline with no preset applied — which looks different from “3FR” immediately after import. This is where the behaviour feels inconsistent even though it follows Phocus’ internal logic.

The important takeaway

The sidecar never contains the preset name or any indication of which preset will be used for preview or export. The preset affects how the image is rendered in Phocus, but it is not stored as part of the adjustment data.

This also explains why the same .3FR + .phos can look slightly different on another machine unless further edits have been made.

How to Apply an HNCS Preset to Multiple Images in Phocus (and Why It Doesn’t Work Like You Expect)

Selecting multiple thumbnails and then changing preset from say “Standard” to “Nature” and then hitting “Save” does not, in fact, save “Nature” to all those selected, only to the one highlighted.

To change all the highlighted ones you need to use the “Modify” option in the toolbar. There’s a dropdown above the individual settings that will be on whatever you set on import: selecting

“Nature” there and then clicking “Modify” seems to apply it to all the selected thumbnails.

This is unexpected since it differs from the behavior of actions from the right-click context menu. So for example if you select multiple image thumbnails, right click, and select HNNR, HNNR is applied to ALL the selected images, not just the highlighted one.

How Phocus Creates and Uses the .phos Sidecar File

By the time Phocus displays an image, a fair amount has already happened internally: the import preset has been applied, the HNCS rendering is active, and the viewer is showing you a developed version of the RAW. But none of this becomes part of the file’s **record** until Phocus decides there’s something worth writing.

This section focuses specifically on the .phos sidecar file: what triggers it, what it contains, and what Phocus does **not** save inside it.

Everything here is based on a combination of behavior I've observed in Phocus 4.0.1 and direct inspection of real sidecar files.

Understanding 'Save Changes' vs Auto-Save in Phocus

Phocus uses terminology that can be misleading. Here's what the key terms actually mean:

Term	What it actually does
Always Save (Preferences)	Automatically writes your adjustments to the .phos sidecar when you navigate away from an image. Enabled by default.
Save Changes (toolbar button)	Creates a timestamped entry in the Embedded adjustment list, a reusable snapshot you can revert to or apply to other images. Does <i>not</i> control whether your edits persist.
Save Adjustments (Image menu)	Same as "Save Changes", confusingly different name for the same function.
Use Last Saved Adjustments	Applies the most recent Embedded adjustment set to the current image. Only available after you've used "Save Changes" at least once.

The practical upshot: your edits persist automatically. You only need "Save Changes" if you want to create a reusable adjustment set or maintain a revertible history.

Understanding 'Save Changes' vs Auto-Save in Phocus

Phocus uses terminology that can be misleading. Here's what the key terms actually mean:

Always Save (Preferences)

Auto-writes adjustments to the sidecar when you navigate away. Enabled by default.

Save Changes (toolbar button)

Creates a snapshot in the Embedded list for reuse or reverting. Doesn't control edit persistence.

Save Adjustments (Image menu)

Same as "Save Changes" – different name, same function.

Use Last Saved Adjustments

Applies the most recent Embedded snapshot. Only available after using "Save Changes" once.

The practical upshot: your edits persist automatically. You only need "Save Changes" if you want to create a reusable adjustment set or maintain a revertible history.

When Phocus Creates or Updates the .phos Sidecar File

Phocus creates or updates a .phos file only when:

- you change an actual adjustment slider
- you crop or rotate
- you apply noise reduction
- you adjust perspective
- you rate or label the image

The timing of when these changes are written depends on your Preferences setting. With 'Always Save' enabled (the default), the sidecar is updated automatically when you navigate away from the image. With 'Manually' selected, you would need to explicitly save - but this setting is not commonly used.

Until then, the .phos exists only as a metadata container (EXIF/IPTC), but with **no rendering adjustments**

Changing HNCS presets, switching between Standard/Nature, or simply viewing the file does **not** create or modify a sidecar.

This seems consistent across 3FRs, denoised RAWs, and images imported via either CFexpress or Mass Storage mode.

Why Only User Edits (Not Preset Changes) Trigger a .phos Sidecar Update in Phocus

The moment you make an explicit adjustment in the UI other than changing a preset, Phocus begins writing to the sidecar.

Examples include:

- Exposure
- White balance
- Tint
- Highlight and shadow recovery
- Contrast and clarity
- Noise reduction sliders (note: NOT HNNR!)
- Cropping
- Straightening or rotating
- Keystone/perspective tools

These changes show up as entries in the XMP portion of the .phos file. They're written as "deltas" — changes relative to Phocus' baseline interpretation — not as full descriptions of the image's state.

Once you make at least one adjustment, the file will contain explicit Phocus adjustment tags.

What's Actually Inside a .phos Sidecar File (and What Isn't)

A typical .phos file has two conceptual layers:

Metadata (always present)

This includes:

- EXIF
- IPTC creator/contact info
- Camera/lens metadata
- Rating tags

- Timestamp data

This part appears even when no adjustments have been made.

Adjustment Data (only present after edits)

This includes:

- exposure delta
- white balance values
- crop geometry
- perspective transforms
- noise reduction settings
- clarity/structure
- other Phocus tool values

Missing entirely — even after adjustments — are:

- HNCS preset names
- import preset information
- tone curve selection at import
- any metadata indicating “this image was imported using Nature”

Those choices are part of the rendering engine’s state, not the sidecar.

The .phos file is best understood as a record of **your manual changes**, not the entire conceptual state of the image.

How Phocus Rewrites the .phos Sidecar Over Time (and Why It Never Stores a History)

Some practical behaviors worth noting:

- Each time you save adjustments, Phocus rewrites the .phos file with updated values.
- Undoing edits may remove certain entries or set them back to defaults.
- The timestamp on the file doesn’t always reflect meaningful changes (as we saw when switching presets).
- If the sidecar is deleted, Phocus reverts to interpreting the RAW using its baseline rendering (which includes the import preset the file was last displayed with).

A sidecar does **not** accumulate a “history” the way Lightroom does, it simply represents the **current** adjusted state.

Why RAW + Sidecar Files Can Look Different on Another Machine in Phocus

If like me you work on multiple machines, some of this matters. I sync my working files and final outputs between a Mac Mini, my NAS, and MacBook Pro using [ResilioSync](#) so that I'm not tied to my Mac Mini.

This is a direct consequence of everything in Section Two:

- The .phos file does **not** store which HNCS preset was applied at import.
- The .3FR file does **not** store it either.
- Therefore, if you move .3FR + .phos to another machine, that machine will interpret the file using **its own current baseline rendering**, not necessarily the same preset you used.

What *is* synchronized:

- exposure/WB/clarity/etc.
- crop/perspective
- ratings
- metadata

What is *not* synchronized:

- the choice of Standard vs Nature vs Portrait
- any preset-based tone curve
- the import preset you originally chose
- any implicit rendering engine state

This is why two machines can show slightly different interpretations of the same RAW+sidecar combination, unless adjustments were explicitly written.

If consistency across machines matters, exporting a TIFF is the only guaranteed method to preserve the exact look.

How HNNR Creates a New RAW and Sidecar Pair in Phocus

I'm saving the full HNNR deep-dive for later, but here's the relevant part for sidecar behavior:

- Running HNNR creates a **new RAW file** (filename_denoised.3FR).
- Phocus pairs it with a **new** .phos file matching the new name.
- The original .phos remains untouched for the original 3FR.
- The new .phos for the denoised file behaves like any other:
- - no HNCS preset recorded
 - adjustments written only after sliders change
 - identical structure to non-denoised sidecars

The denoised-and-original files are effectively two separate RAWs with independent sidecars.

What All This Means for Understanding Sidecars in Phocus

- **Sidecars only exist to record explicit edits**, not Phocus' internal rendering state.
- Import presets, Standard/Nature choices, and other HNCS decisions do **not** appear in the sidecar.
- Phocus only writes adjustment deltas, not a complete history.
- Moving .3FR + .phos between machines may result in **different renderings** if no edits have been made.
- Exporting is where the internal render becomes permanent.

This makes sidecars relatively simple files, but it also means Phocus workflows depend heavily on understanding the line between “visible adjustments” and “saved adjustments.”

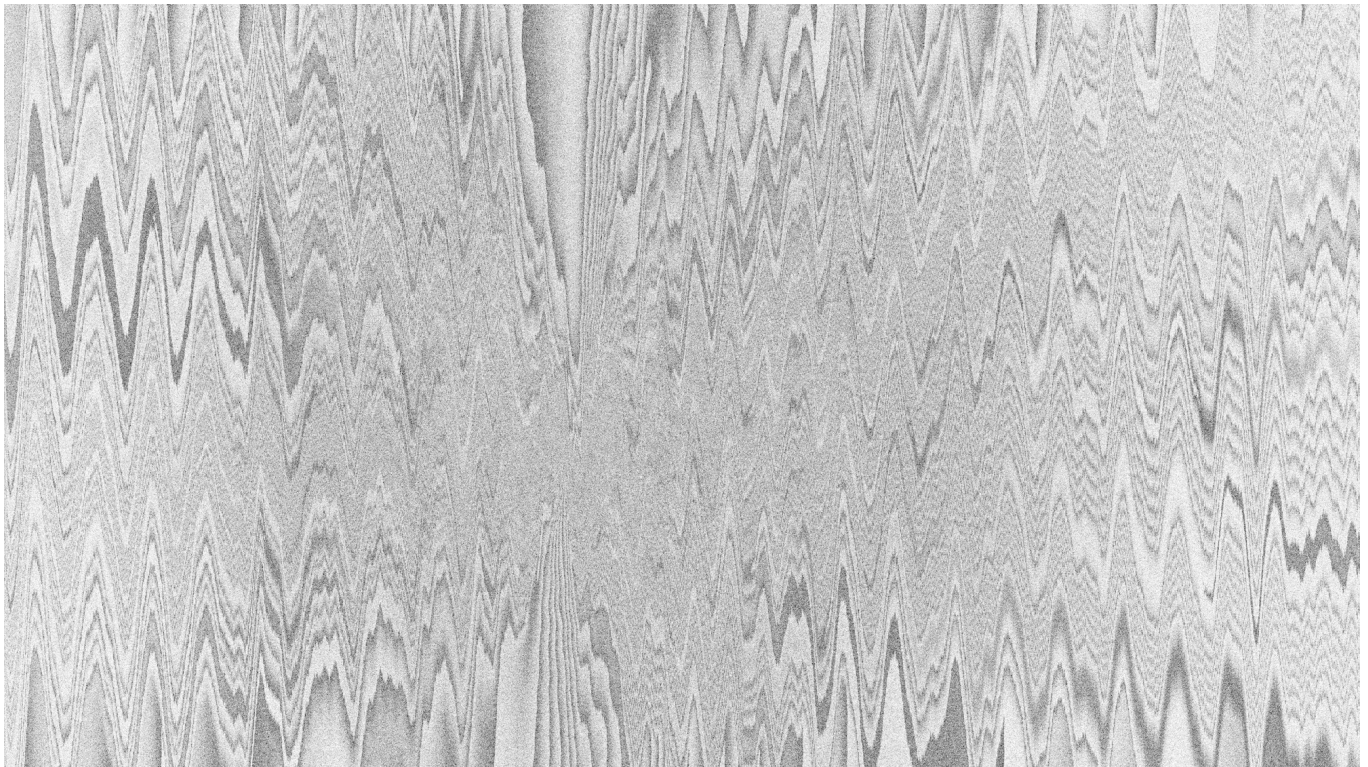


Photo by [Logan Voss](#) / [Unsplash](#)

Understanding Hasselblad Natural Noise Reduction (HNNR)

HNNR is a newer addition to Phocus that behaves quite differently from the normal adjustment tools. Instead of functioning like a slider, HNNR operates directly on the RAW data and produces a **new** RAW file as its output. It sits outside the standard import workflow, but many photographers apply it very early in their process, so it makes sense to document it alongside the other “initial state” behaviours.

This section describes what HNNR appears to do in Phocus 4.0.1 on macOS, how it behaves at the file level, and what to expect in real use.

Note that Hasselblad state HNNR is RAW-only and model-limited; helpful for explaining why it doesn’t appear on all files.

Where HNNR Lives in Phocus and How You Actually Run It

HNNR is oddly not part of the adjustments panel.

There’s no button for it in the toolbar or an option in the import dialog.

Instead, it's accessed via:

- Right-clicking any thumbnail or a selection of multiple thumbnails
- Selecting “**HNNR**”

This brings up no on-screen dialogue or slider. It simply runs. Be aware that Phocus becomes completely unresponsive while this is happening except for the progress slider. Even on top-end Macs it's not super fast, so if you're doing a batch, you're going to be in with a bit of a wait.

HNNR only seems to be available as an option in the context menu on **Apple Silicon** (M1 and later) Macs.

There doesn't seem to be a CPU-only fallback mode for Intel machines.

HNNR appears to use the **Apple Neural Engine** to run some kind of machine-learning-based noise reduction on the demosaiced RAW data.

What Phocus Creates When You Run HNNR (New RAW and New Sidecar)

Applying HNNR does not modify your original RAW.

Instead, Phocus:

1. Generates a **new RAW 3FR file**
2. Appends “**_denoised**” to the filename
3. Creates a **new .phos sidecar** matching the new filename
4. Leaves the original .3FR (and its sidecar, if any) untouched

For example:

```
2025-11-21_01_0001.3FR
2025-11-21_01_0001.3FR.phos

→ after HNNR:

2025-11-21_01_0001_denoised.3FR
2025-11-21_01_0001_denoised.3FR.phos
```

The new RAW seems to be roughly the same size as the original, which suggests HNNR rewrites the entire RAW container rather than storing deltas.

The new RAW behaves completely normally in Phocus — it's simply a cleaner version of the same scene. Just make sure you select the denoised RAWs when exporting

What's Actually Inside the HNNR RAW File (and Why It Looks Like a Normal 3FR)

Based on how these files seem to behave in practice:

- Metadata is preserved
- Lens and EXIF information are intact
- The file behaves exactly like a camera-generated 3FR
- The sidecar remains empty except for the base data until you make explicit adjustments
- No metadata flag identifies the file as HNNR-processed

The only visible indicators of HNNR are:

- the filename
- the visibly reduced noise
- and the existence of a separate RAW file

Phocus doesn't present HNNR as an "edit": it presents it as a new capture.

How HNNR Changes Your Workflow, Storage, and File Management

Because HNNR duplicates the RAW, it changes how you handle:

- storage
- syncing
- versioning
- and downstream editing (e.g., moving into Capture One)

Some things worth noting:

Two independent RAWs

- The original and the denoised 3FR are completely separate.
- Edits applied to one do not sync to the other since it's the raw data that is denoised.

File count doubles

- If applied broadly, HNNR doubles your RAW storage footprint.
- Sync systems like Resilio will treat each denoised file as a new full transfer.

Choose when to apply it

HNNR makes the most sense:

- right after import
- before doing any exposure or colour work
- before exporting to a different editor
- especially with shadow-heavy or high-ISO images

Since it creates a brand-new RAW, it doesn't interfere with your sidecar workflow.

How Fast HNNR Runs on Different Macs (and Why Batch Processing Is Slow)

HNNR is fast for what it does, but speed varies depending on the machine and the capabilities of the Neural Engine on the specific generation and version of the M processor in it:

- M1 → noticeably quick
- M1 Pro/Max → significantly faster
- M2 - M5 chips → even better throughput
- Not available on Intel Macs

Applying it to multiple images at once takes a lot of time, even on an M4 Pro chip.

Key Takeaways for Using HNNR in Your Workflow

- HNNR is applied via right-click, not a slider.
- It requires Apple Silicon and uses the neural engine.
- It creates a **new** RAW file (_denoised.3FR).
- The original RAW remains unchanged.
- A matching new sidecar is created.
- No metadata indicates HNNR was used, the filename is the only hint.
- Good to run early in the workflow before heavy edits or exporting to Capture One.

- Doubles storage for each file processed.