

THE ARTIST COMPACT

A public standard for creative tools

Creative tools should increase an artist's capability without increasing their captivity.

WORKING DRAFT / V007

WILD CONSTRUCT / SEPTEMBER 2026

A PUBLIC STANDARD FOR CREATIVE TOOLS

Creative tools should increase an artist's capability without increasing their captivity.

The Artist Compact is a public standard for how creative tools should behave in real production, what the companies that build them owe the people who depend on them, and what artists owe the craft in return. It speaks both to traditional post-production software tools and the new world of generative tooling as well.

Wild Construct proposed the first draft of this Compact. We expect it to change, and we expect it to outgrow us.

- **01 DIRECTABLE**
- **02 EMBODIED**
- **03 DECOMPOSED**
- **04 PERSISTENT**
- **05 LOW-FRICTION**
- **06 ACCESSIBLE**
- **07 LEGIBLE**

AT A GLANCE

Seven properties. One vitality test. Commitments on both sides.

Creative software is not disposable infrastructure. Artists build practices around it. Studios build pipelines around it. Projects depend on it reopening tomorrow, next year, and sometimes a decade from now.

That creates obligations.

A good creative tool should make an artist more capable: more able to direct, experiment, revise, understand and preserve their work.

A good creative-software company should continue earning its place by creating new capability rather than increasing its leverage over capability artists already depend on.

The Compact provides three things:

- **Seven properties** any creative tool can be tested against.
- **A vitality test** for whether a company is continuing to create meaningful artist capability.
- **Reciprocal commitments** for the people who build tools and the people who use them.

This is not a law. It is a yardstick.

If you only read one thing, read the seven Ask questions.

THE METHOD

Heritage studied, not sanctified.

Professional creative software is the accumulated record of decades of design decisions.

Timelines exist because people think in sequences. Layers exist because complex images need to come apart. Keyframes and curves exist because artists need to control change over time.

Non-destructive editing exists because creative work is iterative and reversible.

These aren't simply interface conventions. They contain knowledge about how people make things.

So before replacing a workflow, ask why it exists.

What problem was it solving?

Does that problem still exist?

If it does, can a new tool solve it better while preserving what artists learned along the way?

But study is not worship.

Creative software carries accumulated debt alongside accumulated wisdom. New technologies have made some of that debt easier to see.

- **Time-to-first-result matters.** Many professional applications still greet newcomers with a wall.
- **Semantic addressing is real.** "Select the sky" can be a better control than manually describing the same region pixel by pixel.
- **Low-friction optionality changes creativity.** Undo lets us go backward. Modern systems can also let us ask, "show me eight versions," before committing to one.
- **A blank canvas can also be a blank void.** Starting from something can be better than starting from nothing.

This cuts both ways, and it should.

New tools should learn from decades of craft.

Heritage tools should learn from what new technology has proved possible.

Neither gets to claim the future by ignoring the other.

THE SEVEN PROPERTIES

A testable standard for creative tools

The technology will change. These questions should survive longer than any particular implementation.

01 DIRECTABLE

The artist holds authorship.

A creative tool exists to help someone make decisions.

It can propose, automate, simulate, generate, interpolate, organize and surprise. It can remove work the artist does not want to do and expose possibilities the artist would not have found alone.

But the consequential creative decisions remain steerable.

Defaults are useful. Defaults are not direction.

A good tool makes it clear which decisions the artist is making, which decisions the system is making, and how to change either one.

When an artist corrects something, the correction should hold wherever the workflow reasonably allows it.

ASK Can I steer this system toward and away from specific results, and when I correct it, does the correction hold?

02 EMBODIED

Use the control surface that fits the decision.

Natural language is a powerful new control surface. It is not the only one.

Artists already work through timelines, layers, nodes, keyframes, curves, rigs, handles, cameras, gestures and direct manipulation. Their hands carry knowledge too.

A good creative tool uses the interaction appropriate to the decision.

Sometimes that means typing what you want. Sometimes it means dragging a point. Sometimes it means drawing a curve, adjusting timing, selecting something by meaning, moving through space or asking an agent to do the tedious part.

New interaction models should expand the control stack, not flatten it into a single interface.

Progressive disclosure matters too. A useful first result should not require mastering the whole system. Depth should be available when the artist needs it, not mandatory before they can begin.

ASK Can I control this in ways appropriate to the decisions I'm making, and go deeper when I choose to?

03 DECOMPOSED

Parts, not meals.

Creative work needs to come apart.

Layers. Passes. Stems. Tracks. Characters. Objects. Masks. Takes. Parameters. Components.

A music tool should let you get to the stems, not only the final mix. A design tool should preserve live type rather than handing back rasterized letters. A generative system should expose useful elements rather than treating the final image as the only artifact that matters.

Artists need to extract, swap, refine and replace parts without destroying everything around them.

You dress the set before you roll. You rough in the comp before you refine it. Starting from something beats starting from nothing, but the something has to come apart in your hands.

ASK Does this tool give me ingredients, or only a finished meal I cannot edit?

04 PERSISTENT

Creative decisions should survive.

Production is not one result. It is revisions, collaborators, versions, shots, episodes, deliverables and projects over time.

The decisions an artist has already made should remain useful as the work moves forward.

A project should reopen. A parameter should still mean something. An identity established in one shot should be reusable in another. A correction should not disappear simply because the system produced another version.

A newer tool should provide a path forward for work created in an older one wherever technically reasonable.

Persistence does not mean freezing every implementation forever. It means treating existing creative decisions as valuable state rather than disposable residue.

ASK Does this system preserve what I've committed to, or make me reconstruct it every time the context changes?

05 LOW-FRICTION

Creativity runs on loops: idea, test, judge, refine.

The fun part is changing your mind.

The easier it is to try another meaningful version, the more room an artist has to explore.

Friction can be render time, setup, destructive workflows, compute, metering, licensing, cognitive load, or simply a system that makes every experiment feel too precious.

A good creative tool reduces the distance between having an idea, seeing what it does, judging it, and trying something else.

The goal isn't speed for its own sake. It's preserving the creative loop.

ASK How much friction stands between an idea and the next meaningful attempt?

06 ACCESSIBLE

People need a real way in.

A tool cannot become part of someone's craft if they cannot meaningfully experience it.

Demos should teach. Trials should demonstrate real behavior. Documentation should show ordinary work as well as hero work. Simple workflows should be approachable without hiding the depth professionals eventually need.

Where possible, free tools should be useful things rather than deliberately broken advertisements for paid ones.

Licensing should reflect how creative work actually happens: collaborators come and go, artists work remotely, studios scale up and down, people change machines and projects cross organizational boundaries.

Access does not mean everything must be free. It means the path into the tool should help someone learn whether it belongs in their practice.

ASK Can someone meaningfully learn and evaluate this tool before committing to it, and does access reflect how creative work actually happens?

07 LEGIBLE

Professional tools leave enough evidence behind.

When something breaks at 3am before a deadline, "the system just does that sometimes" is not an answer.

Artists need enough information to understand what happened: inputs, parameters, versions, transforms, dependencies, actions and state.

Not every implementation needs to be exposed. Trade secrets can remain trade secrets. Complexity can remain under the hood.

But professional behavior needs to be legible enough to debug, reproduce, revise, explain and defend.

This applies to procedural systems, simulations, automation, agents and generative models alike.

Secret sauce is not an excuse for an unknowable black box.

ASK When something breaks or gets challenged, can I trace what happened well enough to fix, explain or repeat it?

THE VITALITY TEST

A good tool is not enough.

A company can make an excellent tool and then spend years extracting more value from the dependency it created.

So the Compact needs another question:

Is this company continuing to create new artist capability, or primarily increasing its leverage over capability artists already have?

Commercial success and artist benefit are not opposites.

Software costs money to build. Updates cost money. Support costs money. Cloud infrastructure costs money. New products cost money.

A healthy creative-software company should be able to charge for all of those things.

The question is what the customer receives in return.

We use three lenses for thinking about meaningful new capability:

RESTORE

Take an important creative workflow that has become stale, neglected, cumbersome or unnecessarily expensive and make it materially better.

Restoration is innovation when it changes what the artist can practically do.

INVENT

Create a meaningful new capability or interaction that expands what an artist can express, control or accomplish.

Novelty alone does not qualify. It has to survive contact with real work.

CONNECT

Preserve and transmit useful creative state where it used to be flattened, trapped or repeatedly reconstructed.

Between tools. Between stages. Between collaborators. Between versions. Between the artist and the systems working on their behalf.

A founder demo proves possibility. An instrument proves itself when people outside the company use it to make real work.

CONTINUITY

What people have, they keep.

Creative tools become infrastructure for individual artists long before they become infrastructure in any technical sense.

People learn them. They build projects around them. They develop muscle memory. They create archives that depend on them.

That investment matters.

A perpetual purchase should not quietly become a rental.

A free tool placed in someone's hands should not later be remotely taken away.

A product reaching the end of active development should not automatically mean the work made with it becomes inaccessible.

No company can promise that an old binary will run on every future operating system, host application or processor forever.

But companies can make choices about continuity.

They can preserve installers. They can document compatibility. They can provide migration paths. They can avoid unnecessary remote kill switches. They can give artists notice before changes put work at risk. They can spend some of the falling cost of software development on keeping useful capability useful.

The obligation is not immortality.

It is stewardship.

RECIPROCAL COMMITMENTS

A compact binds both sides.

What tool builders commit to

You can leave with your work. Provide practical, documented ways to export and preserve useful creative state in the formats production actually speaks. Do not build a moat out of customer files.

What people have, they keep. Do not revoke paid capability merely because a newer commercial model is more convenient. Sunset is not confiscation.

Bundles earn their place. A suite can create integration, convenience and savings. It should not become the only way to obtain an instrument that can reasonably stand on its own.

Your work is yours. Do not train on customer work without explicit, uncoerced permission.

Future revenue comes from future value. Charge for new capability, meaningful updates, services, support, infrastructure and genuinely useful integration. Do not make dependency itself the product.

Demos should teach. Show real behavior. Let people understand what they are buying.

Serve the long tail. As the cost of creating software falls, "too niche to build" should become a less acceptable answer to real creative needs.

What artists commit to

Use the tool enough to judge it. Critique from practice rather than category assumptions.

Demand better, specifically. Ask vendors what happens to your files, your licenses, your projects and your workflow. Reward better behavior with money, credit and advocacy.

Teach the craft forward. Interfaces change. The knowledge behind them should not disappear when they do.

Tell toolmakers what production actually needs. The people making software cannot preserve craft knowledge they never hear about.

Disclose honestly. Be clear with clients and collaborators about the role of automation or generative systems when it materially matters to the work or agreement.

HONEST ACCOUNTING

Every technological shift gives and takes.

New tools create capability.

They also redistribute work, change skills, eliminate tasks and create new dependencies.

The people who lose from those changes are not always the people who gain from them.

We should be able to acknowledge both things at once.

Progress is not measured by the amount of labor removed from a process.

It is measured by whether artists become more capable of doing meaningful work, whether new people can enter the craft, whether knowledge survives the transition and whether the people using the tools retain meaningful agency over what they make.

The goal is not to preserve every old job, workflow or interface.

The goal is to preserve and expand human creative capability while the tools change underneath it.

HOW THIS COMPACT LIVES

Open, testable, revisable.

Open to anyone. Artists, studios, educators, tool builders and researchers can use the Compact. No membership is required.

Owned by no one. Wild Construct proposed the first version. The Compact should not become an endorsement mechanism for Wild Construct or anyone else.

Tested through behavior. A company does not satisfy the Compact by saying the right things. Products, licensing decisions, sunset policies, exports, support practices and actual production use are the evidence.

Built for change. Technology will change faster than craft. The Compact should preserve useful principles without pretending today's implementation is permanent.

Revised in public. New workflows will expose weaknesses in this document. That is expected. Critique and revision are part of the standard.

A NOTE FROM WILD CONSTRUCT

Hold us to it.

Wild Construct proposed the Artist Compact because we needed a standard for our own decisions.

We build creative tools, which means every product decision eventually becomes a relationship with the people who use it.

The seven properties describe what we think an individual tool should provide.

The Vitality Test asks whether we are continuing to deserve the customer's business.

Inside Wild Construct, we use a related Toolchain Vitality framework to evaluate whether our work is restoring important capability, inventing meaningful new capability or connecting creative state that used to be trapped.

The public Compact is the promise.

The internal framework is one way we hold ourselves accountable to it.

Neither matters if our behavior contradicts the words.

Judge the products.

Judge the pricing.

Judge what happens when something changes.

Judge whether your work remains yours.

Judge whether the next thing we ask you to pay for actually makes you more capable.

That is what this document is for.