

Weaver Series: Comprehensive Reference Guide

Last Updated: 2026/08/28



Tip

For Users Who Want to Animate Characters Right Away

This document is a comprehensive technical reference guide covering the entire architecture, design philosophy, and core specifications (CSV/JSON configurations, etc.) of the system.

If you want to jump straight into animating your own character using the provided Krita templates (*.krz , etc.) without worrying about complex specifications, please refer to the "Ultra-Intuitive Quick Start Guide" at the beginning of **11_LayerWeaver: Beginner's Drawing Guide**.

1. Design Philosophy and Core Concepts of the Weaver Series

1.1 Why This Tool Was Born: "Design Philosophy" & "Trial and Error" from a Self-Taught Creator

The Weaver Series is a production pipeline born from countless trials and errors by a single creator without formal animation training, driven by an earnest desire: "To complete high-quality animations entirely on my own, while preserving the intuitive warmth of hand-drawn artwork."

I am neither a professional animator, a professional illustrator, nor a 3D modeler. Without foundational schooling in animation, the greatest walls I encountered when attempting to create anime solo were: "How can I create beautiful movement?", "How can I prevent drawing inconsistencies (off-model errors)?", and above all, **"How can I build perspective-accurate backgrounds, props, and 3D spaces?"**—severe technical bottlenecks for an independent creator.

To overcome this, I developed a mechanism to "weave" together the wonderful existing tools at hand—**Moho, TDPT, Krita, CTA5, Poser, and Blender**—during solitary work, enabling creators to focus their creative passion solely on artistic expression with less effort and more joy. That is how the Weaver (one who weaves together) Series was born.



Note

Origins of the G3 Template Standard

The "G3 Template Standard" supported by this system is an exceptionally refined existing specification originally designed by **Reallusion**, the developers of Cartoon Animator 5 (CTA5). The Weaver Series did not invent this magical template standard from scratch; rather, it functions as a "translator" to seamlessly guide and weave your artwork into this existing, brilliant standard.

If you are hesitating because you want to create solo but feel blocked by technical hurdles, this tool is your ally. Weaver is a robust bridge built between the "geometric consistency" of 3D and the "free, expressive stylization" of hand-drawn 2D animation.

1.2 The Data Baton Relay: Studio-Division Philosophy Across 6 Major Tools

Weaver is not a single monolithic application, but a "division-of-labor studio" where 6 specialized tools pass data like runners in a relay race. Each tool focuses strictly on what it does best, while the system bridges all complex mathematics and data conversions.

```
graph TD
    subgraph "① Acting & Motion Preparation"
        BVH["<b>3D Motion (BVH/TDPT)</b><br>Motion Capture Data"] --> MW["<b>MotionWeaver (Hub)</b><br>3D Physi
        MW --> BW["<b>BatchWeaver (Multi-Angle)</b><br>Batch Auto-Export: Front, Side, High/Low Angles"]
    end

    subgraph "② Character Drawing & Part Extraction"
        Krita["<b>Krita (LayerWeaver)</b><br>@Head / >Scarf / j12* Tagged Drawing"] --> LW["<b>LayerWeaver (Extr
    end

    subgraph "③ Character Integration (Moho)"
        BW --> RW["<b>RigWeaver (Moho)</b><br>Auto Rigging, Bone Binding & Motion Injection"]
        LW --> RW
        RW --> SS["<b>SmartSwitch (Moho)</b><br>Bind Inheritance, Part Unitization & Vector Cleanup"]
    end

    subgraph "④ Environment & Prop Modeling (Moho)"
        FW["<b>FoldWeaver 3D</b><br>Draw Flat & Fold (Shift+A)<br>Summon Corner to Origin for Frontal Drawing (
    end

    subgraph "⑤ Curved Surface Texture & Shadow Projection (Moho)"
        MEW["<b>Mesh & Pin Weaver 3D</b><br>Camera-Facing Bake (Shift+W / Shift+P)<br>Frontal Drawing for Curve
    end

    subgraph "⑥ Final Animation Completion"
        SS --> Final["<b>Completed Animation!</b><br>Stylized 2D hand-drawn nuance layered over rock-solid 3D st
        FW --> Final
        MEW --> Final
    end
```

Specialized Domains Handled by Each Tool

- 1. **MotionWeaver (Core Hub Engine):**
 [Translation Center / 3D Motion Hub] Accurately transforms and projects 3D bone rotation data into 2.5D canvas coordinates and layer stacking order (Z-depth). Serves as a universal motion hub with support for Poser and Blender integration.
- 2. **BatchWeaver (Multi-Angle Generator):**
 [Cinematographer / Batch Exporter] Automatically exports multiple camera perspectives (Front, Oblique, Profile, High-Angle, Low-Angle) from a single performance in a single pass, completely eliminating manual multi-angle setup.

3. **LayerWeaver (Krita Plugin):**

【Parts Factory】 Analyzes tagged layers in Krita (@ , > , *) and automatically segments artwork into animatable "parts (islands)" and "joint holes (overlap margins)" to prepare rigging assets.

4. **RigWeaver (Moho Script):**

【Assembly Line】 Automatically integrates MotionWeaver's skeletal data and LayerWeaver's part assets inside Moho, instantly creating perspective-driven animated characters.

5. **FoldWeaver 3D (Moho Script):**

【Stage & Prop Carpenter】 Select 2 points on a 2D flat vector and press `Shift+A` to fold backgrounds and props into 3D space like papercraft; use `Shift+0` to summon any corner to the world origin for direct frontal hand-drawing.

6. **Mesh & Pin Weaver 3D (Moho Script):**

【Dual Projection Engine / Surface Artisan】 Directly projects and bakes 3D motion onto camera-facing 2D canvases with 100% mathematical precision.

- **MeshWeaver (`Shift+W`):** Bakes a smart quad mesh from a 4-point guide, achieving perfect adhesion for cylinders, signboards, curved textures, occlusion masking, and cast shadows.
- **PinWeaver (`Shift+P`):** Bakes zero-length pin bones from 1 to 10 3D vertices, allowing fluttering scarves, ribbons, sashes, badges, and accessories to track smoothly like silk.

1.3 The Production Pipeline That Dramatically Streamlines Animation

At the foundation of this system is an efficient workflow born from self-taught wisdom: "Lock down the movement (bones) and stage (space) first, then refine the artwork (expression) at your own pace."

1. **Prepare the Motion "Guide" (MotionWeaver / BatchWeaver):** Import geometrically correct 3D motion (BVH) from smartphone apps (TDPT, etc.) to establish a rock-solid foundation that prevents drawing jitter.
2. **Automate Drawing Preparation (LayerWeaver):** Import 3D poses into Krita as reference underlays, automatically cutting parts and generating joint margins based on layer tagging.
3. **Generate "Moving Storyboards" via Auto-Rigging (RigWeaver):** Automatically bind bones to graphics inside Moho, producing a fully animated draft (animatic) at this stage.
4. **Intuitive Environment & Prop Modeling (FoldWeaver 3D):** Draw 2D flat layouts and fold them with `Shift+A` to assemble rooms and props.
5. **Project Curved Textures, Details & Shadows (Mesh & Pin Weaver 3D):** Bake 3D motion using `Shift+W` (quad mesh) or `Shift+P` (pin bones) to paint cel-style cast shadows and decorations as if drawing on a flat sheet of paper.
6. **Polish as Much as You Desire (Cleanup & Stylization):** Trace over locked motion guides with vector layers, infusing hand-drawn timing, pauses, eases, and exaggerated expressions to finalize your animation.

1.4 Ultimate Application: Layout-First (Keyframe-First) Pose-to-Pose Workflow

When you want to create animations with entirely custom timing and choreography without using 3D BVH mocap data, the Weaver Series enables a professional "Layout-First Pose-to-Pose Workflow":

1. **Create Pose Sets (LayerWeaver):** Draw key animation poses (such as attack stances) in Krita using "stick figures (markers connected by islands)" and save them as PSD files.

- 2. **Convert Poses into Data (MotionWeaver):** Process those PSDs through MotionWeaver to mass-produce 100% pure keyframe coordinate data (`_3d.csv`).
- 3. **Assemble Choreography & Interpolation (RigWeaver):** Arrange the `_3d.csv` files on the Moho timeline and utilize Moho's native Pose-to-Pose automatic interpolation to build character performances and animatics.
- 4. **Finalization (Cleanup):** Once timing and easing are perfected, draw final production artwork via LayerWeaver and bind them over the completed skeletal structure.

2. Naming Engineering & Tagging Logic: Naming as Blueprints

In the Weaver system, layer and folder names are far more than mere human-readable labels. They function as **direct operational commands (Visual Scripting) instructing the system how to rig and animate.**

2.1 Visual Scripting Tag Reference (Complete Symbol Dictionary)

The following is a comprehensive reference of symbol tags recognized by LayerWeaver and RigWeaver:

Symbol	Name & Syntax	Syntax Examples	Role & Operational Mechanism
@	Spatial Snap / Layer Binding <code>ChildPart @</code> <code>ParentBone</code>	<code>@Head</code> <code>Sword @ RHand</code> <code>Normal@Mouth</code>	Firmly fixes and binds a layer or group to a specified bone. The most efficient approach is grouping facial variations (blinks, mouth shapes) together and naming the parent group <code>@Head</code> .
>	Dynamic Hierarchy / Bone Chain <code>Parent > Child ></code> <code>Grandchild</code>	<code>Arm > Hand</code> <code>Hip > Scarf_Root(0)</code> <code>> Scarf_Tip(0)</code>	Constructs a dynamic skeletal parent-child hierarchy (bone chain) inside the output PSD directly through layer naming.
*	Guide Marker (Non-Cutter Declaration) <code>j[Number]*</code>	<code>j12*</code> , <code>j13*</code>	Declares a joint marker strictly as a "pure alignment guide that cuts no holes into artwork (Islands) and performs no layer merges." *Displayed in green as <code>j12*</code> in preview; accidental holes are automatically patched and repaired from the Art layer.
-	Batch ID Range <code>j[Start]-j[End]</code>	<code>j1-j5</code> , <code>i1-i3</code>	Range symbol used in LayerWeaver's UI table (Joints/Islands columns) to input and specify sequential IDs in batch.
(0)	CTA5 Pivot Strength Zero <code>BoneName(0)</code>	<code>Scarf_Root(0)</code>	Sets bone influence/strength to zero for Cartoon Animator 5 (CTA5), making it function purely as a rotational pivot (ignored by Moho).

2.2 Canvas Architecture: Separating Design-First Drawing from "Export Layout"

The fundamental principle of Weaver 2D is "**Consolidation into a Single Island**" (merging drawn parts into unified images), but this does NOT mean you must draw parts separated from the initial design phase.

- **The "Draw Overlaid, Move Aside" Workflow:**

When designing elements like skirts or capes, it is natural to draw them overlaid at the waist while evaluating character balance. Draw them with full aesthetic consideration.

However, running analysis (Step 1) in that state causes the image engine to mistake the skirt and legs for a single connected body, merging pixels and ruining leg motion.

- **Leveraging Krita's Layer Features:**

To prevent this, the Weaver methodology is: "**Once drawn, shift the finished layer to an empty area on the canvas.**" This maintains design consistency while clearly communicating to the system: "This is an independent part."

- **Automated Logical Binding:**

When imported into Moho, the parts appear in their drawn (shifted) positions, but bone binding is already complete. The moment you move the part back to the waist in Moho, it begins moving in perfect synchronization with the design.

2.3 Specialized Auto-Snapping: The Creator's Lip-Sync Shortcut

While parts are generally aligned manually, "**Face (@Head)**" and "**Hands (@RHand , @LHand)**" are **special**. This reflects the creator's dedication to eliminating the most tedious repetitive tasks:

- **Target Tags:** @Head (and nested @Mouth , etc.), @RHand , @LHand
- **Auto-Snap Behavior:** Regardless of where they are drawn on the canvas, these elements **automatically snap to their correct physical positions (waist/wrist bone locations)** upon executing RigWeaver.
- **Freedom of Layer Order (Independent Binding DNA):**

This is the true power of @ tags . Layers retain **independent binding metadata (DNA)** upon import:

- **Directorial Freedom:** If you want "BackHair" behind the head bone and "FrontHair" in front, you can freely rearrange layers inside Moho.
- **Liberation from Rigid Grouping:** Even when layer stacking order is rearranged as independent layers, each part stays synchronized with its assigned bone.
- **The Deliberate Choice Not to Fully Automate:**

To preserve this "freedom of layer stacking order," the Weaver system intentionally does not chain rigging and unitization (SmartSwitch) into an unstoppable single macro. Users can adjust layer order until fully satisfied, then pack the rig by executing SmartSwitch at their own timing.

3. Core File Specifications: The Weaver Body of Knowledge (The Lawbook)

The reason the Weaver Series can flexibly control diverse characters and 3D objects without modifying core source code lies in its consistent architecture (The Lawbook), which strictly decouples "empirical facts" from

"governing rules."

3.1 Tight Inter-File Synergy

The system completes character rigs through the tight cooperation of three core files:

1. `islands.csv` (**Physical Parts Manifest**): The empirical record of extracted physical parts (islands) and joint marker coordinates generated by LayerWeaver.
 2. `rigging_map.csv` (**Wiring Blueprint**): The structural blueprint defining "which island (part) connects to which bone, and in what sequential order."
 3. `templates.json` (**Master Catalog & Governance Rules**): Top-level governance rules dictating "which wiring blueprint (e.g., `rigging_map_human.csv`) and physical constants to use for humanoid characters."
-

3.2 `rigging_map.csv` : The Art of Wiring Bones to Artwork

This file is not a mere static list, but an advanced wiring diagram that sets "Layer Name," "Island Name," and "Bone Name Chain" together on each line.

- **Bone Assignment (Designation)**: Specifies which bone governs a specific part (island) and defines its motion tracking.
- **Hierarchical Chain Construction**: Writing multiple bone names within a single line (e.g., `Hip, Torso, chest`) automatically builds a parent-child chain from left to right.

[Important] Branching Hierarchy Construction Mechanism

While a single line in `rigging_map.csv` can only express a linear series, "**sharing identical bone names across multiple lines**" unlocks complex branching structures:

- **Example**: Including `LShoulder` in both the "Torso line" and "Left Arm line" allows the system to identify the shared intersection pivot (`LShoulder`). Like connecting puzzle pieces, the system automatically completes a massive tree branching from torso into left and right arms.

A. Force & Physics Control

- `hip_force` (**Default: 0.4**): The strength with which the hip bone influences/pulls the image (island). Lower values cause the torso to lag behind, while higher values create overly rigid movement.
- `flexi_force` (**Default: 0.1**): Application strength of flexi-binding (organic, rubber-like deformation). Prevents unnatural mesh collapsing at sharp joint bends.

B. The Bridge Between 3D and 2D (Virtual Bones Logic)

- **2D-Specific "Bone Extension (Structural Complementation)"**: In 2D drawing, the torso is often drawn as a single unified layer with only hip and chest markers. However, bending or twisting backward with insufficient intermediate joints causes mesh tearing. Through `virtual_bones` in `templates.json`, the system **automatically injects virtual control joints (e.g., Torso, Neck) without altering production artwork**, applying detailed 3D motion cleanly across a single mesh.

- `redirect_children` : Directs child bones (e.g., arms) to automatically reconnect their parent links to the newly injected virtual parents.

3.3 Core Lawbook: `templates.json` Technical Parameter Dictionary

`templates.json` constitutes the fundamental governance rulebook of the Weaver system, providing a complete configuration guide to customize behavior safely.

Parameter Name	Type / Default Example	Technical Details & Behavioral Specifications
<code>bind_all_unassigned_layers</code>	<code>bool (true)</code>	When <code>true</code> , automatically infers and binds unassigned layers (missing in CSV) to the closest bone based on coordinates. Prevents forgotten shadow layers or accessories from falling off.
<code>strict_match</code>	<code>bool (false)</code>	When <code>true</code> , skips analysis if layer and template names differ by even a single character.
<code>search_radius_factor</code>	<code>float (0.75)</code>	Multiplier for the marker search radius relative to an Island. Can be fine-tuned to fit different character body proportions.
<code>flexi_binds</code>	<code>array</code>	List defining which bones throughout the body are permitted to undergo non-rigid, flexible physical deformation.
<code>face_bone_keywords</code>	<code>array</code> <code>(["hair", "eye", "mouth"])</code>	Keywords excluded from full-body physical deformation (flexi-binding). Parts matching these keywords retain crisp, rigid shapes.
<code>variation_mouth_maps</code>	<code>object</code>	Defines mouth variation phoneme names and import sequences. The core dictionary for automated lip-syncing.
<code>joint_maps</code>	<code>object</code>	"Bone Name Translation Dictionary" mapping 3D motion (BVH) bone naming conventions to Weaver internal logical bone names.
<code>redirect_children</code>	<code>object</code>	Settings that automatically remap child bone parent connections to newly injected virtual joints when <code>virtual_bones</code> are used.
<code>is_accessory</code>	<code>bool (true)</code>	Flag that automatically detects parts using marker names as triggers, allowing different costume variations (e.g., skirts) to share the same template.

Parameter Name	Type / Default Example	Technical Details & Behavioral Specifications
<code>rig_perspective_base_z</code>	<code>float (4.0)</code>	【For RigWeaver】 Reference value to calculate perspective depth from Moho layer scale (smaller values exaggerate perspective).
<code>motion_perspective_default_z</code>	<code>float (8.0)</code>	【For MotionWeaver】 Default perspective depth value used when perspective intensity is unspecified during motion conversion.

3.4 File Placement Hierarchy: Project-Specific vs. System Standard

- **Project-Specific Files (`templates.json` , `islands.csv` , etc.):**
Place these in the same folder as your PSD file or Moho project. Weaver always prioritizes loading from this directory first.
- **System-Standard Resources:**
Located in `[Moho Install Dir]/Scripts/ScriptResources/rigweaver/` . Common default templates are consolidated here.

4. MotionWeaver & BatchWeaver: Performance Translation & Multi-Angle Batch Generation

This pipeline captures 3D motion as "performance data from your dedicated virtual actor" and mass-produces the camera angles required for anime production in batch.

4.1 `_3d.csv` : The "Performance Record" by Your Dedicated Actor

The generated `_3d.csv` is keyframe data that translates raw 3D motion (BVH, etc.) into 2.5D Moho space with high mathematical precision.

- **Focusing Entirely on Artwork:** By offloading complex numerical spatial calculations to the system, artists can dedicate 100% of their creative energy to drawing compelling artwork.

4.2 BatchWeaver: Automated Multi-Camera Angle Extraction

Automatically exports multiple directorial camera angles (Front, Profile, Back, High-Angle, Low-Angle) from a single motion performance in a single pass.

- **Dramatic Production Cost Reduction (Scene-Driven Rigging):** By locking camera angles in advance, artists only need to draw the precise assets required for each specific shot, completely avoiding the pitfall of attempting to build an overly complex "all-purpose universal rig."

4.3 Pose-to-Pose (Still-Image Markers & Single-Frame CSV)

- **Instant Extraction of Key Poses:** Placing a zero-duration marker on the Moho timeline automatically exports a reference PNG image along with a single-frame 3D CSV file (`_f[frame]_s.csv`).
- **Directing Animation via Automatic Interpolation:** Arranging key poses across the timeline and interpolating them with Moho's robust bone interpolation enables ultra-fast construction of custom animation timings, holds, and eases.
- **【Mode 2 Exclusive: Automatic T-Pose Marker Insertion】** : Selecting Mode 2 in `BatchGen` automatically injects a T-Pose marker at Frame 0 and exports an integrated underlay PNG (`T-Pose_f0.png`) ready for drawing in Krita.

4.4 Fully Portable (Self-Contained) Package Export

Selecting "Create New Folder" in `BatchGen` automatically copies the source BVH file and all governing configuration files (`templates.json` , etc.) into the destination folder. The exported package remains completely self-contained with no broken paths, allowing immediate execution across different environments.

5. FoldWeaver 3D: Papercraft Geometry for Environments & Props

FoldWeaver 3D is an intuitive papercraft modeling tool designed to assemble background environments and props—such as room walls, desks, chairs, treasure chests, and building facades—directly within Moho's 3D space.

5.1 Intuitive Paper-Folding Geometry (`Shift+A`)

- **Liberation from Complex 3D Math:** No need for manual vertex coordinate entry or complex transformation matrix calculations common in traditional 3D software. Simply draw flat 2D layouts using Moho's vector tools, select 2 vertices along the edge you want to fold, and press `Shift+A` .
- **Automatic Hinge Construction:** The tool instantly calculates the angle and rotational hinge axis between the 2 selected points, immediately raising the 3D surface folded at your specified angle (e.g., 90°, 45°).

5.2 Posture Inheritance & Cel-Style Cast Shadow Drawing

- **100% 3D Posture Inheritance:** By duplicating a folded and assembled surface (such as a wall or chest lid) and redrawing the vector paths inside, you can draw window frames, grooves, or panel lines that flawlessly inherit that surface's tilt, orientation, and perspective.
- **Intentional Hand-Drawn Cast Shadows:** Rather than relying on hyper-realistic CG lighting engines, artists can paint hand-drawn vector cast shadows wherever they look most aesthetically pleasing in traditional 2D animation.

5.3 Universal Centroid Snapping (`Shift+0`): Origin Summoning & Frontal Drawing

- **Instantly Summon Any Target Corner:** When you want to paint an emblem on a specific tilted corner or a keyhole on a rotated box face, simply select the target vertex and press `Shift+0` .
- **Frontal Drawing at World Origin (`0,0,0`) :** The tilted corner is instantly snapped to the world origin directly facing the camera. Rotate the camera, create a new vector layer, and draw details with the familiar ease of standard frontal 2D drawing—guaranteeing zero drawing distortion even on the most extreme angles.

6. Mesh & Pin Weaver 3D: Dual Projection Engines Weaving 3D Anchors & 2D Silk

Mesh & Pin Weaver 3D is a revolutionary rigging engine that directly projects and bakes the motion of 3D models onto a camera-facing (2D screen plane) drawing canvas with 100% mathematical precision. By calculating Moho's complete composite transformation matrix, drift and jitter common in traditional image-based motion tracking are fundamentally eliminated.

6.1 Roles & Separation of the Two Projection Engines

Provides two specialized engines tailored to your specific asset structure and rigging goals:

Engine	Execution Command	Output Rig Structure	Best Use Cases
MeshWeaver 3D	<code>we_meshweaver_tool</code> (<code>Shift+W</code>)	<code>Auto_Warp_Rig</code> └ <code>Mesh_Warp</code> (4-point Smart Mesh) └ <code>Paint_Canvas</code> (Auto-bound)	Curved Surface Textures & Surface Occlusion: Cylinders, signboards, pipes, robot armor plates, eliminating Moho 3D artifact lines.
PinWeaver 3D	<code>we_pinweaver_tool</code> (<code>Shift+P</code>)	<code>Auto_Pin_Rig</code> └ <code>Pin_1 .. Pin_N</code> (Pin Bones) └ <code>Paint_Canvas</code> (Flexi-bound)	Fluttering, Accessories & Organic Tracking: Scarves, sashes, badges, buttons, joint tracking, multi-point pin deformation.

6.2 PinWeaver 3D Core Engineering: "Zero Length" & "Wide-Area Influence"

What fundamentally differentiates PinWeaver 3D from traditional rigging tools is the unique geometric property of its generated bones:

1. **Zero-Length Pin Bones (`fLength = 0` / `fIsPin = true`):**

- Unlike conventional bones with rotational axes, they operate purely as "single 2D coordinate anchors" driving across the screen plane.
2. **Flexi-Binding via Wide Influence Fields (`fStrength = 6.0`):**
- By granting wide influence fields, artwork drawn on `Paint_Canvas` between multiple pins receives **smooth, organic weight interpolation resembling silk or rubber**, responding dynamically to each pin's movement.
3. **"Button Script" Format Preserving Drawing Rhythm:**
- No need to switch drawing tools; pressing `Shift+P` completes rig generation, keyframe baking, and switching to the drawing canvas automatically in ~1 second.
-

6.3 Three Extreme-Power Use Cases

1. **Fluttering Ribbons, Scarves & Sashes (Multi-Point Pin Deformation / 3–5 Points):**
 - Select 3 to 5 guide vertices along a 3D character's back or head and press `Shift+P`. Drawing a ribbon connecting each pin creates a scarf fluttering organically against wind resistance.
 2. **Snapping Accessories, Badges & Shoulder Armor (1–2 Points):**
 - Baking 1 or 2 points at the chest emblem or shoulder armor position locks parts to extreme rotations without tedious 3D binding setups.
 3. **Curved Surface Tracking via Hierarchy:**
 - Parent intermediate control bones (children) to 3D guide pin bones (parents) to smoothly control the bending mid-curves of flexible arms or legs.
-

6.4 Non-Destructive Iteration (Re-Baking) & Fail-Fast Architecture

- **Complete Protection of Drawn Artwork (`Paint_Canvas`):**
 - After revising 3D motion or camera work, simply press `Shift+W` or `Shift+P` again. The script updates keyframe coordinates with the latest data without deleting any drawn linework or fills.
 - **Fail-Fast Architecture (Safety Limits):**
 - Features strict internal safety checks—MeshWeaver enforces "Exactly 4 Points (`EXACTLY_4_POINTS`)" while PinWeaver enforces "Maximum 10 Points (`MAX_POINTS = 10`)" —aborting immediately upon invalid conditions to prevent project corruption or performance drops.
-

6.5 [Special Feature & Practical Technique] Eliminating Moho 3D Layer "Artifact Line Glitches" via the Weaver Trio (Fold / Mesh / Pin)

To resolve the long-standing issue in Moho Pro's 3D Extrusion where **"the original center vector line remains embedded inside, exposing unwanted intersecting artifact lines and see-through glitches on top and side surfaces,"** the Weaver trio works in unison to elevate models into pure hand-drawn cel looks.

■ **Core Philosophy: Use 3D as a "Kinematic Anchor" Rather than the "Final Render"**

Instead of rendering the raw 3D extruded mesh, shift your mindset: **"Use 3D strictly as an anchor for accurate perspective and rotation, while painting over and covering all visible surfaces and lines on Weaver's 2D canvas."**

【Artifact Line Elimination & Pure Cel-Look Trio Workflow】

① FoldWeaver : Organize 3D structure, origin snapping (Shift+O), and markers.

② MeshWeaver : Cover quad surfaces (Shift+W) with 4-point mesh, 100% occluding internal artifact lines.

③ PinWeaver : Snap patch shapes or hand-drawn details to corner glitches (Shift+P).

→ 【Complete】 Unwanted artifact lines vanish 100%, yielding a pristine hand-drawn cel-look prop!

Tool	Role in Artifact Elimination & Cel-Look Transformation	Concrete Example
FoldWeaver	3D Foundation & Reference Setup Hinge connections, marker setup, origin alignment (Shift+O)	Clean 3D folding from 2D layouts, organizing spatial relationships of intersecting faces in advance.
MeshWeaver (Shift+W)	Complete Quad Surface Occlusion & Elimination Covers entire surfaces with 4-point mesh, 100% hiding internal artifact lines	Book covers/spines, box faces, signboards, smartphone screens (flawless solid fills and clean outer contour lines).
PinWeaver (Shift+P)	Corner/Point Patching & Detail Accents Bakes arbitrary pin bones to adhere localized patches and decorations	Masking awkward T-junction seams at corners, adding clasps, emblems, and hand-drawn styling lines.

7. RigWeaver & SmartSwitch: Operating Animated Assets & Rig Promotion

While Weaver establishes the logical skeletal structure, infusing rich emotions and stylized expression is the artist's craft. This chapter explains the mechanisms of rig promotion and asset operation provided by RigWeaver and SmartSwitch.

7.1 Automated Mechanism: Inheriting Binding DNA & Unitization

Parts auto-snapped by RigWeaver (such as @Head , @Hand , @Skirt) are elevated from simple static image layers into "intelligent units" through the Weaver_SmartSwitch script:

- **Separating Switches (Exclusive Display) and Groups (Simultaneous Display):**
 - **Facial Expressions & Hand Poses:** Grouped into **Switch Layers** where only one variant is displayed at a time, enabling instantaneous switching on the timeline.
 - **Fluttering Skirts & Ruffles:** Multiple segmented parts are consolidated into a **Group Layer** where all components are displayed simultaneously.

- **The Core: "Inheritance" & "Promotion" of Binding DNA:**

Normally in Moho, simply grouping existing layers strips their individual bone bindings, requiring tedious manual re-binding of the parent group. SmartSwitch **automatically transfers and promotes the original Island binding metadata from the @tagged layers directly onto the newly generated group or switch layer itself.**

- **Benefits to Artists: Vector Inking & Line Detail Cleanup:**

Because the parent group is securely bound to the bone, any subsequent vector cleanup or line accentuation added inside the group **flawlessly follows joint movement across all frames.**

7.2 "Instant Part Creation" from Bone Selection

SmartSwitch supports execution not only when layers are selected, but also when **bones (especially PinBones like _Scale) are selected:**

- **Behavior:** Executing the script with only a bone selected automatically generates an empty vector layer nested inside a parent group, completing bone binding instantly.
 - **Benefits to Artists:** When adding new elements (e.g., chest patches, armbands) to an existing rig, simply select the bone and run SmartSwitch—a dedicated, perfectly aligned drawing canvas is instantly prepared at the correct hierarchy.
-

7.3 Fine-Tuning Forces to Control Material "Feel"

Adjusting Force parameters in `templates.json` directly reflects clothing rigidity and bodily flexibility:

- **Heavy / Thick Costumes:** Slightly increase `hip_force` and lower `flexi_force` to create the feel of heavy fabric clinging firmly to body movement.
 - **Flexible / Creature Characters:** Increase `flexi_force` to create organic, undulating movement where flesh ripples beyond the underlying bone motion.
-

7.4 The Ultimate Internal Failsafe Hard-Coded by the Developer

To guarantee technical stability in production, a **hard-coded safety mechanism** is embedded within `RigWeaver.lua` that executes regardless of external configuration files:

- **Forced Shoulder Re-Anchoring Logic:** Inside `ExecuteCreateBones2D`, shoulder bones (`LShoulder`, `RShoulder`, etc.) are **forcibly reconnected as children of the Torso**, regardless of raw 3D data or CSV structure.
 - **Purpose:** Prevents 2D rig shoulders from detaching from the torso even when importing extreme or distorted 3D motions, guaranteeing rock-solid mesh deformation at all times.
-

7.5 Choosing Between 2D Mode and 3D Mode

- **3D Mode (Recommended / Standard):** Run with the "2D Mode" checkbox unchecked. Captures 3D rotation and perspective scaling directly onto the 2D rig, achieving dynamic spatial action.
- **2D Mode (Specialized Use Cases):** Check this option only when you wish to animate artwork flatly without altering 2D silhouette proportions (locks X/Y rotational axes to 0°).

8. The 6 Unified Keybindings: Complete Reference

The Weaver Series (FoldWeaver / Mesh & Pin Weaver) incorporates 6 unified left-hand-centric keybindings, allowing creators to proceed with production at lightning speed without interrupting creative thought with tool switching:

Keybinding	Function Name	Target Tools	Behavioral Specification & Production Role
Shift+M	Marker (Place Marker)	Fold / Mesh / Pin	Instantly places alignment point markers at selected vertices or coordinates.
Shift+A	Align / Fold (Fold Geometry)	FoldWeaver 3D	Automatically calculates the hinge axis between 2 selected points, folding flat 2D vectors into 3D space at specified angles (90°, 45°, etc.).
Shift+C	Clear (Clear Guides)	Fold / Mesh / Pin	Clears all auxiliary markers and guide geometries in batch, leaving the viewport clean.
Shift+O	Origin Snap (Universal Centroid Snap)	Fold / Mesh / Pin	Instantly summons the centroid of selected corners or parts to the world origin (0,0,0) directly facing the camera (Frontal Drawing Mode).
Shift+W	Warp Bake (Quad Mesh Bake)	MeshWeaver 3D	Calculates 3D rotation and perspective distortion across all frames from a 4-point guide (Guide_Quad), auto-baking a camera-facing 2D Smart Warp rig (Auto_Warp_Rig).
Shift+P	Pin Bake (Pin Bone Bake)	PinWeaver 3D	Auto-bakes a zero-length pin bone rig (Auto_Pin_Rig) from 1 to 10 3D vertices, allowing fluttering scarves, badges, and organic deformation to follow.

9. Troubleshooting & Creator's Wisdom (FAQ)

Bottlenecks encountered by the creator during countless hours of self-taught experimentation, along with their definitive solutions.

Q1. The image does not follow the bone, or stretches in strange directions.

- **Cause:** Are markers placed too close to the outer border of the "Island (part)"?

- **Solution:** When LayerWeaver analyzes artwork, markers placed outside the `search_radius_factor` range become lost. Ensure markers are firmly placed inside the "meat" of the intended part.
-

Q2. I added a guide marker (*), but a hole was still punched into my part.

- **Cause:** The asterisk `*` may not be placed correctly at the end of the marker name (e.g., using full-width characters), or the marker fell outside threshold detection.
 - **Solution:** Name the marker with a standard ASCII asterisk, such as `j12*`. LayerWeaver recognizes `*` tagged markers as "non-cutters (pure guides)," punching zero holes and automatically patching the area using pixels from the Art layer.
-

Q3. How do I create quadrupeds (animals) or special non-humanoid characters?

- **Answer:** Simply add a new section (e.g., `quadruped`) to `templates.json` and define its `target_hierarchy`. The system will immediately understand the new anatomy.
 - **Creator's Wisdom:** The fastest approach is copying the existing `human_v2` preset and swapping bone names. Using `virtual_bones`, even the most asymmetric creatures can be operated inside Moho as "clean, straightforward bones."
-

Q4. I am uncertain about the mathematical accuracy of my motion data.

- **Answer:** Launch `csv_viewer.py` without hesitation. You will see the "raw 3D motion conversion result" before passing data to Moho. If the skeleton walks correctly in the viewer, you can instantly conclude that any issue lies within Moho rigging settings (such as CSV mistakes).
 - **【Future Roadmap】**: An advanced "**Diagnostic Tool**" is planned for `csv_viewer.py`. It will cross-check PSD structure against `rigging_map.csv` definitions and alert users to "impossible wiring" in advance (currently in development). Once complete, it will transform self-taught guesswork into definitive, confident production.
-

Q5. How do I add hand-drawn details or decorations after folding surfaces in FoldWeaver 3D?

- **Answer:** Use the Universal Centroid Snap (`Shift+O`) without hesitation.
 - **Procedure:** Select the corner vertex you want to decorate and press `Shift+O` —that corner is instantly summoned to the world origin `(0,0,0)` directly facing the camera. Rotate the camera, create a new vector layer, and paint marks or details with the familiar ease of standard frontal 2D drawing.
-

Q6. How do I apply curved surface textures or fluttering accessories in Mesh & Pin Weaver 3D?

- **Answer:** Place a 4-point guide (`Guide_Quad`) and press `Shift+W` (Mesh) or select points and press `Shift+P` (Pin).

- **Procedure:** Place a 4-point guide on the curved surface (align with `Shift+O` and enter radius into Z position). Pressing `Shift+W` bakes 3D rotation and distortion across all frames into a camera-facing 2D Smart Warp rig. Simply paint cel-style cast shadows or textures on the auto-generated frontal canvas to achieve flawless curved adhesion.
-

Q7. Can I safely place and bake multiple motion CSV files sequentially along the timeline?

- **Answer:** "Yes, this is fully supported. Sequential baking across multiple spliced motions can be performed with rock-solid stability."
 - **Reason:** The latest Step 4B. Bake & Bind (`ExecuteBakeMotion2D`) incorporates a "Multi-Action Baker" that automatically scans all CSV paths and start frames (`path|start`) stored within timeline markers. This applies individual CSV scale data and durations per action block; even motions starting mid-timeline receive correct perspective, scaling, and rotation without skeletal collapse.
 - **Creator's Advice:** You can chain different dances or fight actions sequentially along the timeline, place markers, and bake them into clean 2D animations in one go (staging with pose interpolation is equally simple). However, base rig structure design (Steps 1A–2B) and initial artwork alignment (Rest Pose) strictly adhere to "Frame 0 (Setup Mode)" as the absolute baseline.
-

Q8. The perspective foreshortening feels unnaturally extreme or too weak.

- **Cause:** Depth variation intensity in the motion data (TDPT, etc.) may not match the template baseline value.
 - **Solution:** Adjust `rig_perspective_base_z` in `templates.json`.
 - **For Stronger / More Dynamic Perspective:** Decrease the value (e.g., `4.0`).
 - **For Weaker / Near-Orthographic Perspective:** Increase the value (e.g., `12.0`).
 - **Creator's Wisdom:** If entering `100` makes little visible difference, perspective has weakened to the point where variations become imperceptible. Fine-tuning within the `4.0` to `10.0` range represents the optimal "sweet spot" to feel the impact.
-

Q9. What does the warning displayed after executing Step 4A mean?

- **Answer:** This is a safety warning alerting you that right after Step 4A (Bone Generation), image layers are not yet in a state where they can deform properly (requiring Bake & Bind).
 - **Solution:** Proceed immediately to **Step 4B "Bake & Bind"**. This bakes motion to the rig and binds image layers with proper weights.
 - **If Aborting:** If you must pause without running Step 4B, open Moho's top menu and run `Menu > Bone > Reset All Bone Rigging` to reset the rig. Failing to do so leaves the mesh in a state where manually moving bones causes unnatural distortion (or no movement at all).
-

Q10. Can I execute Batch Generation (BatchGen) from an action inserted mid-timeline (e.g., Frame 48)?

- **Answer:** "Batch Generation (3B. BatchGen) must strictly be executed from Frame 0/1 origin (Timeline Sync State)."
 - **Reason:** BatchGen references Moho timeline marker frame positions to generate the batch instruction manifest (MotionWeaverData.csv) extracted from the "raw BVH file." If Moho timeline frames and raw BVH frame numbers do not match 1:1, extracted ranges will desynchronize. For this reason, BatchGen displays a "Timeline Sync Warning" dialog.
 - **Creator's Advice:** However, once motion is registered to timeline markers, the subsequent "Bake & Bind (4B)" process executes accurately from any frame mid-timeline. Remember the distinct roles: "Action Extraction (BatchGen)" requires synchronized frames, while "Motion Baking (Bake)" can be placed freely anywhere on the timeline.
-

Q11. My 2D character artwork tilts awkwardly or the Hip bone points straight downward after loading motion.

- **Answer:** In older versions, directly applying raw 3D X/Y rotations caused 2D artwork to tilt obliquely or the Hip bone to flip violently at gimbal lock singularities. In current versions, projection angle extraction (matrix transformation) and dual-unwrapping algorithms automatically resolve unintended flips and jitter.
- **Solution: Running RigWeaver with the "2D Mode" checkbox OFF (Unchecked = 3D Mode) is now the strongly recommended standard workflow.**
 - **When "2D Mode" is OFF (3D Mode):** Captures rich 3D rotational data (X/Y/Z) properly, expressing dynamic 3D foreshortening, scaling, and organic twists cleanly on the 2D rig.
 - **When "2D Mode" is ON:** Use only for specialized flat-cutout animation where you want to completely eliminate 3D perspective distortion (locks X/Y 3D rotation to 0° and applies 2D projection fallback angles to the spine).

This reference guide will continue to evolve alongside your creative journey. Once you master the logic, feel free to "lie" artistically with confidence, weaving your own unique stories.

10. Detailed Technical Specifications Directory

For in-depth architectural specifications and underlying logic, please refer to the dedicated technical specifications for each tool:

- **00_Weaver Series: [MUST READ] The Book of Beginnings**
~ The Compass of Production: Stories & Foundational Flows of all 6 Major Tools ~
- **12_LayerWeaver: Professional Technical Specifications**
~ Island Analysis, Threshold Processing, Naming Engineering & Deep Guide Marker (*) Logic ~
- **22_MotionWeaver: Professional Technical Specifications**
~ 3D Coordinate Projection, Noise Filtering & Mapping Logic ~
- **42_RigWeaver: Professional Technical Specifications**
~ 5-Stage Pipeline, Auto-Binding & Hierarchy Optimization ~