

Weaver Series: Comprehensive Reference Guide

Last Updated: 2026/04/20

1. Weaver Series: "Design Concept" and "Philosophy"

1.1 Why this tool was born: "Design Philosophy" and "Trial and Error" born from self-study

The Weaver series is the answer reached after countless trials and errors by a single creator (the author) who, without any formal education in animation, had an earnest wish to "create high-quality animation alone in a short period of time."

I am neither a professional animator nor a professional illustrator. Having no experience learning the basics at a vocational school, the biggest wall I faced when I tried to make an animation by myself was the wall of serious technical constraints: "How can I add movement beautifully?" and "How can I prevent drawing collapse?"

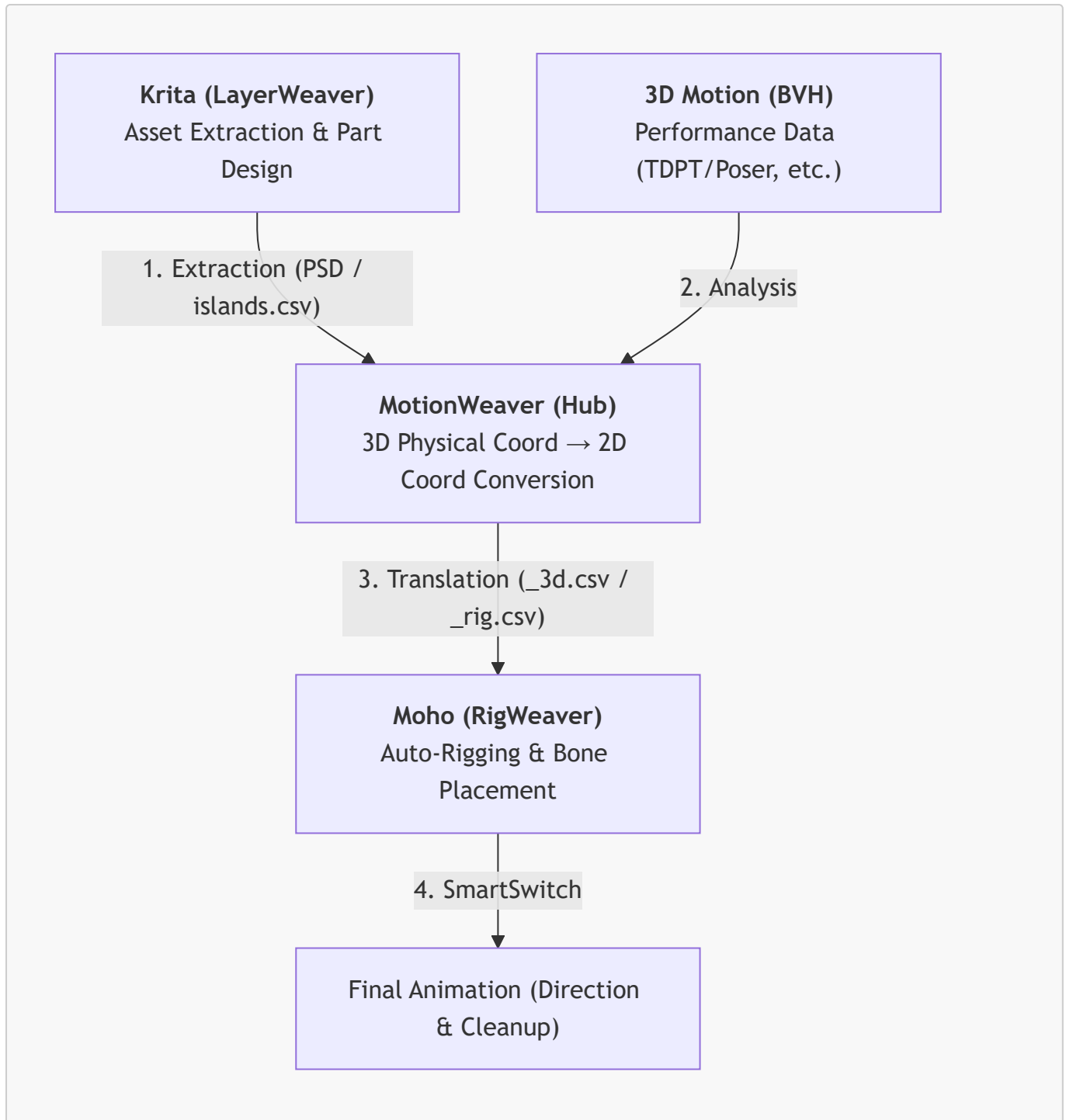
So, I thought about how I could "weave" together the wonderful existing tools I had at hand— **Moho**, **TDPT**, **CTA5**, **Poser**, **Blender** —in my solitary work, and whether I could create a system where I could pour my passion for creation only into creative expression more easily and enjoyably. That's how this Weaver series was born.

[!NOTE] **About the Origin of the G3 Template Standard** The "G3 Template Standard" supported by this system is an extremely sophisticated existing standard designed by **Reallusion**, the developer of Cartoon Animator 5 (CTA5). The Weaver series did not invent this magical template standard from scratch; rather, it functions as a "translator" to correctly guide and weave (Weave) your drawings into this "wonderful existing standard."

If you are "wanting to create alone but lost, blocked by technical walls," this tool is your ally. Weaver is a bridge built between the "geometrical consistency" of 3D and the "deformed expression" of 2D.

1.2 Data Baton Relay: Philosophy of a Specialized Studio

Weaver is not a single piece of software, but is designed as a "specialized studio" where multiple specialized tools pass the baton. Each tool concentrates on what it is "good at," and the system bridges the gaps.



"Specialized Domains" of Each Software

- **LayerWeaver (Krita Plugin): [Parts Factory]** The starting point of production. It separates illustrations not as mere images, but into movable "bodies (islands)" and "joint margins (seam allowances)" as connection points. The folder structure here is the "Design Plan (Lawbook)" that dictates the entire process until completion.
- **MotionWeaver (Core Engine): [Translation Center / 3D Hub]** The heart of the system. It precisely converts and projects 3D bone rotation information onto 2.5D canvas coordinates and depth (Z-value) that determines the overlapping order. It functions not just for Moho, but as a "Universal Motion Hub" with applications for Poser and Blender in mind.
- **RigWeaver (Moho Script): [Assembly Line]** It materializes the "logical numbers" calculated by MotionWeaver as visual bones on Moho. It automatically "wires (binds)" images to bones, setting the stage so that the drawings you created start moving in synchronization immediately.

1.3 "4 Steps" to Make Animation Dramatically Easier

At the root of this system is a proposal for an efficient workflow born from the wisdom of a self-taught creator: "Solidify the movement (bones) first, then carefully finish the meat (drawings) later."

1. **Prepare a "Guide" for Movement (MotionWeaver):** Read "3D-accurate movement (BVH)" from smartphone apps (TDPT), etc., to create a solid foundation that prevents drawing inconsistency. You don't need to draw difficult movements by hand from scratch.
2. **Automate Drawing "Preparation" (LayerWeaver):** Read 3D poses into Krita as underlays, and automatically perform the time-consuming "part separation" and "naming" according to a unique marker placement.
3. **Create a "Moving Storyboard" with Auto-Rigging (RigWeaver):** Automatically link bones and images on Moho, completing a "moving draft (video storyboard)" at this stage. The biggest strength is being able to judge the quality of the direction here. ***Note: Skipping Step 1B "Bind Character" will cause the character to scatter in subsequent steps (2B onwards). Always follow the sequence strictly.***
4. **Finish Carefully as Much as You Like (Cleanup):** Clean up with vector layers as if tracing over the movement-fixed guide. Since the parent group already holds the keys to the movement, the drawings start moving in perfect synchronization with the system the moment you redraw them.

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

If you want to create animations with completely original timing and acting without using 3D BVH materials, the Weaver series enables a "Layout (Key Pose) First Professional Workflow".

1. **Create Pose Collection (LayerWeaver):** Instead of official drawings, use "stick figures (markers connected by islands)" in Krita to draw multiple key poses (e.g., attack stances) and save them as PSDs.
2. **Digitize Poses (MotionWeaver):** Pass those PSDs through MotionWeaver to mass-produce 100% pure key pose coordinate data (`_3d.csv`).
3. **Build Acting and Interpolate (RigWeaver):** Line up the `_3d.csv` files on the Moho timeline and utilize Moho's standard "Pose-to-Pose (Auto-Interpolation)" feature to complete the character's acting and timing (animatic).
4. **Finalize (Cleanup):** Once the animation length and timing (ease-in/out) are perfectly set, draw the final official illustrations in LayerWeaver, place them over the completed skeleton, and bind them.

By using this method, you can build the highest quality animations in the shortest time without the hassle of searching for 3D materials or the rework of drawing wasted pictures.

2. The Blueprint Named "Naming" (Naming Engineering)

In the Weaver system, layer names and folder names are not mere labels. They are **direct configuration information (logic) for the system on "how to rig and how to move."**

2.1 Canvas Design: Separating Design-First Drawing and "Export Placement"

The basic principle of Weaver 2D is to integrate drawn parts into a single image ("**Aggregation into a Single Island**"), but this does not mean "you must draw parts separately from the drawing stage."

- **"Draw Overlapping, Move to Place" Workflow:** For example, when designing a skirt or a cape, it is natural to draw it overlapping the waist. Please draw it attractively as part of the character while looking at the overall balance. However, if you execute the analysis (Step 1) in that state, the image analysis engine will misidentify the "skirt" and "leg" as the same single body, integrating the pixels and hiding the movement of the leg or causing the shape to collapse.
- **Utilizing Krita's Layer Functions:** The Weaver way to prevent this is to **"move the finished layer to a completely empty space on the canvas."** This allows you to maintain design consistency while correctly telling the system, "This is an independent separate part." The trick is to function the canvas not as a "place to put the final prediction map," but as a "parts shipping area" where separate parts are eventually lined up.
- **Automation of Logical Binding:** When imported into Moho, the parts appear in the positions they were drawn (shifted state), but don't worry. Binding with bones is completed automatically, so the moment you return the part to the waist position on Moho, that part starts moving in synchronization with the system at the position as designed.

2.2 Special Snap (Auto-Adsorption): The Author's Lip-Sync Reduction Technique

Basically, parts are aligned manually, but **"Face" and "Hand" are special**. The author's wish to "finish the most tedious work in an instant" is contained here.

- **Target Tags:** @Head (and nested @Mouth, etc.), @RHand, @LHand
- **Auto-Snap Function:** No matter where they are drawn on the canvas, when RigWeaver is executed, they are **automatically placed (snapped) to the "place where they should originally be (waist or wrist bone positions)" on the character body**.
- **Development Intent:** Lip-syncing (mouth animation) and hand replacement are the most frequent tasks in animation production. By auto-adsorbing these, creators are freed from monotonous "positioning" and can concentrate more on creative "facial performance."
- **Avoiding Overlap (Moving Layers):** For example, if you draw a skirt overlapping the waist position, it will merge with the "leg" during image analysis, and the movement of the leg will be invisible. Therefore, the correct Weaver way is to **move accessory or variation part layers to an "empty space" away from the body** after the design is solidified, and then perform the analysis.
- **Automation of Logical Binding:** When these are imported, the parts are placed in their positions on the canvas (scattered state), but RigWeaver has completed all binding calculations with the bone structure behind the scenes.
- **Auto-Snap Function:** No matter where they are drawn on the canvas, when RigWeaver is executed, they **automatically move (snap) to the "place where they should originally be (waist or wrist bone positions)" on the character body**.
- **Freedom of Layer Order (Independent Binding):** This is the true value of @Tags. These layers **retain individual binding information (configuration information with bones: DNA) at the time of import**.
 - **Range of Expression:** For example, if you want to place "BackHair" behind the head bone and "FrontHair" in front, you can freely move these layers on Moho.

- **Freedom from the Shackles of Grouping:** Normally, there is no need to forcefully group them to maintain binding. Even if you freely swap the overlapping order as independent layers, each part continues to synchronize correctly with the bone movement.
- **The "Will" Not to Automate Everything:** To protect this "freedom of overlapping order," the Weaver system does not make rigging and unitization (SmartSwitch) a single continuous automated process. Adjust the front-to-back relationship of layers with the user's hand until satisfied, and then execute **SmartSwitch** to pack at your own timing. Because of this "selectable interval," Weaver can stay close to the craftsman as a tool.

B. Other Parts (Manual Assembly)

Skirts (**Weary_Acc(0)**, etc.) are not auto-snapped. These are manually placed by the user in the correct position (such as the waist) on Moho. The moment the position is aligned, they start moving due to the binding that was automatically completed.

2.3 Scalability and Manual "Weaving": Registration Process to the Analysis Table

The reason users don't have to edit **templates.json** every time is thanks to the **"Discovery Phase"** the system performs at the start of rigging, but there is an **important step here to let the user's "will" intervene.**

- **Auto-"Detection" of Unknown Parts:** If layer names in the PSD contain naming tags such as **@** or **>** (e.g., **Sword>Hand**), the system immediately displays them as "new units not described in the CSV or template" in the preview. In the Viewer, "Island IDs (i)" and "Margin IDs (j)" corresponding to them are automatically assigned and can be confirmed.
- **Manual Registration to the Analysis Table (CSV) (Essential Step): This is the important point.** The system "detects" them, but does not complete the rigging on its own.
 1. **Confirmation in Viewer:** First, check the "Island ID (i)" and "Margin ID (j)" of the part on the preview screen.
 2. **Add Row:** Next, open the analysis table (**islands.csv** or **rigging_map.csv**) yourself and add a new row.
 3. **Manual Entry of Information:** Enter the "Part Name" attached to the marker, the confirmed "Island ID (i)", and the "Margin ID (j)".
- **Final Control for Craftsmen:** Because of this effort of "manual entry," it is possible to prevent unintended garbage layers from being rigged, and craftsmen can "weave (Weave)" parts exactly as they intend. In the margin number column, range specification such as **j23-40** is also supported, allowing one part to have the influence of multiple bones (such as bending) as desired.

3. Core File Specifications: Weaver's Knowledge System (The Lawbook)

The Weaver series can move diverse characters without individual program modifications because it has a consistent configuration system (Lawbook) behind it that separates "facts" and "rules."

3.1 Close File Integration

The system completes a single character by linking the following three files:

1. **islands.csv (Parts List of the Body)**: A "factual record" of physical parts (islands) extracted by LayerWeaver and marker placement information.
2. **rigging_map.csv (Wiring Diagram)**: A "blueprint" describing "which island (part) to connect to which bone, and in what order."
3. **templates.json (Master Catalog)**: Top-level "governance rules" such as "if it's a humanoid type, use this wiring diagram (**rigging_map_human.csv**)."

3.2 rigging_map.csv: The Secret of "Wiring" Bones and Body

This file is not a mere list, but a sophisticated wiring diagram that describes "layer name," "island name," and "bone name chain" as a set in one row.

- **Which Bone to Move With (Assignment)**: Specify the bone to be assigned to one part (island) and define the tracking to its movement.
- **How to Connect Which Bones (Chain Construction)**: By writing multiple bones in one row (e.g., **Hip, Torso, Chest**), the system automatically builds a parent-child relationship (chain) from left to right.

[IMPORTANT] How Branching is Built

Only a serial chain can be written in one row of **rigging_map.csv**, but by "**sharing the same bone name in multiple rows**," complex branching is realized.

- **Example**: Include the bone name **LShoulder** in both the "Torso row" and the "Left Arm row." Then the system finds a common connection point (Pivot) where "there is an LShoulder in the torso part" and "the starting point of the left arm is also LShoulder," and automatically completes a huge tree structure branching from the torso to the left and right arms, like connecting puzzle pieces.

A. Control of Strength and Physics (Forces)

Parameters born while the author pursued "anime-like, supple movement."

- **hip_force (Default: 0.4)**: The strength with which the hip bone pulls the image (island). If this is low, the torso will be left behind by the bones during intense movement; if high, it will be stiff movement. The golden ratio of "most human-like and not collapsing" reached after dozens of tests by the author is set.
- **flexi_force (Default: 0.1)**: The intensity of flexi-binding (rubber-like deformation). It prevents meat from unnaturally collapsing at the bend of a joint and produces organic softness.

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

One of the most intelligent mechanisms of Weaver is the auto-generation of "fictional bones," but the true reason for its existence is not a lack on the BVH (3D) side, but "**physical constraints on the PSD (drawing) side**."

- **Challenges Because PSD is the "Correct Answer"**: Weaver builds rigs using PSD islands and marker positions as the absolute correct answer. However, in 2D illustrations, the "Torso" is usually drawn as one large mass. If it's too faithful to the PSD marker configuration (e.g., only waist and chest), only one long bone will be assigned to the torso.

- **"Bone Expansion (Structural Interpolation)" Unique to 2D:** In this state, when bending the body, there will be a physical shortage of "intermediate joints" to softly bend one 2D mesh. Therefore, by `virtual_bones` in `templates.json`, **fictional joints for control (such as Torso and Neck) not drawn on the PSD are "expanded"**, and fine movements possessed by 3D data are applied to one mesh without collapse to stabilize deformation.
- **`redirect_children`:** An instruction to automatically reconnect the connection destinations of the original children (arms, etc.) to the expanded fictional bones. This maximizes the beauty of the animation while maintaining the simplicity of the drawing.

C. Other Important Parameters

- **`face_bone_keywords`:** `hair`, `eye`, `mouth`, etc. Bones containing these keywords are specially treated to maintain their shape, protected from severe physical deformation (binding) of the body.
- **`mismatch_indicators`:** A collection of hints for the system to guess "this is that bone" when 3D BVH bone names and template names do not match.
- **`search_radius_factor` (**LayerWeaver**):** The magnification of the radius when the drawing software on Krita searches for "which bone this image part should belong to."

3.3 Core Rules: Dictionary of Technical Parameters in `templates.json`

`templates.json` is the core rule of the Weaver system and is a practical configuration guide for customizing behavior without touching the program at all.

Parameter Name	Technical Details and "Author's Intent"
<code>bind_all_unassigned_layers</code>	When set to <code>true</code> , floating layers with no names in the CSV are also auto-bound to the nearest bone guessed from coordinates. It's a safety net to prevent "small shadows forgotten to be colored" from being left behind.
<code>strict_match</code>	If <code>true</code> , analysis is skipped if the layer name and template name differ even by one character. A strict mode to avoid unintended confusion when managing many characters.
<code>search_radius_factor</code>	Threshold for how far to look for markers for an island. Finely adjust according to the character's proportions (fatness) to prevent misjudgment.
<code>flexi_binds</code>	Defines which bones among all body bones are allowed "non-rigid (flexible)" physical deformation. Define parts you want to have organic movement, such as skirts and tails, here.
<code>face_bone_keywords</code>	"Exclusion keywords" to remove specific bones from overall physical binding calculations (deformation). Bones containing words registered here (eye, mouth, etc.) are protected from mesh distortion due to large body movements and are processed to maintain their shape.
<code>variation_mouth_maps</code>	Definition of mouth variation names (phonemes) and loading order. A "dictionary" to fully automate lip-syncing and directly pour the

Parameter Name	Technical Details and "Author's Intent"
	rhythm of words in 3D motion into 2D.
<code>joint_maps</code>	A " Bone Name Translation Dictionary " connecting bone names on the 3D motion (BVH) side with logical bone names inside Weaver. It serves as a bridge to convert different naming on the 3D side for each software into a common language that Weaver can understand.
<code>redirect_children</code>	A logical remap to automatically reconnect the connection destination of original child bones (wrists, etc.) from the "original parent" to the "newly established fictional parent" when <code>virtual_bones</code> (fictional joints) are added.
<code>is_accessory</code>	A flag to auto-detect parts using marker names (e.g., <code>@Skirt>Hip</code>) as a trigger. Since rigging starts only when a marker exists, it has extremely high flexibility, allowing the same template to be shared between characters with and without skirts.
<code>reorder_to_template_joints</code>	Regularizes the export order of rigging files according to the template. A stabilization function to keep the file structure completely constant during batch processing on Moho.
<code>rig_perspective_base_z</code>	[RigWeaver] A constant defining the strength of perspective baked into the Moho rig. Recommended around 4.0 for data with strong perspective like TDPT.
<code>motion_perspective_default_z</code>	[MotionWeaver] The default perspective intensity (Camera Z) used during 3D-to-2D conversion. Standard value is 8.0.

3.4 CSV Files: Loose Coupling of Physical (Island) and Logical (Rigging)

- **islands.csv**: A record of **physical facts**: "This drawn island has these joint markers on it."
- **rigging_map.csv**: A record of **logical instructions**: "Move this island (meat) with which logical bone (bone) of MotionWeaver." By separating these two, even if the names of 3D bones change, users can reuse existing drawing data just by rewriting one row of the CSV correspondence table.

3.5 File Placement: Project-Specific Settings and System Standards

Weaver automatically finds the most suitable configuration files according to the project you are working on. Basically, place files according to the following rules:

- **Project-Specific Files** (`templates.json`, `islands.csv`, etc.): Place these in the same folder as your PSD file or Moho project. This is effective when you want to perform custom definitions specific to a particular character. Weaver checks this location first.
- **System Standard Resources**: Place these in the `Moho Installation Directory / Scripts / ScriptResources / rigweaver` folder. It is convenient to centralize common templates used across multiple projects (such as standard humanoid types) here.

[!TIP] Operation Without Polluting the Script Folder In the Weaver series, there is no need to rewrite the contents of the script body (scripts/tool). Since all settings are managed in either the "project

folder" or a "dedicated resource folder," settings will not disappear even during script updates, allowing for safe operation.

4. MotionWeaver: 3D Motion Hub and "Emotional Verification"

MotionWeaver is not mere conversion software. It is the "control tower" connecting the 3D world and the 2D world.

4.1 Aspects as a Universal Motion Hub

MotionWeaver is not a Moho-only tool. The function to read BVH and cut it out as a "clip" ideal for animation production is also a powerful weapon in **3D production with Poser and Blender**.

- **Starting Point of 3D Workflow:** Create sophisticated actions in 3D software and trim and save only the necessary parts in Weaver. This allows you to share reliable 3D production power across both 2D and 3D projects.

4.2 `_3d.csv`: "Performance Record" by Your Exclusive Actor

The output `_3d.csv` is not mere coordinate data. It is the "completed performance" itself, carefully converted and projected onto a 2.5D canvas (Moho space) from complex 3D movements.

- **Concentrate on Drawing:** This file is intentionally designed so that "users cannot tamper with the contents." This is so that difficult numerical adjustments are left to the system, and creators can pour all their heart and soul into "drawing convincing pictures." Refer to **"Chapter 5: Management of Moving Assets"** for the concept of long-term data storage, reuse, and asset management.

4.3 Visual Verification: Viewer Function (`csv_viewer.py`)

MotionWeaver is equipped with a **Viewer (`csv_viewer`)** for visually checking output coordinate data.

- **Preview Before Rigging:** Before reading into Moho, you can check in 3D space whether the skeleton is moving correctly and whether joint connections are not collapsing.
- **Diagnostic Tool (Future Prospects):** A diagnostic function currently under development is planned to automatically detect contradictions between PSD structures and rigging maps and return feedback such as "The spelling of this layer name is wrong" or "Markers are missing for this part."

4.4 Emotional "Deformation": High Angle, Low Angle, and Viewpoint Control

"Impact" unique to anime is not born from only 3D faithful to reality.

- **Freedom of Viewpoint:** "High Angle (Fukan)" and "Low Angle (Aori)" directions can be flexibly adjusted by manual settings.
- **Emergency Escape to 2D:** If you are producing on a 3D basis and think, "I want to break the shape more boldly!", please shift to Moho using `_3d.csv` output from the same BVH. With 2D vector operations, anime-like exaggerated expression can be realized in seconds.

5. Management of "Moving Assets": Data Storage and Character Reuse

In the Weaver system, data evolves from mere "records" into reusable "assets." Here, we explain the philosophy for achieving efficient production across tools.

5.1 Concept of Data Storage in Animation Production

- **Distinction Between Intermediate Data and Final Actions:** `_3d.csv` for Moho and BVH files are merely "intermediate action data (waypoint)" for bringing 3D performance into 2D, and are not the final product.
- **Completion on the Timeline:** Final actions, including the addition of effects and brushing up of movements, should be built on the "timeline" of Cartoon Animator 5 (CTA5) or Moho.
- **Choice of Storage Format:** When saving actions, there is no need to write back to BVH or `_3d.csv`.
- **Reason (Retention of Effects):** If saved in an intermediate data format (CSV, etc.), additional effect information cannot be saved, so it is more appropriate to save as "Actions" in each software's unique format for proper organization and management.

5.2 Character Reuse and Efficiency (Choosing the Right Tool)

The software used in this workflow is not just for moving characters with CSV data; it is possible to develop moved characters as "moving assets."

- **Cleanup and Expansion of Characters (Moho):** It is possible to clean up moved characters with vectors and add further actions and save them.
- **Character Replacement and Diversion (CTA5 / Moho):** If you want to change characters at the beginning of each section, you can divert the actions on the timeline as they are by replacing the character itself.
- **Minimizing Corrections:** When replacing characters, there may be a need for re-correction of vector data, but the correction work will be minimal. Especially in the case of CTA5, vector creation is not possible, but the fact that the same action can be reused surprisingly easily is an extremely powerful weapon.

5.3 Inheritance of Bones and Effects (The Power of Common Bones)

- **Commonality of Bones:** Even if characters are different, if the bone structure is the same, you can continue to use actions for the same bones even if you replace the character.
- **Inheritance of Effects:** If effects are set for bones, and the bones are common, that effect setting is inherited as is even if you replace the character.
- **Dynamic Detection of Accessories:** Since accessories are dynamically detected from markers, even for variation characters with different costumes (e.g., with or without a skirt), the same bone set settings can be reused as is. The system discovers "parts that are there" each time and builds a rig.

5.4 Optimization of Production Cost: Scene-Driven Rigging

The workflow recommended by the Weaver series is **"Movement (3D) first, Rig (2D) later."** By following this order, production costs can be dramatically reduced.

- **Avoid the Trap of "Universal Rigs":** In traditional 2D production, a tragedy often occurs where people try to create a "universal character that does not collapse no matter where you look from 360 degrees" first, spend weeks just rigging, and end up not using most of it.

- **Clean Up Only Necessary Angles:** By deciding the angle (camera work) first with BatchWeaver, the "angle" and "number of layers" really needed in that scene become clear.
- **Production Without Backtracking:** Backtracking such as "I built a rig, but I lacked material for the angle I wanted to use" cannot happen in Weaver. Because the 3D guide always shows the "shortest route" for the drawings you should draw.

6. Professional Automation: Batch Integration

The ultimate in automation, which can be called the culmination of the Weaver system, built by the author to achieve speeds comparable to professional production sites.

6.1 Turning Moho into a "Video Storyboard and Controller"

In this pipeline, Moho is not mere animation software, but transforms into a **"Digital Director's Console"** that governs the direction of the entire project.

1. **Skeleton as a Moving Underlay:** Using the skeleton of the read `_3d.csv` (performance data) as a guide, assemble a direction plan on Moho such as "shoot from this angle at this timing (rotation of camera/layer)."
2. **"Cutting Out" Direction with Markers and Still Image Export (Video, Still, and Pose-to-Pose Interpolation):**
 - Place Moho "Markers" on the timeline. Any marker with a label (name) is automatically targeted for **PNG format export** of that frame. To clearly indicate that the exported PNG is a pose for video/stills, it is saved with a `_p` suffix added to the filename (`{label}_p_f{frame}.png`).
 - **Game-Changing "Pose-to-Pose (Pose-to-Pose Interpolation)" Workflow:** Markers with a duration (width > 0) set by dragging are recognized as "Video Cuts (CSV/Batch)," while markers with a width of 0 are treated as "Still (1-frame) Cuts." Moreover, **still markers with a duration of 0 are now automatically output simultaneously as "1-frame only 3D CSV data" (with a suffix `_f[frame]_s`) along with the PNG export.** This is the ultimate good news for animators. By extracting only the representative key stances or key poses from 3D motion data using markers and retargeting them, and then connecting each pose using Moho's powerful vector and bone auto-interpolation features, **"highly efficient and smooth animation production via Pose-to-Pose interpolation"** can be established in an instant. This makes it dramatically easier to not only bake long motions from scratch but also inject only the necessary key poses and customize details manually for creative drawing direction.
 - **[Mode 2 Special: Automatic T-Pose Marker & Template PNG Generation]:** By selecting **"Mode 2: Add T-Pose (Frame 0)"** during the `BatchGen` (Step 4) mode selection, the system automatically checks if a marker already exists at frame 0. If none is found, it creates and inserts a timeline marker labeled **"T-Pose"** at frame 0. This ensures that when the batch generation is completed and the still image export trigger runs, the initial pose is automatically rendered as `T-Pose_f0.png`. This PNG serves as a perfect underlay for illustration layout during the subsequent Krita integration (Layerweaver). If another manual marker already exists at frame 0, this automatic insertion is safely skipped by the conflict prevention logic to protect your existing work.
 - **Organizing Cuts by Layers:** By creating group layers such as `story1` and `story2` within the `3D_Angle_Controller` and placing timeline markers (TimelineLabel) on each layer, you can organize and compare multiple cut-out plans within a single project.

3. **Fully Automated Scan of Information:** When the script is executed, it high-speed scans the global timeline and markers on the selected layer. Regardless of which layer the markers are placed on, animation data is always accurately extracted from the **3D_Angle_Controller**, allowing for safe directorial management by separating cuts into layers.

6.2 Auto-Generation and Mass Production of Execution Program (RUN.bat)

Executing **RigWeaver Step 4** (BatchGen) is not mere data output. It is the work of assembling "switches" to move a huge factory while simultaneously performing "Photography (PNG)."

- **Instant Setup:** By integrating direction information extracted from Moho, **RUN.bat (execution batch file)** and the corresponding **CSV instruction sheet** are created in the current folder in an instant.
- **[Fully Portable (Self-Contained) Package Function]:** If you save the batch file in a custom new folder separate from the original BVH, the system automatically copies the **original BVH file** and all required **configuration files (such as templates.json)** into that destination directory. Concurrently, the BVH paths written in **RUN.bat** are automatically transformed from absolute paths into relative path references pointing to the copied BVH filename. This decouples your workspace from any machine-specific absolute path limitations, allowing you to compress, share, or migrate the entire folder to other PCs while maintaining a completely executable double-click-and-run package (assuming **MotionWeaver.exe** is registered in the environment variable **PATH**). *If you save in the same folder as the source BVH (Quick Save), this copy process is safely auto-skipped to prevent file locks or redundant overrides.
- **[Manual File Naming Rule for Custom Directory Saves]:** When choosing "No (Select Folder...)" to save in a custom directory, the Moho Pro Lua API (**LM.GUI.SaveFile**) cannot pre-populate a default file name in the system file dialog. Therefore, you must **manually enter RUN or RUN.bat in the filename box** before clicking the save button. This will automatically align the generation of the companion CSV as **RUN.csv**. For the fastest workflow, use "Yes (Quick Save)" which immediately outputs to the original BVH directory without typing.
- **Automated Still Export (we_makepng.lua):** If labeled markers are found, a confirmation dialog "Export PNG now?" appears immediately after batch generation. Simply approving this will batch-save all specified frames to the project folder.
- **The Pleasure of Batch Instruction:** Just double-click the created **RUN.bat**, and MotionWeaver starts up at high speed in the background. Dozens of specified angles and hundreds of frames of scenes are exported one after another without requiring your intervention.
- **Mass Production over a Wide Area by BatchWeaver:** By utilizing **BatchWeaver.py**, it is even possible to apply uniform processing to all BVH files in a specific folder. This makes it a reality to automatically prepare dozens of patterns of motion material needed for a project in a few minutes.

This is the crystallization of passion where a single self-taught person's passion for "wanting to take it easy" resulted in successfully bringing a professional mass production system to a personal environment.

7. Techniques to Deepen "Deformation" and "Emotion" of Animation

Weaver arranges the "logic," but it is your job to add "rich expression (emotion)" there. Here are fine-tuning techniques for mastering the system found by the author in daily production.

7.1 Automatic Mechanism: Inheriting Binding and Unitizing

Parts groups such as `@Head`, `@Hand`, or `@Skirt` auto-adsorbed by RigWeaver are sublimated from mere image layers to "intelligent units" by passing through the `Weaver_SmartSwitch` script.

- **Choosing Between Switch (Single Display) and Group (Batch Display):**
 - **Facial Expressions and Hand Poses:** These are gathered as **Switch Layers** that "display only one of them," allowing for instantaneous replacement.
 - **"Fluttering" of Skirts and Frills:** Multiple parts groups (masses of diffusion) drawn dispersedly can be integrated as a single **Group Layer**, and all parts can be displayed simultaneously.
- **Core: "Inheritance" and "Promotion" of Binding Information:** This is one of the most powerful points of the Weaver system. Normally, simply grouping existing layers in Moho causes original individual binding information to be lost or requires reconfiguration for the group. However, SmartSwitch **automatically applies the "Island binding information" possessed by the original @Tag layers to the newly generated group (or switch) layer itself.**
- **Benefit to Creators: Vector Cleanup and Line Emphasis:** Since the group layer itself is correctly bound to bones, even if you later add cleanup with vectors to layers within it or perform emphasis of line drawings (detail-up), **all additions perfectly follow the movement of the joints.** This allows "precise drawing unique to 2D" to coexist without contradiction on "stable movement based on 3D."

[New] "Instant Part Creation" from Bone Selection

From SmartSwitch v3.2 onwards, the script now supports execution while **a bone (especially PinBones like `_Scale`) is selected**, in addition to layer selection.

- **Behavior:** If you execute the script with a bone selected and no layers selected, it automatically generates an "Empty Vector Layer" and a "Parent Group" linked to that bone, completing the binding immediately.
- **Benefit to Creators:** When you want to add new parts like a chest badge or arm accessories to an existing rig, simply select the bone and press the button. A "clean canvas for drawing" will be prepared in the correct hierarchy without any worry of position shifting.
- **Naming Automation:** Since the selected bone name is automatically assigned to the created layers, the effort for management is also kept to a minimum.

7.2 Controlling "Texture" with Fine-Tuning of Strength (Force)

Strength settings in `templates.json` are a means of reflecting the "hardness of clothes" or "flexibility of the body" of a character.

- **For Heavy Costumes:** By slightly increasing `hip_force` and decreasing `flexi_force`, you can express the texture of thick clothes moving in close contact with the body.
- **For Supple Creatures:** Conversely, by setting `flexi_force` high, you can produce unearthly eeriness or elegance where the body waves more than the movement of the bones.

7.3 "Ultimate Safety Device (Fail-Safe)" Embedded by the Developer

To increase technical stability to the limit, there is a **"hard-coded safety device"** inside the program (`RigWeaver.lua`) that activates even if configuration files are ignored.

- **Forced Shoulder Connection Logic:** In processing within `ExecuteCreateBones2D`, shoulder bones (`LShoulder` / `RShoulder`, etc.) are **forcibly reconnected as children of "Torso"** regardless of the

structure of CSV or 3D data.

- **Purpose:** This promises stable mesh deformation at all times, not allowing 2D rig shoulders to detach from the torso no matter how tricky the 3D motion read is. This is the ultimate safety device that can be called the "invisible guardian deity" of Weaver, reached after the author struggled with thousands of crashes.

7.4 3D and 2D Hybrid: When to "Emergency Escape"

The true value of Weaver is that you can switch between the stability of 3D and the freedom of 2D at any time.

- **When to Stay in 3D:** Walking, dancing, scenes involving many people. These movements where "consistency" is questioned are best suited for 3D projection by MotionWeaver.
- **When to Leap into 2D:** "Key frames" where the character's emotions explode, "exaggeration of perspective" ignoring physical laws. In such scenes, do not be bound by the constraints of 3D data; boldly manipulate each layer (regardless of whether it's raster or vector) imported from the PSD with mesh deformation or vector operations. The trick to mastering Weaver is to use the accurate actions provided by `_3d.csv` as a foundation, while telling "lies" unique to 2D on top of it.

Appendix: Troubleshooting and the Author's Wisdom (FAQ)

Walls the author hit while using the tools through self-study and how to overcome them.

Q1. The image does not follow the bones, or it stretches in a strange direction.

- **Cause:** Is the marker placement too close to the boundary of the "Island"?
- **Solution:** When LayerWeaver recognizes an image, it gets lost if there is no marker within the range of `search_radius_factor`. Place markers firmly "inside the meat of the part."

Q2. I want to create a new animal or a special character.

- **Answer:** Just by adding a new section (e.g., `quadruped`) to `templates.json` and defining `target_hierarchy`, the system understands that skeleton.
- **Wisdom:** The shortest way is to copy existing `human_v2` and just replace the bone names. By using `virtual_bones`, any distorted skeleton can be treated as "straight and easy-to-handle bones" on Moho.

Q3. I am uneasy about the accuracy of the data.

- **Answer:** Execute `csv_viewer.py` without hesitation. You can see the "raw 3D conversion results" before flying to Moho. If the skeleton is walking correctly here, it can be immediately judged that the cause of the problem lies in the rigging settings on the Moho side (such as a CSV error).
- **[Future Prospects]:** Currently, a further "**Diagnostic Tool**" is planned to be mounted on `csv_viewer.py`. This is a function that pre-checks whether there are contradictions between the structure in the PSD and the definition in `rigging_map.csv` and warns of "impossible wiring" (currently under development/incomplete). Once completed, the trial and error of self-taught people should change to production with even more conviction.

Q4. Is it okay to place multiple motions (CSVs) at different points in the timeline and bake them?

- **Answer:** "Yes, it is fully supported. You can safely bake sequential motions chained together along the timeline."
- **Reason:** The latest Step 4B. Bake & Bind ([ExecuteBakeMotion2D](#)) is equipped with a "Multi-Action Bake" feature that automatically scans all CSV paths and start frames ([path|start](#)) recorded on the timeline markers. Consequently, individual CSV scale data and animation durations are correctly applied to each action interval, allowing you to bake motions starting mid-timeline without collapsing proportions or scattering character parts.
- **Author's Advice:** You can easily stitch different dances or action files along the timeline, place markers, and bake them all at once into clean 2D rig animations (Pose-to-Pose workflows are also easy to set up). Note that the basic rig construction (Step 1A–2B) and initial artwork alignment (Rest Pose) still absolutely require "Frame 0 (Setup Mode)" as their reference point, so the initial rigging setup must be done at Frame 0.

Q5. The perspective in Moho looks different from the 3D source, or the magnification is too intense.

- **Answer:** This can be solved by adjusting the perspective constants in [templates.json](#).
- **Solution:**
 - For **RigWeaver**, adjust [rig_perspective_base_z](#) (decreasing it from 8.0 to 4.0 strengthens the perspective effect during baking).
 - For **MotionWeaver**, adjust [motion_perspective_default_z](#).
- **Author's Wisdom:** When using TDPT or other data with extreme depth changes, setting these values to **4.0** minimizes the discrepancy between 3D depth and 2D visual scale, resulting in a more natural "internalized perspective" for the rig.

Q6. What does the warning after Step 4A mean?

- **Answer:** It is a safety device informing you that although bones are created (4A), the images are not yet "Baked & Bound" to them.
- **Solution:** Proceed to **Step 4B "Bake & Bind"**. This bakes the motion into the rig and ensures correct mesh deformation.
- **If stopping here:** If you wish to stop after 4A without proceeding to 4B, you must run [Menu > Bone > Reset All Bone Rigging](#) in Moho to reset the rig. Failing to do so will leave the character in an unstable state where mesh deformation may be broken or unresponsive.

Q7. Can I execute batch generation (BatchGen) from actions retargeted in the middle of the timeline (e.g., Frame 48 onwards)?

- **Answer:** "The batch generation (3B. BatchGen) itself must be performed on a timeline aligned to Frame 0/1 (synchronized state)."
- **Reason:** Batch generation ([BatchGen](#)) references Moho marker frame positions to generate commands for slicing the raw BVH file. Thus, the Moho timeline frame must match the original BVH frames 1:1, or the sliced ranges will shift. This is why a "Timeline Sync Warning" dialog forces you to check the synchronization before running BatchGen.
- **Author's Advice:** However, once the CSV paths are successfully registered to the timeline markers via BatchGen, the subsequent "Bake & Bind (4B)" process can be executed at any arbitrary position on the timeline. Therefore, keep in mind this workflow separation: execute "Action Slicing (BatchGen)" in a

synchronized state, and then you are free to place the generated actions anywhere on the timeline to run "Motion Baking (Bake)."

Q8. My 2D character illustration gets skewed or squashed after loading motion, and the Hip bone points straight down.

- **Answer:** In previous versions, baking raw X/Y/Z 3D rotations directly onto 2D image layers caused the artwork to skew or shear unnaturally. Furthermore, Hip bone movements would occasionally flip or break due to gimbal locks or Euler angle singularities in the 3D data. In the latest version, however, the spatial 2D projection (matrix transformation) and double-stage angle unwrap (jump prevention) algorithms have been highly refined, completely eliminating sudden flips or abnormal joint behavior.
- **Solution:** Now, **running with the "2D Mode" checkbox disabled (unchecked = 3D Mode) is the highly recommended standard workflow.**
 - **With "2D Mode" Disabled (Unchecked = 3D Mode):** All X/Y/Z 3D rotation components are formally baked onto the layers. This allows the 3D motion's rich rotational details to adapt naturally, expressing organic body twisting and visual depth (foreshortening/perspective scale) without any bone flips or image distortion.
 - **With "2D Mode" Enabled (Checked):** Use this mode only for special assets where you want to protect the flat silhouette of the original artwork 100% (X/Y rotations are fixed to 0, and the Hip bone orientation falls back to the projected 2D vector relative to the spine).

This reference guide will continue to evolve along with your creation. Once you have learned the logic, then tell "lies" to your heart's content and weave your own story.

8. Professional Technical Specifications

If you want to understand deeper system behavior or the logic of configuration files, please refer to the dedicated specifications for each tool below.

- **LayerWeaver: Professional Technical Specifications** ~ Depths of Island Analysis, Threshold Processing, and Naming Engineering ~
- **MotionWeaver: Professional Technical Specifications** ~ 3D Coordinate Projection, Noise Removal, and Mapping Logic ~
- **RigWeaver: Professional Technical Specifications** ~ 5-Step Pipeline, Auto-Binding, and Hierarchy Optimization ~