

RigWeaver: Beginner's Rigging Guide (v10.0)

~ Practical Guide for Efficiently Connecting Illustrations and Motions ~

Last Updated: 2026/04/21

[!IMPORTANT] Before Reading This Guide: Confirmation of Supported Assets RigWeaver supports only **PSD files with simple structures created using LayerWeaver**.

It **does not support advanced templates such as G3 characters** from other companies' products (Cartoon Animator 5, etc.). Those products have their own excellent designs and cannot be substituted by this tool. Please note that using unsupported assets will lead to errors or rigging collapse.

RigWeaver is an auxiliary tool for logically connecting 3D motion data to illustrations (PSD) drawn by creators, allowing them to smoothly take the "first step" of animation production.

When the drawings you created gain 3D movement and start moving, it would normally require a massive amount of manual bone adjustment and trial and error, but RigWeaver takes over that "tedious calculation." This is not just an automation tool; it is a "production assistant" for reducing frustration caused by technical walls on the path to realizing the animation you envision.

This guide will walk you through the "bridge-building procedures" for moving characters step-by-step.

Chapter 1: Preparation: "Signs" in Drawing Data and Necessary Data

The key to successful rigging lies in correctly engraving "signs" in the PSD (illustration data) indicating which bone each part should follow.

1.1 Layer Naming Conventions: Utilizing @ (Bone Name) and > (Hierarchy)

RigWeaver scans the layer names in your PSD and determines which bone each part should follow.

- **@ Tag (Specifying Binding):** By putting @ (Bone Name) in the layer name, you specify which bone's movement that part follows.
 - **Example: @Head:** Layers with this tag are automatically linked to Moho's head bone, and their movement is synchronized.
 - **Example: Art@Hand:** The image "Art" will follow the "Hand" bone, following the wrist's movement.
- **> Tag (Parent-Child Bond):** Used when wanting to create finer parent-child relationships under a specific bone.
 - **Example: Sword>Hand:** Places the Sword as a child of the Hand. This builds a logical parent-child relationship where if the hand moves, the sword also moves, the moment it is imported.

1.2 Efficiency in Binding Settings: Preparation for SmartSwitch

Organize your layers in advance to use the "packing function" called **SmartSwitch** later.

- **Recommended Grouping:** It is a trick to gather variation expressions (eyes, mouth) or accessories consisting of multiple parts (weapons, clothing decorations, etc.) into a single folder.
- **Inheritance of DNA:** If you give bone names to each layer with the @ tag, when you group them later, the information of "which bone the entire group should follow" (DNA) is automatically inherited to the upper hierarchy.

[!TIP] **Recommended Brush for Markers: Pixel Art (No Blur)**

When drawing or adjusting markers in Krita (LayerWeaver), we recommend using the **"Digital"** -> **"Pixel Art"** category, specifically brushes like **"Pixel Art Fill."**

These brushes are "blur-free and 100% opaque" by default. This allows you to take full advantage of the internal `alpha_threshold` logic, ensuring that parts are precisely and reliably segmented even with a tiny **1-pixel gap**.

1.3 Final Confirmation of Necessary Data: Placement and Consistency of Essential Datasets

Before starting rigging, confirm that the following data are gathered in the same folder.

- **"Pairing" of Files:** For RigWeaver to correctly recognize your character, the following two filenames must match:
 1. **Character_A.psd** (The illustration itself)
 2. **Character_A_rig.csv** (Rigging Map: Instruction sheet for which part to attach to which bone)

[!NOTE] **About _3d.csv (Motion Data)**

The performance data `_3d.csv` does not need to match the PSD name. Rather, the standard way to use Weaver is to sequentially load motion files with different names—such as "walk," "run," and "dance"—for a single character to "change performances."

1.4 Strategic Rigging: Why "3D Direction" Comes First

The maximum efficiency recommended by the Weaver series lies in **"solidifying camera work in 3D before making the rig."**

- **Eliminating Waste:** You don't need to create a "360-degree universal rig" first as in traditional 2D production.
- **Identify Only Necessary Angles:** By solidifying direction in 3D first, you know from the start "the angles of the character and number of frames truly needed" for that scene. You can completely eliminate "backtracking" and "time-wasting" of endlessly creating rigs for angles you won't use.
- **Maximum Effect with Minimum Effort:** Draw only what is needed, rig only what is needed. This "subtractive production" is the strongest weapon for self-taught creators to complete works individually.

Chapter 2: "3 Stages" of Utilization until the Character Moves

RigWeaver does not require you to "wait" until all settings are perfectly finished. You can start moving characters in stages according to your work progress. Mastering this can prevent the tragedy of "redrawing" later.

2.1 Stage 1: Bone Previs (Advance Confirmation of Movement Timing and Direction)

In the early stages when illustrations are not yet drawn or the design is not solidified, if you have 3D motion data ([_3d.csv](#)), you can move "only the bones" on Moho.

- **What can be done?:** You can confirm the core of direction at this stage, such as the timing of performance, the sense of distance from the camera, and the placement of characters on stage.
- **Benefit?:** Even without "drawings," you can create a "moving video storyboard" based on actual motion, eliminating directional mistakes in upstream processes.

2.2 Stage 2: Layer Proxy (Verification of Character Thickness and Overlap)

Once rough illustrations are ready, "temporarily attach" them to bones and move them immediately.

- **What can be done?:** You can verify with actual images whether "arms are hidden by the torso" or if there is any discomfort in "front-to-back relationships (depth)" when the character moves.
- **Point:** This is a phase where you can physically check if it "works as a drawing" while skipping all tedious rigging settings. You can discover "lack of drawing" or "alignment shifts" in assets early here.

2.3 Stage 3: Full Rigging and Cleanup Sync (Production Completion Flow)

This is the highest-density completion phase where everything is ready, and final animation is exported.

- **What can be done?:** Your detailed 2D vector lines (cleanup) will perfectly synchronize with the accurate 3D movement.
- **Primary Benefit of Weaver:** **"By tracing over the moving draft, the lines you draw immediately synchronize with the movement."** This smooth cleanup experience is a major benefit provided by the Weaver series.

Chapter 3: Shortest 5 Steps: Operational Procedures (Part 1: Stage Construction)

Finally, the practice. RigWeaver programs are listed under [Scripts](#) -> [RigWeaver](#) in Moho's menu bar. Rigging is completed just by pressing buttons **"in order from left to right,"** but before that, there is "the most important preparation."

3.0 Step 0: PSD Import and "Scale Confirmation" (The Most Important Preparation)

Before launching RigWeaver, you first need to bring your character (PSD) onto the Moho stage and **"adjust the scale to match the size of the stage."**

- **Operation:** From Moho's menu [File](#) -> [Import](#) -> [Moho Layer...](#), select and import the rigging PSD exported with LayerWeaver.
- **Most Important Point:** Immediately after loading, the PSD may be too huge or too small for the Moho workspace. **Before starting rigging, confirm the scale with the layer scale tool so that the illustration fits nicely within the workspace (the blue frame in the center of the screen).**
- **Why is it important?:** Each process of RigWeaver performs calculations based on this "confirmed coordinate." If you press buttons in order while forgetting scale adjustment, layer positions will shift drastically, or bones will be generated in unexpected places.

3.1 Step 1: Setup & Viewer (Project Setting and 3D Guide Projection)

This is the ceremony of overlapping "3D performance (Soul)" onto the illustration (Body) placed on stage.

- **Operation:** Execute **1A. Load 3D Data**.
- **What happens?:** Motion data (**_3d.csv**) is loaded, and a **"Viewer (Red Skeleton)"** appears in a size that fits the height of the canvas (workspace).
- **Engineering Value:** This becomes the "Guide for the Soul." Since the skeleton size is always auto-adjusted based on the workspace, **if you matched the PSD to the workspace in Step 0, the skeleton and illustration appearing here should roughly match in size.** (If fine adjustment is needed, fit the illustration scale perfectly to the skeleton here).

3.2 Step 2: Build Skeleton (Skeletal Construction and Illustration/Pose Synchronization)

Next, find the "signs (markers)" engraved in the PSD and forge a final skeleton synchronized with the body.

- **Operation:** Execute **2A. Build Hierarchy**.
- **What happens?:** Based on the Rigging Map (**_rig.csv**), a bone hierarchy accurately capturing the joint positions of the illustration is automatically assembled.
- **Magical Calibration (Step 2B: Align Base Pose):** Even if the illustration pose (initial posture) is slightly shifted from the default posture of the 3D data, don't worry. Executing this step makes RigWeaver judge based on the illustration pose and automatically corrects (synchronizes) all initial bone angles to match the illustration.

Chapter 4: Shortest 5 Steps: Operational Procedures (Part 2: Injecting the Soul)

Stage and skeleton are ready. From here, it's the exciting process of injecting 3D dynamism into the character and completing it as a final "moving body."

4.1 Step 3: Import Motion (Injecting Movement and Completing Video Storyboard)

Transfer the 3D performance residing in the Viewer, or **any _3d.csv file prepared in advance**, to your character's skeleton.

- **Operation:** Execute **3. Retarget Motion**.
- **Retarget Options:** An options dialog appears upon execution. Select according to your purpose:
 - **Start motion at current frame:** If checked, performance starts from the current timeline position (e.g., frame 24). Effective when wanting to connect behind another movement.
 - **Clear existing motion before applying:** If checked, all old movements already on the timeline are completely erased before injecting the new performance.
- **Selecting Performance:** After settings are complete, a file selection dialog opens. Select the movement (CSV file) you want the character to perform.
- **What happens?:** The selected performance data is transferred to bones under the specified conditions, and the character starts moving while maintaining the illustration pose.
- **State at this point:** Completion of the so-called "video storyboard." Images imported from the PSD are not yet attached to bones, but you can **confirm as a guideline until satisfied** whether the bones (skeleton) are moving correctly and if the performance timing matches the directorial plan. If the movement is different from your image, select another CSV and re-inject as many times as you like to solidify a satisfying "foundation of performance."

[!IMPORTANT] **New Feature: "Multi-Action Baking" from Timeline Markers** In the latest version, by placing multiple markers (`path|start` containing the absolute path of the import source CSV and the start frame offset) at arbitrary positions on the timeline, you can automatically scan and batch-bake/bind all actions during the subsequent **4B. Bake & Bind** execution. This dramatically reduces the effort of manual import and switching, allowing complex performance stitching to be handled visually and safely on the timeline.

[!TIP] **Pro Tip: Pose-to-Pose (Pose-to-Pose Interpolation) Workflow**

Placing a "still image marker" with a duration of 0 on the timeline exports a "1-frame CSV (`_f[frame]_s`)" simultaneously with the PNG output. By injecting these sliced key poses into the Moho timeline, you can establish an extremely efficient **"Pose-to-Pose workflow that connects key poses using Moho's unique vector and bone auto-interpolation."** This avoids the high computing load of baking long, frame-heavy motions, allowing you to automatically place key poses and customize details manually for creative direction.

4.2 Step 4A: Convert 2D (Spatial Translation/Transfer Calculation from 3D to 2.5D/2D Angles)

This is the process where the intelligence of the Weaver series is most demonstrated, translating complex 3D rotations into "charming angles" that look best for 2D/2.5D animation.

- **Operation:** Execute **4A. Build 2D Rig**.
- **What happens?:** Movements created with 3D "Iron Logic" are converted into controller bones that allow 2.5D "Emotional Lies."
- **Lies of Physics:** 2D bones generated here mathematically justify perspective exaggeration impossible in 3D and illustration-specific "best-looking angles," achieving animation without collapse.
- **Recommended Setting (Toggling 2D Mode):** Unlike in previous versions, the latest system utilizes highly robust spatial projection and double unwrap (anti-flip) processing. Therefore, **running with the "2D Mode" checkbox disabled (OFF = 3D Mode) is the standard recommended workflow**. This allows the 3D-specific three-dimensional scale changes (perspective) and flexible body twists to be expressed naturally and dynamically. Enabling "2D Mode" is restricted to special asset use cases where you want to completely eliminate 3D perspective distortion and protect the flat silhouette of the original illustration 100%.

4.3 Step 4B: Hybrid Bind (Linking Image and Bone: The Power of SmartSwitch)

Finally, physically connect the body (image) and bone.

- **Operation:** Execute **4B. Bake & Bind**.
- **What happens?:** The image snaps to the bones, and the "meat" of the character starts to follow the movement.
- **Hybrid Intelligence:** Binding settings, such as "soft joints (Flexi)" and "rigid accessories (Layer)," are automatically applied according to the CSV definitions.

4.4 Step 5: Finalize (Hierarchy Organization, Baking, Animation Finalization)

The conclusion of the rigging "ceremony."

- **Operation:** Execute **5. Flatten Rig**.

- **What happens?:** Temporary hierarchies (matrices) used for rigging work are destroyed and organized, and all movements are "Baked" into world coordinates.
- **Result:** This completes a very stable "Production Rig" capable of withstanding final production.

Chapter 5: Efficient Cleanup Flow: "Cleaning Up" Over Moving Illustrations

Just because rigging is finished doesn't mean your creativity is over. Rather, from here starts the real work of "finishing" to put a true soul into your drawings.

5.1 Practice of Cleanup Sync: "Inheriting Settings" within Groups

In the Weaver series, this "cleanup sync" is the most efficient technique. It's the process of cleaning up moving rough drafts with beautiful final lines (vectors).

- **"Vessel (Group)" for Cleanup:** If a part is a standalone image, group it (SmartSwitch) and attach the "Magic Spell (@ tag)" to the name (e.g., `Arm_Group@Arm`).
- **Inheritance of Settings:** You can create vector layers within that group for cleanup, or just distort the original PSD image (raster) as is with bones. **"Just trace over the moving draft and clean up,"** and the final lines will synchronize with the movement, allowing you to add exaggerations unique to 2D as you wish.

5.2 Finalizing with Groups: "Polishing Lines" using Rough as Guide

Instead of replacing images, use existing rigs as "puppet skeletons" and polish and complete lines within them.

- **Defending Overlap and Quality:** Coexist guide images and cleanup vector lines within a single part group (e.g., `Eye@Head`). This allows you to proceed with cleanup while always confirming final line connections and overlaps within the movement.

[!TIP] Reuse of Performance: Utilizing Moho Actions

Wonderful performances injected with RigWeaver can be registered and saved as "Actions," a standard Moho function. For PSDs with the same character profile (skeletal structure), you can immediately reproduce the same performance just by loading the action.

Chapter 6: Taking Advantage of 2D "Lies of Expression"

3D data is "physically correct," but that is not necessarily "charming as 2D." In this chapter, we learn craft techniques for further refining animation.

6.1 Essence of Angle Correction: Adding "2D Flair" to Raw 3D Data

Movements immediately after injecting 3D motion can sometimes feel "too raw."

- **Thinking for Correction:** On the Moho timeline, manually adjust bone angles of specific keyframes (such as a finishing pose) to "lie" slightly.
- **2D Flair:** Even if facing forward in 3D, by "daring to show slightly diagonally" for 2D purposes, you can make the character's silhouette more beautiful and powerful.

6.2 Maintaining Shape via Virtual Bones: Preventing Joint Collapse with "Additions"

When performing intense movements, neck or joint overlaps may collapse. This is a structural limit of PSDs, but it can be solved by "adding" bones.

- **Virtual Bones:** By setting `templates.json`, you can generate "auxiliary bones" for control not drawn in the PSD.
- **Guarding Shape:** For example, by creating a bone dedicated to a "collar" and controlling it so that it doesn't flip unnaturally as the neck rotates, you defend the quality of the illustration.

Chapter 7: Rescue: Checklist to Prevent Collapse (Monsterization)

Don't panic if your character suddenly takes a "monster-like" shape during rigging. Most of it is caused by simple setting inconsistencies.

7.1 Resolving Bone Flips and Twists

In some cases, the character may be flipped 180 degrees or joints may be unnaturally twisted.

- **Cause:** The rotation axis of the 3D motion (`_3d.csv`) and the direction of the 2D illustration may not match.
- **Solution:** Review the `Angle Offset` set during **Step 1A**. Also, by selecting the target bone in Moho and performing "Reset Bone," you can regain initial sanity.

7.2 Correcting Coordinate Shifts: The Ceremony of Returning to Origin

If bones and the illustration are significantly separated, check the following checklist:

- **Reconfirm Magic Spells:** Are layer names in the PSD (`@` or `>`) incorrect?
- **Severed Parent-Child Relationships:** Confirm if all bones are in a correct family tree starting from `Hip` (Waist).
- **Warning of Infinite Loop:** If the script stops with a "CRITICAL ERROR: Circular reference" warning, it's a sign that virtual bone parent-child relationships are returning to themselves (looping). Review the `templates.json` settings.

Chapter 8: Step Up: Entrance to Mass Production Engineering (BatchWeaver)

Congratulations! Having reached this far, you have mastered the rigging workflow and gained an efficient animation production technique connecting 3D and 2D.

8.1 Batch Converting 100 Motions: From Creator to Director

The true power of RigWeaver is demonstrated in the mass production phase.

- **BatchWeaver:** By executing the `batch.bat` (batch processing file) exported in **Step 4**, you can fully automate baking processes previously done manually one by one for dozens or hundreds of files.
- **Toward a Director's Perspective:** Once you draw one character and complete rigging once, from then on, simply by injecting dozens of types of performance data, a large amount of animation material will

be automatically generated within minutes.

Conclusion: Step by Step, Toward Ideal Movement

RigWeaver is a tool for reducing "production frustration" due to technical constraints and expanding the possibilities of expression. Now that you have mastered the correct procedures (Step 0–5), you are at the starting line of pursuing character performance without being bothered by complex settings.

The Weaver series will continue to evolve as a "reliable scaffold" for shaping the heat of your creation. Now, let's get started on producing your next character!