

# LayerWeaver: Beginner's Drawing Guide (v12.9)

---

~ An Efficient Data Creation Guide for Animating Illustrations ~

[!IMPORTANT] **Before Reading This Guide: Confirmation of Supported Assets** This tool (LayerWeaver) is an auxiliary tool for creating **PSDs with relatively simple layer structures** based on Weaver series specifications.

The **G3 Template Standard** is an extremely sophisticated proprietary standard designed by **Reallusion**, the developer of Cartoon Animator 5 (CTA5). LayerWeaver does not automatically generate this existing standard from scratch, nor is it a substitute for it.

The role of this tool is as a **"Translator (Bridge)"** to correctly adapt your drawings to these "wonderful existing standards." Please understand in advance that if you use the exported PSD in other third-party tools, final manual adjustments may be necessary to match the specifications of each tool.

LayerWeaver is an auxiliary tool to efficiently cut your illustrations into "movable parts" and automate the preparation for rigging. This system has a simple and absolute rule: **"Delete the sample images in the template and place your own drawing as the main subject (Art)."**

---

## 0. GUI Menu Reference (Interface Correspondence Table)

During operation, button names and colors may change. Pay close attention to the signals on the screen as you proceed.

### Main Operation Buttons (Standard State)

- **[1] 1. Analyze & Generate Preview:** The first step of analysis (layer generation).
  - **[2] 2. Build & Show Final PSD:** Joins and reconstructs the separated fragments into a new PSD.
  - **[3] 3. Save Current PSD...:** Exports the constructed rigging assets.
  - **[4] 4. Run MotionWeaver (Gen\_3d/\_rig.csv):** Shifts to the process of linking with performance data.
- 

## 1. Preparation Process: Importing (Replacing) Illustrations and "Hierarchy Rules"

For LayerWeaver to work correctly, **"how you place your drawing"** on Krita is more important than anything else.

### 1.1 Basic Rule: Delete the sample Art and place your own drawing

Replace the sample images inside the **Human** group, etc., of the template with your own drawings.

[!IMPORTANT] **Why is "Replacement" Necessary? (System Specifications)** When starting analysis, the system (program) scans the root (the top-most hierarchy) and forcibly selects only **"one node with the largest area, regardless of whether it is visible or hidden"** as the "Main Subject (Art)."

Visibility settings at the root hierarchy are ignored. Instead of hiding unnecessary sample characters, the most reliable and clean way is to physically delete them before placing your own drawing.

## 1.2 Adjusting Markers (The Cutting Blade)

The "black circles" in **Markers** are bone rotation axes (pivots) as well as **"cutters" that separate parts**. Adjust the size of the markers (diameter of the perfect circle) to be **"slightly overlapping the sides"** of the illustration parts. This is the secret to a clean cut.

### 1.2.1 Important Rules for Neck Length and Drawing (Preventing Neck Bending Issues)

To ensure the character's neck rotates (bends) correctly, the **"length of the neck (seam allowance)"** in the illustration is extremely important. If the neck is too short when exporting the PSD and importing it into animation tools like CTA5, the neck image pixels will not sufficiently overlap with the position of the neck bone (Neck), preventing the automatic weights from being applied. This can cause issues such as the neck image not bending when the bone is moved, or the head simply shifting in parallel and detaching.

- **Tips to ensure correct bending:** Draw the neck part long enough so that it **"fully extends behind the head (deeply overlapping with the head part)"**. If the neck part is sufficiently long, the movable weights for the bone will be applied correctly, allowing the neck to bend smoothly.
- **Alternative of sliding (shifting) without bending:** Depending on the design or character traits, you might prefer the neck not to bend flexibly but simply shift in parallel with the head's movement. CTA5 might automatically determine whether to apply weights for bending or simply slide the part based on the overlap length of the neck. Adjust the neck length according to your drawing intent.

## 1.3 [Advanced Technique] How to Create Animatic Assets Using Stick Figures

LayerWeaver is also ideal for creating skeletal materials for "motion previews (animatics)" before drawing the official illustrations. However, if you place "only marker dots" on a transparent layer in Krita, the system cannot recognize them correctly.

### [!IMPORTANT] Technical Requirement: Connect Markers with a Brush to Create an "Island"

LayerWeaver calculates bone lengths by detecting not just the marker dots, but the "cluster of pixels (island) between markers." Therefore, when creating stick figures for pre-visualization, **after placing the markers, quickly connect them with a brush (or enclose them with circles, etc.) to create an opaque pixel "Island"**. This is a technical requirement.

By drawing key poses like "jumping" or "attacking" as stick figures using this method, saving them as PSDs, and passing them through MotionWeaver, you can complete your own original initial poses (**\_3d.csv**). By connecting these in Moho, you can rapidly build the foundation for completely original manual animation (Pose-to-Pose).

## 1.4 [Advanced Technique] How to Register Semi-Transparent Parts (Scarfs, etc.) to a Character

If you want to cleanly import semi-transparent decorations such as scarfs or veils as part of a character, the technique of **"filling them with markers to detect them as joints (Hole) and merging them into base parts"** is effective.

### [!TIP] Procedure and Mechanism

1. **Fill with Markers:** In the **Markers** layer, fill the area of the semi-transparent part you want to register.
2. **Run Analysis:** Execute [1. Analyze & Generate Preview]. The semi-transparent area painted with markers is reliably detected as a joint (**j** number) without being affected by the alpha threshold.
3. **Integrate in the Table:** In the correspondence table on the UI screen, append the detected **j** number (e.g., **j3**) to the **"Joints" column** of the base part like the body (**Torso**) using comma separation (e.g., **j1, j3**).
4. **Build:** Execute [2. Build & Show Final PSD]. The pixels of the semi-transparent part will be cleanly merged and integrated into the base part in the exported PSD.

## 2. Step 1: 1. Analyze & Generate Preview (Button [1])

Once preparation is complete, click the button **"1. Analyze & Generate Preview."**

### 2.1 🎨 Craftsman's Finishing (Neatly Arranging Part Layers and Joint Margins)

After clicking, once the **islands** (parts) and **hole** (joints) layers are auto-generated, the button will **light up yellow**, entering a pause state waiting for your corrections.

**Note: At this point, neither "colors" nor "numbers" have been attached yet.** Use Krita's brushes and erasers to perform the task of **"neatly arranging"** these two important auto-generated layers to optimize the balance between them.

#### [!TIP] Correction Techniques:

- **Quick Tip:** If you want to finish easily, simply cut off parts of the part layer (islands) with a thin eraser.
- **Standard Correction:** If adjustment is difficult, delete the layers once generated, adjust the Markers, and then press Button [1] again.

## 3. Next Step: ▶ Run 'Analyze' Again (Yellow Button)

After finishing "arranging" the part layers and joint margins, click the yellow button **"▶ Run 'Analyze' Again"** to finalize the analysis.

### 3.1 🔍 Confirmation of Analysis Results (Preview Numbers and Table Correction)

With this click, the program inside Krita scans your corrected layers and displays the **"Analysis Results."**

[!CAUTION] **"Auto-Determination" is Not Omnipotent (T-Pose Recommended)** As a rule, the program can automatically link parts to joints only in the case of a **"T-pose with arms spread to the side."** In other poses (such as arms down), it is normal for it not to be able to determine correctly.

**Recognize the Table as "Something to be Corrected"** The correspondence table displayed on the screen is merely a "guess" by the program. You must check it with your own eyes and **manually rewrite to complete the integration** of any incorrect parts (such as unnatural Island numbers).

[!TIP] **Row Order in the Table = Layer Stacking Order in the PSD (Z-Order)** The order of rows in the table on the screen (and the saved **rigging\_map.csv**) directly dictates the stacking order of layers in

the exported PSD.

- **Determining Factor:** The final stacking order is determined by **the row order of rigging\_map.csv (and the initially displayed table)**, NOT by the layer order of markers in Krita or the order in `islands.csv`.
- **Higher rows:** Placed in the **foreground (front)** in the PSD.
- **Lower rows:** Placed in the **background (back)** in the PSD.

For example, if a character is facing left, placing the "left leg" (which should be in front) in a row above the "right leg" will maintain the correct depth relationship. Normally, the default order (from `rigging_map.csv`) works fine, as characters are typically drawn in a T-pose.

## 4. Step 2: 2. Build & Show Final PSD (Button [2])

After correcting the analysis results (table) and linking correctly, click the button **"2. Build & Show Final PSD."**

### 4.1 Final Inspection After Construction: Check by "Rotating" Parts

Once construction is complete and a new PSD document opens, be sure to perform the following "operational test" without saving it as is.

1. **Align the Axis:** Select the part layer you want to correct and press **Ctrl + T** (Transform).
2. **Align the Center:** Overlap the transformation center point (pivot) with the **point (pivot position) engraved in the "Joint Margin (Red Number)"** visible on the back side of that part.
3. **Check by Rotating:** Try dragging and rotating the part.
4. **Correction of Seam Allowances:** If there are gaps when rotating, redraw or erase on the spot to perfectly arrange the part overlap (seam allowance).

## 5. Step 3 & 4: Save and Launch

- **Button [3] "3. Save Current PSD...":** Saves the PSD.
- **Button [4] "4. Run MotionWeaver":** Shifts to animation production.

## 6. [Advanced] Tag Control and Material Design as Visual Programming

A primary design philosophy of the Weaver series is that rigging preparation can be completed simply by attaching specific tags to layer names, without having to write external configuration files. This is an efficient configuration task performed by the illustrator on the canvas.

### 6.1 Naming Rules (Tagging Logic): Automated Instructions to the Rigging Engine

Specific symbols included in layer names function as execution instructions to the subsequent construction engine.

- **"@" Tag: Spatial Snapping and Variation Expansion**
  - **Basic:** `Child Part Name @ Anchor Name for Snapping` (e.g., `Sword @ RHand`)
  - **Effect:** Snaps child parts like magnets to the parent bone's rotation axis (pivot).

- **Advanced (Switch Groups):** Attaching @Head, etc., to a group layer name allows all variations within it, such as "blinking" or "mouth shapes," to be snapped and fixed (Layer Binding) collectively.
- **">" Tag: Dynamic Construction of Skeletons (Bone Chains)**
  - **Basic:** Parent Bone Name > Child Group Name > Child Bone Name
  - **Effect:** Builds complex bone parent-child relationships within the exported PSD just through naming in Krita.
  - **(0) Parameter:** Writing it like Scarf\_Tip(0) allows you to set the influence range on the mesh to zero, leaving only the function as a rotation axis for the joint. This suppresses unnatural distortion when bending the joint.

## 6.2 Movement Settings (Material Design): Supple Movement vs. Rigid Movement

Give drawing parts "Material" properties for how they should move.

- **Tube Expression (Organic / Flexi-Bind):**
  - **Features:** A single drawn part bends "supplely" according to the movement of multiple bones.
  - **Recommended for:** Arms, legs, tails, long hair, capes.
  - **Setting:** Rubber-like smooth curves are drawn along the bone chain connected by ">" tags.
- **Robot Expression (Rigid / Region-Bind):**
  - **Features:** Moves "rigidly" in 100% synchronization with bone rotation, allowing no deformation of the part itself.
  - **Recommended for:** Swords, shields, helmets, mech armor.
  - **Setting:** Automatically applied to parts snapped to a single bone with the "@" tag, reproducing accurate joint movement.

## 6.3 Special Automation: Boundaries of Lip-Sync and Facial Control

- **Lip-Sync Auto-Mapping:** When mouth shapes are drawn in a group assigned the @Mouth tag, phonemes are automatically assigned based on the order of detection.
  - **5 Phonemes (Japanese Standard):** Simply lining up five shapes in the order of "A, I, U, E, O" supports basic conversation.
  - **15 Phonemes (Full Spec):** High-level lip-sync can also be supported by drawing in a specific order.
  - **Tool-Specific Specs:**
    - **CTA5:** Since the structure of the exported PSD is simple, basic lip-sync functions may work after loading, but operation as an official G3 standard is not guaranteed.
    - **Moho:** Similar operation is possible, but switch layer settings on the Moho side or fine-tuning according to the project is necessary.
- **Know "What Weaver Does Not Do":** Weaver does not go as far as turning blinking into smart bones or complex mesh deformation. This is because finishing directly using editors of specialized tools results in higher quality and lower man-hours. Weaver specializes in providing an "organized foundation (structure)" for that purpose.

---

## 7. Troubleshooting: SOS Protocol

The Weaver series is not an inflexible finished product.

- **Rely on Engineers:** Wasting valuable time investigating the cause of an error or a syntax mistake in a configuration file is a "failure."
  - **Philosophy:** Artists should concentrate on drawing, and technical walls should be consulted and resolved early. This "efficient progress without worrying alone" is the smooth workflow envisioned by Weaver.
- 

## 8. Finishing: Expansion to Multi-Platforms (Usage in Each Tool)

The PSD exported by LayerWeaver is a universal rigging PSD with "drafts" of bone structure and material data embedded.

- **Usage in Cartoon Animator 5 (CTA5), etc.:** The exported PSD can be imported into other third-party tools as a simple material with bones. However, it is not designed as a sophisticated dedicated template (G3, etc.). After loading, use it as a "foundation" for performing re-adjustment of weights or facial settings according to the conventions of each tool.
  - **Usage in Moho:** After loading the PSD, execute the `Weaver_SmartSwitch` script. A basic structure of bone control and mesh joining will be constructed based on the naming conventions designed in LayerWeaver.
- 

*LayerWeaver: Beginner's Drawing Guide - The first step for efficient asset production.*