

Data Structure and Automation Logic in the Weaver Series: Explanation of the Relationship between CSV and Templates

1. Introduction: Data Linkage Concept of the Weaver Series

The Weaver series (LayerWeaver, MotionWeaver, RigWeaver) is a pipeline for **connecting "3D consistency" to "free expression unique to 2D."**

In this extensive system, CSV and JSON files function as "hubs" for maintaining consistency and automating processes between different applications (Krita, MotionWeaver, Moho). As a design philosophy, creators do not need to be conscious of backend technical details, but by understanding "what these files determine," you will be able to control the production process more efficiently.

2. Roles and Definitions of the Three Core Files

We organize the three data files that support the backbone of the Weaver system, focusing on detailed information and user access rights.

File Name	Role	Main Stored Information	User Access
islands.csv	Geometric Catalog of Parts	Name, center coordinates, joint positions (Holes), and boundary areas of each part (island) from layer analysis (Pixel Analysis).	Auto-generated (Verification required)
rigging_map.csv	Bone Wiring Diagram	Parent-child relationships of bones (Hierarchy). A skeletal blueprint defining how the rig works together.	Auto-generated (Controlled by tags)
templates.json	Master Catalog	System-wide operational rules such as skeletal definitions, global binding rules, and calculation order.	System Managed (Adjusted by technicians)

3. The Master Catalog: templates.json

templates.json is the core foundation governing the operational rules (Global Rules) of the entire Weaver series. In its latest specification, the file is segmented into tool-specific sections (RigWeaver, MotionWeaver, and LayerWeaver), allowing for more precise engineering control optimized for each environment.

- **Tool-Specific Configuration Sections:**
 1. **RigWeaver Specific:** Defines rigging behavior in Moho, virtual bone structures, and the rig_perspective_base_z constant for back-calculating perspective from layer magnification.
 2. **MotionWeaver Specific:** Defines 3D motion analysis rules and the motion_perspective_default_z constant for default perspective intensity.
 3. **LayerWeaver Specific:** Defines layer analysis algorithms and part extraction rules within Krita.

- **Skeletal Concepts and Kinematic Constraints:** Holds the basic structure for each skeleton, such as Humanoid or Quadruped, and standard specifications for bone names.
 - **Hiding the "Law of Conversion":** The design of coordinate axes differs fundamentally between 3D software and Moho (such as inversion of X and Z axes). This conversion calculation is automatically processed in the backend, allowing users to control "what they see" without being conscious of spatial coordinate discrepancies.
 - **Definition of Auto-Binding Rules:**
 1. **Flexi-bind:** Rules for parts that require smooth deformation, such as the torso or neck.
 2. **Layer-bind:** Fixation rules for parts that should keep their shape rigid, such as limbs or the head.
 - **3-tier Path Resolution:** The system searches for `templates.json` in the following order of priority. This achieves both project-specific customization and system versatility.
 1. Same hierarchy as the PSD file.
 2. Within the plugin folder.
 3. Custom path described in `templates_path.txt`.
-

4. Creator's Workflow: Setting Tags by Layer Name

A feature of the Weaver series is that "naming" on the canvas familiar to the creator functions directly as rigging configuration information.

1. **Tagging (Visual Scripting):** During drawing in Krita, symbols are added to layer names or marker names.
 - `@` : Definition of grouping or accessories (e.g., `@Hat`).
 - `>` : Description of explicit parent-child relationships (e.g., `Arm > Hand`).
 - `*` : Guide marker specification (e.g., `j12*`). A non-cutter declaration used as a position reference without punching holes in the artwork. *Note: Since it is automatically removed during final PSD build, it has no impact on exported CSVs or subsequent rigging steps.*
 2. **Automated Construction of Rigging:** LayerWeaver reads these tags and automatically generates `islands.csv` and `rigging_map.csv`.
 3. **Naming = Design:** The daily act of "organizing layers" for the creator directly becomes the act of updating the wiring diagram of `rigging_map.csv`.
 4. **Immersion in Drawing:** Users are freed from the trouble of describing the backend of data and can focus on "composition" on the canvas.
-

5. Division of Roles: Expression Field and Rescue Mode

The Weaver series clearly divides technical roles so that creators can concentrate on "2D expression."

- **2D Expression Field**
 - `_3d.csv` is the "final output data" for moving characters in Moho. Once generated based on 3D information, it is not bound by 3D mathematical constraints and can be adjusted for 2D.
 - Even if the mesh is collapsed on the 3D side, it is recommended to overwrite "deformed expressions (perspective exaggeration, smears, etc.)" using Moho's Z-axis (depth) or vector deformation.
- **System Tuning (Technician's Field)**

- **Mode 1 (Standard):** Basic mode for assembling a skeleton based on the drawn pose (A-pose, etc.).
 - **Mode 2 (Force T-pose):** For "emergency rescue" if the illustration pose is too extreme and the mesh collapses. Forcibly inserts a T-pose at frame 0 to stabilize calculations.
 - **Externalization of Parameters:** If special multi-legged robot definitions or physical behavior adjustments are needed, a technician can update the entire pipeline simply by adjusting `templates.json` for a few seconds.
-

6. Conclusion: Dedication to the "Conducting" of Direction

The linkage between CSV and templates is not just a reduction of work. It is a mechanism for liberating creators from the technical work of "rigging settings" and allowing them to concentrate on the control of direction.

Particularly, the utilization of **BatchWeaver** pushes this automation into the mass production phase. The CSV issued by RigWeaver becomes an "Order Form" for BatchWeaver. Based on this order form, it is possible to mass-produce numerous animations with different angles, such as "Front," "High-angle," and "Low-angle," from a single BVH.

By building on the foundation of "3D consistency" and pursuing "expression unique to 2D" upon it, the automation mechanisms provided by the Weaver series will become a powerful tool to support your expression.