

Existing Armatures and the Setup Wizard

IN THIS GUIDE

Existing Armatures and the Setup Wizard

1. Bone Root and Component Host

Bone Root

Component Host

2. Starting Profile

3. Branch splitting

4. Virtual Endpoint

None

Extend Last Segment

Local Direction

5. Excluded Transforms

6. Discovered Preview

Unrepresented branches

Large component

Non-uniform scale

Zero-length segment

Nested under another Reactive Bone Chain

7. Create the chain, then author collision separately

8. After creation: the Reactive Bone Chain inspector

Recommended existing-armature workflow

Use the **Setup Wizard** when the object already has Transform bones representing the part you want to simulate.

Open it from:

Tools > STL Dynamics > Reactive Bones > Setup Wizard

Selecting the first bone before opening the window will prefill the root.

The Setup Wizard is deliberately focused on **chain creation only**. Collision authoring is handled later in the **Smart Collision Manager** so existing rigs and Rig Painter-generated rigs use the same collision workflow.

1. Bone Root and Component Host

Bone Root

Choose the first existing Transform that should participate in secondary motion.

Good roots are usually the first dedicated bone of the tail, hair strand, ear, strap, cable, tentacle, or other reactive part - not the entire character root.

Component Host

The host receives the `ReactiveBoneChain` component.

The wizard defaults the host to the selected root GameObject. This is usually fine. A separate host can be useful when you want components organized away from the actual skeleton.

2. Starting Profile

Every chain needs a **Reactive Bone Profile**.

You can:

- assign an existing profile;
- choose a **New Profile Preset** and click **Create Profile Asset...**

Profiles are reusable. If several ponytails or tails should feel the same, reuse one profile rather than duplicating settings per character.

See [Profiles](#), [Chain Tuning](#), and [Wind](#).

3. Branch splitting

Reactive Bones simulates linear strands. A Transform cannot rotate toward several outgoing children at the same time, so branching hierarchies need independent chain roots.

When the wizard detects branches it can enable:

Split all hierarchy branches into independent roots (recommended)

Keep this enabled for most branching hair, feathers, plants, or multi-pronged appendages.

The result is still one `ReactiveBoneChain` component, but its primary root and **Additional Roots** represent the independent strands.

If branch splitting is disabled, only the first child path through a branching joint is simulated. The wizard and inspector warn when outgoing branches are not represented.

4. Virtual Endpoint

The endpoint controls what happens after the final real Transform.

None

No virtual tip is created.

Use this when the final real bone already represents the complete simulated length. A one-bone chain with no child cannot simulate unless a virtual endpoint is enabled.

Extend Last Segment

Extends the final segment beyond the last real bone. This is the best general-purpose mode for tails, hair, ears, cables, and generated chains.

Local Direction

Creates the endpoint using an explicit local direction and length.

Use it when the final bone's outward direction cannot be inferred correctly from the previous segment.

If **Local Direction** is zero, the chain inspector will warn you.

5. Excluded Transforms

Excluded Transforms are ignored by chain discovery.

Use exclusions for:

- helper bones;
- sockets;
- decorative child Transforms;
- collision/helper objects that should not become simulated particles;
- zero-length nodes;
- branches that should not be simulated by this chain.

These are **simulation exclusions**. Automatic collider exclusions are configured separately in the Smart Collision Manager when you author collision.

6. Discovered Preview

Before creating the component, the wizard previews the Transform path and particle count.

Pay attention to these warnings:

Unrepresented branches

Enable branch splitting or configure additional roots manually.

Large component

A chain with roughly more than 128 particles is intentionally flagged. Split large systems into smaller components when independent culling or smaller collision sets would help.

Non-uniform scale

Reactive Bones supports the hierarchy, but uniform world scale produces more predictable segment lengths and collision radii. Apply unusual scale in your DCC tool when possible.

Zero-length segment

Move the child bone away from its parent or exclude the decorative/helper Transform.

Nested under another Reactive Bone Chain

This can be valid, but two components must not write to the same bones.

7. Create the chain, then author collision separately

When the setup is valid, click:

Create Reactive Bone Chain

The Setup Wizard does not:

- generate Reactive Bone Colliders;
- assign connected collider sets;
- synchronize linked armatures;
- configure world/Unity collision;
- tune collision radius.

That separation is intentional.

If the chain needs collision, open:

Tools > STL Dynamics > Reactive Bones > Smart Collision Manager

Then:

1. Set **Character / Part** to any Transform belonging to the character.
2. Choose the collision sources the chain actually needs.
3. Generate/replan sparse automatic colliders or create manual shapes if needed.
4. Set **Chain Collision Radius** for the connected chains.
5. Use **Synchronize All Connected Chains** if body, hair, clothing, or linked armatures should share the same local analytic set.
6. Enable **Create / update runtime collision group** when modular/runtime-attached parts need that binding refreshed when enabled.
7. Apply world collision settings only when independent objects/environment collision is needed.

See [Smart Collision Manager](#).

8. After creation: the Reactive Bone Chain inspector

The generated `ReactiveBoneChain` inspector is divided into:

- **Chain Setup**
- **Simulation**
- **Collision**
- **Optimization**
- **Advanced Contact API**
- **Debugging**

The inspector performs live setup validation and provides fixes for common errors.

Collision controls in the chain inspector are intentionally minimal. Collision authoring and connected-set synchronization route to the **Smart Collision Manager**, avoiding a second hidden collision workflow.

Useful actions include:

- **Open Setup Wizard** - revisit guided chain setup.
- **Split All Branches** - automatically populate missing branch roots.
- **Open Smart Collision Manager** - author/synchronize collision for this chain and its connected character.
- **Open Runtime Debugger** - inspect live performance and state.

Recommended existing-armature workflow

1. Start with one small, clearly defined appendage.
2. Use a profile preset close to the desired material.
3. Keep branch splitting enabled.
4. Use **Extend Last Segment** unless the hierarchy already contains a proper final tip.
5. Create the chain and test motion **before** adding collision.
6. Open Smart Collision Manager only if the part needs body or environment contact.
7. Add only the collision volumes/sources that actually affect the moving part.
8. Configure distance/offscreen culling before duplicating the character into crowds.
9. Validate in the Runtime Debugger.

Critical body-collision rule

After synchronization, inspect ReactiveBoneChain > Collision. If this reactive rig touches or intersects a body collider in its normal pose, add that collider to Excluded Reactive Colliders on this chain. A collider that starts embedded can push or jitter immediately. Turn off React To Connected Colliders only when the chain should ignore the entire connected set. These per-chain filters survive manager synchronization and runtime-group refreshes.