REACTIVE BONES V1 / STL DYNAMICS

Optimization and Runtime Debugging

IN THIS GUIDE

Reactive Bones V1 includes per-chain culling, sleeping, a shared world collision broadphase, and an adaptive crowd simulation budget.

The system automatically creates a `[Reactive Bone World]` at runtime when a chain first registers if no world already exists.

You may also place a **Reactive Bone World** component in the scene yourself when you want to tune global behavior explicitly.

---

1. Per-chain distance culling

The chain inspector exposes:

- **Use Distance Culling**
- **Reduced Update Distance**
- **Sleep Distance**
- **Reduced Update Rate**
- **Sleeping Update Rate**
- **Sleep Mode**
- **Sleep Blend Duration**
- **Wake Blend Duration**

Default chain distances are approximately:

- reduced update beyond 15 world units;
- sleep beyond 35 world units.

Adjust these to your camera scale and game type.

A close third-person game may want shorter distances. A wide RTS/simulation camera may want larger distances but lower reduced update rates.

---

2. Sleep modes

Blend To Animation

Smoothly returns the simulated pose toward animation while sleeping.

Recommended general-purpose mode.

Freeze Pose

Keeps the last solved pose while sleeping.

Reset Immediately

Immediately returns to animation.

Continue At Low Frequency

Keeps simulating at the configured sleeping update rate instead of fully freezing/resetting.

Use this when distant movement still matters visually but full-rate simulation is wasteful.

---

3. Offscreen optimization

Offscreen Mode can be used in addition to distance culling.

Modes:

- Ignore
- Reduced Rate
- Sleep

When offscreen behavior is enabled, assign one or more **Visibility Renderers** or leave the list empty so the chain can use renderers found under the component at initialization.

If no renderer can be found, the chain inspector reports an informational validation message.

---

4. Reactive Bone World culling reference

The world can use:

- an explicit **Culling Reference** Transform;
- an explicit **Culling Camera**;
- `Camera.main` when **Use Main Camera** is enabled.

If no reference exists, chains remain at full culling state rather than disappearing unexpectedly.

If your game replaces the main camera at runtime, either provide an explicit reference or refresh the world's main-camera reference from your own runtime flow.

---

5. Adaptive crowd performance

The Reactive Bone World can distribute expensive full simulation advances across rendered frames when measured workload exceeds a configured budget.

Important settings:

Use Adaptive Simulation Budget

Enables crowd time slicing.

Chain Simulation Budget Milliseconds

Default target: approximately **5 ms** of full chain simulation work per rendered frame.

Minimum Full Simulation Ticks Per Frame

Guarantees a minimum number of eligible chains receive a full advance even while the budget is active.

Maximum Crowd Step Delta Time

Limits how much simulation time one deferred tick is allowed to advance, reducing expensive catch-up spirals.

The system still captures animation and reapplies solved poses every rendered frame; the expensive simulation advances are what get distributed.

---

6. Collision performance

Author collision only after the chain/generated rig is working. Use the **Smart Collision Manager** as the single place to reduce collider count, narrow external layers, and synchronize connected chains.

Collision cost can exceed raw chain integration cost if every chain tests against every collider.

Use these levers first:

1. Keep connected analytic collider sets sparse.
2. Avoid finger/toe/full-body volumes when the reactive part never reaches them.
3. Narrow world Reactive Collider layer masks.
4. Narrow Unity Collider layer masks.
5. Generate fewer smart capsules instead of one collider per bone.
6. Split giant chains so distant pieces can cull independently.

The Smart Collision Manager warns when the connected set's chain-to-collider pair count becomes unusually large.

---

7. Shared Reactive Collider broadphase

The world uses a spatial broadphase for world-visible Reactive Bone Colliders.

Global controls include:

Reactive Collider Cell Size

Default: `0.5` world units.

Human-scale rigs generally perform well around `0.35-0.75` according to the inspector tooltip.

Maximum Cells Per Reactive Collider

Large volumes that would occupy too many cells use a bounded global fallback rather than flooding the grid.

Only change these settings after profiling a real scene.

---

8. Direct Unity Collider performance

Each chain can limit:

- Unity collider layers;
- trigger interaction;
- candidate capacity;
- self-collision filtering.

The chain inspector exposes runtime diagnostics including:

- external Reactive Collider count;
- peak Unity candidate count;
- Unity candidate buffer saturation.

If saturation occurs, first narrow the layer mask or simplify nearby collision. Increase **Candidates Per Particle** only when the scene genuinely requires it.

---

9. Runtime Debugger

Open:

`Tools > STL Dynamics > Reactive Bones > Runtime Debugger`

The debugger only becomes active in Play Mode.

Global statistics include:

- registered chains;
- active chains;
- sleeping chains;
- culled chains;
- particles;
- colliders;
- contacts;
- update rate;
- last solver-frame cost;
- rolling average simulation cost.

Each chain shows:

- current culling/sleep state;
- particle count;
- collider count;
- contact count;
- effective weight;
- average simulation cost.

Runtime controls:

- **Reset**
- **Enable / Disable**
- **Wake / Force Sleep**
- **Basic Gizmos / All Gizmos**

---

10. Debug gizmos

Per-chain debug drawing can display:

- bones;
- particles;
- collision radii;
- animated targets;
- velocity;
- contacts;
- virtual endpoints;
- bounds.

Use **All Gizmos** briefly for diagnosis. Keep only the views you need during normal authoring to avoid visual noise.

Recommended optimization pass

For a finished character prefab:

1. Confirm motion with no unnecessary contacts/debugging enabled.
2. Keep **Generate Contacts** off unless gameplay or visualization consumes contact data.
3. Enable distance culling.
4. Add offscreen sleeping/reduced rate when appropriate.
5. Reduce analytic collider count.
6. Narrow collision layers.
7. Test multiple characters in the real target scene.
8. Use the Runtime Debugger to find the heaviest chains.
9. Split oversized components or reduce generated bone count where the visual difference is minor.
10. Only then adjust the global adaptive budget/broadphase settings.

> **Fix rest-pose overlap before profiling**
>
> Do not spend iterations on a collider the rig already intersects at rest. Add that volume to the chain's Excluded Reactive Colliders; it is more stable and avoids useless collision work.