STL DYNAMICS

User Guide

Retro Water - User Guide

Complete authoring, rendering, physics, ocean, effects, profiles, and production workflow reference.

System architecture

Retro Water is split into six cooperating layers.

1. **Profile** - reusable authoring data.
2. **Surface** - binds a profile/material to a planar water body and reproduces the shader wave equation on the CPU.
3. **Ocean** - generates bounded tiled water meshes and updates mesh/wave LOD around a viewer.
4. **Buoyancy** - samples the surface and applies distributed Rigidbody forces.
5. **Renderer feature** - draws projected caustics and the underwater full-screen pass.
6. **Shaders** - render the surface, depth, shadows, caustics, and underwater image treatment.

Profile application flow

```
RetroWaterProfile
        |
        v
RetroWaterSurface.ApplyProfile()
        |
        +--> material instance / baked material / manual material
        +--> surface sampling settings
        +--> caustics settings
        +--> underwater settings
        |
        +--> RetroWaterOcean.ApplyProfileSettings() when Apply To Ocean is enabled
```

SOURCE

- `RetroWaterProfile.cs`
- `RetroWaterSurface.Profile.cs`
- `RetroWaterOcean.Profile.cs`

Profiles and material workflows

Create a profile from:

```
Assets > Create > STL Dynamics > Retro Water > Profile
```

The custom profile inspector has four tabs.

Overview

Contains:

- Display name and description.
- Optional Base Material.
- Optional Baked Material.
- Shader readiness.
- Enabled-system summary.
- Material creation and profile application actions.

Surface

Controls the water material and CPU sampling.

Surface appearance

- Surface texture.
- Shallow color.
- Deep color.
- Surface mode.
- Opacity or cutout threshold.

Texture mapping

`Texture Space` is independent from object scale.

- **Object Anchored** - mapping follows object position and rotation but ignores Transform scale.
- **World Locked** - mapping stays fixed in world space and can align across separate water meshes.

`Texture Tiling` is measured by the shader's independent surface mapping. Stretching a water object does not have to stretch the texture.

Motion

- Flow direction and speed.
- Wave scale and height.
- Quantized wave-height bands.
- Smooth or stepped animation frame rate.

`Motion Frame Rate = 0` keeps time smooth. Positive values quantize animation time.

Geometry

- Vertex snap space: Off, Object, World, or View.
- Vertex snap density.
- Faceted normal blend.

Ocean profiles normally force world snapping to keep generated chunks consistent.

Texture rendering

- Texture Pixel Resolution.
- Affine warp.
- Repeat or Clamp.

`Texture Pixel Resolution <= 0` samples at full authored resolution.

Palette rendering

- Palette levels and bias.

- Bayer dither amount and pixel size.
- Lighting bands.
- Opacity bands.

Lighting

- Minimum light.
- Received shadow strength.
- Highlight mode.
- Highlight strength and sharpness.
- Retro glint breakup.
- Rim strength and power.

Highlight modes:

Mode	Result
Off	No directional specular highlight.
Retro Glints	Breaks the broad camera-dependent highlight with stable surface texture detail.
Classic Specular	Preserves the smooth view-dependent lobe.

Use **Off** or **Retro Glints** when a large white circular highlight follows the camera.

Depth and shoreline

- Depth effects toggle.
- Deep color distance.
- Foam color, distance, softness, and strength.

These read camera depth and require URP Depth Texture.

CPU surface mapping and sampling

The surface profile also contains:

- Plane axes.
- Finite-surface bounds.
- Bounds padding.
- Wave UV scale/offset.
- Local height offset.
- Normal sample distance.
- Flow velocity multiplier.
- Per-frame material refresh.

`RetroWaterSurface.TrySample()` reproduces the shader's stepped value-noise wave equation. For ordinary meshes, the closest visual/physics match assumes a planar mesh with UVs that map linearly across its local bounds. Ocean mode supplies world-space wave coordinates instead.

Effects

Caustics

The caustics renderer:

- Reconstructs opaque scene positions from depth.
- Restricts projection to the water surface or ocean bounds.
- Reuses the surface wave parameters.
- Supports stepped animation and intensity banding.

- Modulates against the main light and main-light shadows.
- Fades near the surface and with depth.

Important controls:

- **Texture** - required unless a default Resources texture is packaged.
- **Scale** - world-space size of one caustics tile.
- **Maximum Depth** - caustics stop beyond this water depth.
- **Surface Fade** - avoids hard contact at the waterline.
- **Depth Falloff** - controls how quickly caustics disappear.
- **Priority** - higher values win when more surfaces are active than the renderer feature limit.

The renderer feature default limit is four caustics surfaces per camera. Surfaces are sorted by priority, then camera distance.

Underwater camera

The underwater state is position-based, not volume-based.

Each camera is tested against the animated water sample at its world position. The system:

- Respects finite surface bounds.
- Follows wave height.
- Uses the stable water-plane up axis rather than the changing wave normal.
- Selects the highest-priority overlapping surface.
- Uses the shallower camera depth as the tie-breaker for equal priorities.

Controls include:

- Transition depth and surface offset.
- Fog density and clear distance.
- Maximum visibility.
- Depth tint distance.
- Brightness and desaturation.
- Screen distortion.
- Chromatic aberration.
- Vignette.
- Animation frame rate.
- Color bands and dithering.
- Priority.

Ocean

Set `Apply To Ocean` when a profile should configure `RetroWaterOcean`.

If disabled, the profile remains valid for normal surfaces and the ocean keeps its local grid/LOD values.

Surface component

`RetroWaterSurface` is the central scene component.

Source references

- Runtime and sampling: `RetroWaterSurface.cs`
- Profile integration: `RetroWaterSurface.Profile.cs`
- Inspector: `Editor/RetroWaterSurfaceEditor.cs`

Main responsibilities

- Registers active surfaces.
- Resolves Renderer, MeshFilter, and optional Ocean.
- Applies and refreshes profiles.
- Assigns active materials.
- Computes finite bounds and water-plane axes.
- Evaluates CPU wave height, normal, position, and velocity.
- Supplies caustics property data.
- Selects the active underwater surface.
- Provides nearest-surface lookup for buoyancy.

Profile actions

The component inspector exposes:

- Apply Profile.
- Open Profile.
- Create / Update Material.
- Capture Current Into Profile.
- Create Profile From This Surface.

When creating a profile from a tuned surface, material, surface, caustics, underwater, and optional ocean values can be captured into one asset.

Material modes in detail

Runtime Instance

`RetroWaterSurface` calls `RetroWaterProfile.CreateMaterialInstance()` and stores the result as a hidden, non-saved material. It is private to the surface and released when no longer needed.

Use it when:

- Each water body may switch profiles independently.
- You do not need a `.mat` asset.
- You want the simplest default workflow.

Baked Profile Material

Uses `RetroWaterProfile.BakedMaterial`.

Use it when:

- Several surfaces should share the same material.
- You want to inspect or reference a normal `.mat`.
- You want predictable shared material batching behavior.

If the profile has no baked material, the surface creates a temporary runtime material instead.

Manual Material

The surface resolves the explicitly assigned manual material or an existing renderer/ocean material.

Use it when:

- A technical artist wants direct material control.
- Shader values should not be overwritten by the profile.
- The profile should still drive sampling, caustics, underwater, and ocean settings.

Ocean system

Source references

- Ocean generation and LOD: `RetroWaterOcean.cs`
- Profile bridge: `RetroWaterOcean.Profile.cs`
- Inspector: `Editor/RetroWaterOceanEditor.cs`

Unit-scale root

Keep the ocean root at `1,1,1`.

Generated chunks are children scaled to their requested local chunk dimensions. Scaling the root multiplies those dimensions, reduces effective vertex density, and makes waves appear broad or flat.

Use **Ocean Size** to change the ocean dimensions, not Transform scale.

Grid generation

Requested chunk counts use ceiling division per axis.

If the requested count exceeds `Maximum Chunk Count`, `CalculateEffectiveChunkSize()` increases the actual chunk size until the entire bounded ocean fits within the limit.

The edge chunks are shortened when the ocean size is not evenly divisible by the effective chunk size.

Generated chunk objects:

- Are children of a generated container.
- Use `HideFlags.HideAndDontSave`.
- Are reconstructed by the component.
- Receive shared LOD meshes and per-renderer property blocks.

Mesh LOD

`Near Grid Resolution` controls the closest mesh.

Each following LOD reduces interior resolution. LOD selection uses distance from the viewer to each chunk renderer's bounds.

Thresholds begin at `First LOD Distance` and multiply by `LOD Distance Multiplier`.

LOD updates occur after the viewer moves at least `LOD Update Distance`.

`Maximum Render Distance = 0` disables the explicit cutoff and leaves normal frustum culling in charge.

Seam handling

Use:

```
Crack-Free Shared Borders: Enabled
Legacy Seam Skirts: Disabled
```

Crack-free meshes preserve a dense shared perimeter while reducing the interior. Neighboring chunks therefore retain matching border vertices across different LODs.

Legacy vertical skirts are a compatibility fallback. On transparent water they may appear as dark vertical grid lines.

After changing seam or mesh settings, press **Build / Rebuild Ocean**.

Wave LOD

Mesh LOD and wave LOD are separate.

- `Full Wave Detail Distance` - full displacement detail near the viewer.
- `Wave Fade End Distance` - point where far settings are fully reached.
- `Far Wave Height` - remaining displacement amplitude.
- `Far Wave Detail` - remaining high-frequency detail.

The viewer position is supplied to the shader through a per-chunk material property block.

Texture and wave continuity

Ocean mode supplies:

- World-space ocean origin.
- World-space U and V axes.
- World-unit wave UV scale and offset.
- Viewer position and wave LOD values.

Enable:

- **Force World-Locked Surface Texture**
- **Force World Vertex Snap**

to keep mapping and snap density consistent across chunks.

Viewer selection

The ocean uses:

1. Explicit `Viewer`, when assigned.
2. `Camera.main`, when automatic lookup is enabled.
3. No LOD updates, if neither is available.

Assign the actual gameplay camera when possible.

Buoyancy

Source references

- Runtime: `RetroWaterBuoyant.cs`
- Inspector: `Editor/RetroWaterBuoyantEditor.cs`

Model

Automatic buoyancy fills collider bounds with body-local volume samples. For each sample:

1. The nearest/assigned water surface is sampled.
2. Immersion is measured along gravity.
3. A smooth submerged fraction is calculated.
4. Buoyant force, vertical damping, and horizontal drag are applied at the sample position.
5. Distributed force positions naturally create pitch and roll torque.

This avoids the invisible flat-platform behavior produced by gravity-aligned underside detectors.

Point modes

Mode	Use
Center Of Mass	Very simple floating objects or debugging. Produces no distributed buoyancy torque.
Automatic Collider Bounds	Recommended default. Builds a 3D sample grid and filters points against enabled colliders.
Local Offsets	Hand-authored local positions.
Transforms	Scene-authored point transforms.

`Point Orientation` applies to Local Offsets and Transforms. Automatic collider-volume samples always rotate with the Rigidbody while immersion remains gravity-aligned.

Float response

- **Buoyancy Strength** controls maximum lift relative to body weight.
- **Sink Depth** moves the effective buoyancy waterline.
- **Maximum Submersion Depth** is the vertical depth represented by one sample.
- **Response Exponent** shapes how lift grows with immersion.
- **Surface Normal Influence** leans force toward the sampled wave normal.

Water motion

- Wave Velocity Influence.
- Current Influence.
- Vertical Damping.
- Linear Water Drag.
- Angular Water Drag.
- Hydrodynamic and absolute force caps.

The surface's `Flow Velocity Multiplier` must be non-zero if the visual flow direction should become a physical horizontal current.

Self-righting

The self-righting system applies a damped angular acceleration toward the authored local up axis.

Use lower strength for free-rolling debris and higher strength for crates, rafts, or boats that should recover from inversion.

Do not freeze Rigidbody rotation axes that the water system needs to correct.

Practical physics setup

- Rigidbody must be non-kinematic.
- At least one enabled, non-trigger collider is recommended.
- Keep built-in Rigidbody drag modest.
- Use Interpolate for smoother visual motion.
- Use Continuous collision detection for small or fast props when needed.
- Test in water deeper than the object's required travel.
- In multiplayer, run Rigidbody authority on one peer and replicate the result.

Caustics and underwater renderer feature

Source: `RetroWaterCausticsFeature.cs`

Default pass order:

Pass	Default injection
Caustics	`AfterRenderingOpaques`
Underwater	`BeforeRenderingPostProcessing`

Feature settings:

- Caustics injection point.
- Underwater injection point.
- Maximum caustics surfaces per camera.
- Render in Scene View.
- Render in Preview cameras.
- Render in Reflection cameras.

The feature requests depth input for both passes.

Manual material workflow

Create a material and select:

```
STL Dynamics/Retro Water
```

The custom ShaderGUI is implemented by `Editor/RetroWaterShaderGUI.cs`.

It exposes the same major groups as the profile inspector and includes:

```
Create Profile From Material
```

Use `RetroWaterMaterialUtility.ApplySurfaceMode()` when editor code changes surface mode programmatically and needs to synchronize blend state, render queue, tags, and keywords.

Editor tools

Setup and diagnostics

```
Tools > STL Dynamics > Retro Water > Setup & Diagnostics
```

Renderer setup

```
Tools > STL Dynamics > Retro Water > Setup > Install Renderer Feature
Tools > STL Dynamics > Retro Water > Setup > Enable URP Depth Texture
```

Profile creation

```
Assets > Create > STL Dynamics > Retro Water > Profile
Tools > STL Dynamics > Retro Water > Create Profile From Selected Material
Tools > STL Dynamics > Retro Water > Create Profile From Selected Surface
Tools > STL Dynamics > Retro Water > Create Starter Profiles
```

Scene objects

```
GameObject > STL Dynamics > Retro Water > Water Surface
GameObject > STL Dynamics > Retro Water > Ocean
```

Performance guidance

Surface

- Reduce mesh density when vertex displacement does not need fine detail.
- Lower additional-light counts in URP when many lights affect water.
- Disable depth effects when not used.
- Set Highlight Mode to Off when highlights are not needed.
- Lower Texture Pixel Resolution for a deliberate retro look; use `0` for full-resolution sampling.

Ocean

- Increase Chunk Size to reduce Renderer count.
- Lower Near Grid Resolution.
- Reduce LOD Count when distant layers are unnecessary.
- Use Maximum Render Distance for bounded scenes.
- Fade far wave height/detail.
- Keep shared borders enabled.
- Disable shadow casting unless required.

Caustics and underwater

- Limit active caustics surfaces per camera.
- Reduce caustics intensity/depth rather than projecting through the whole world.
- Disable Scene View rendering when editor performance matters.
- Disable underwater distortion/chromatic aberration individually when not required.

Buoyancy

Automatic sample count is approximately:

```
Samples X x Samples Y x Samples Z
```

A `2x2x2` grid is eight candidate samples and is a good general default. Increase only for large or irregular objects.