**Optimizor 1.0 — Action Catalogue (UE 5.6) 13/08/25**

A **chaptered, concrete checklist** of everything the plugin can do to a project, split by domain. Each action is labeled by **Mode**:

* **AUTO** — performed automatically by the plugin (with Dry‑Run and caps)
* **SEMI** — offered with a one‑click wizard/confirm (bulk but user‑approved)
* **ADVISE** — best‑practice surfaced in UI/report (not changed by code)

All AUTO actions respect: Dry‑Run, Max‑Changes, Selection vs Paths, Exclusions, Skip‑Collection, close editors, source‑control checkout, verify‑after‑write, and history snapshot.

**1) Textures — Concrete Actions**

**1.1 Classification & Inference**

* **AUTO** Detect texture role via name/path tags (normal/base/masks/grayscale/UI/LUT/packed/UDIM).
* **AUTO** Infer usage from referencing materials (up to N referencers) and map to sampler buckets (normal/masks/grayscale/linearcolor/color).
* **ADVISE** Flag ambiguous/contradictory usage (same texture sampled as color & linear in different materials).

**1.2 Format & Color Pipeline**

* **AUTO** Set *CompressionSettings* per class: TC\_Normalmap for normals, TC\_Masks for packed masks, TC\_Grayscale for scalar maps, TC\_Default (or TC\_BC7 when preset allows) for color.
* **AUTO** Set sRGB on/off appropriately (off for normals/masks/linear, on for color/UI).
* **SEMI** Project‑level remap table (e.g., force BC7 for hero albedos on PC Ultra).
* **ADVISE** Report alpha‑channel waste (normals with unused alpha, color with solid alpha).

**1.3 Mipmaps & LOD**

* **AUTO** Toggle GenerateMips and set MipGenSettings (FromGroup / SharpenX / BlurX / NoMipmaps).
* **AUTO** Enforce minimum LODBias (e.g., ≥ 1) except UI/LUTs/small detail maps.
* **SEMI** Clamp MaxTextureSize per role (color/normal/mask caps); offer resample pass for oversize assets.
* **ADVISE** Highlight textures lacking mips that are sampled minified (aliasing risk).

**1.4 Streaming & VT**

* **AUTO** Set NeverStream=true for UI/LUTs; ensure world content streams.
* **AUTO** Respect landscapes/UDIMs: don’t disable VT there; otherwise allow **force‑disable VT** when preset requires.
* **SEMI** Adjust Texture Groups to preset (UI, World, LUT); optional per‑group max size overrides.
* **ADVISE** Recommend Virtual Texturing for giant tileables on high‑end targets.

**1.5 Safety, Exceptions & Scope**

* **AUTO** Skip assets by path/name hints and **Skip‑Collection**.
* **AUTO** Close editors before mutation; attempt source‑control checkout; verify after write.
* **SEMI** One‑click *Revert Last Run* using snapshot JSON.

**1.6 Reporting & Budgeting**

* **AUTO** Report before/after settings, top change reasons, CSV of changes, memory bytes estimate, and create Collections: Optimizor\_Changed\_\*, Optimizor\_Skipped\_\*.
* **ADVISE** Texel density and atlas suggestions (when we can’t change source art).

**2) Static Meshes — Concrete Actions**

**2.1 Nanite & Build Settings**

* **AUTO** Enable/disable **Nanite** per preset; set fallback screen size.
* **SEMI** Batch *Build Nanite* for eligible meshes; skip skeletal/unsupported.
* **ADVISE** Suggest Nanite for CAD/high‑poly imports.

**2.2 LOD Strategy**

* **AUTO** Generate LODs with IMeshReduction where missing; set LOD count from tri budget.
* **AUTO** Apply screensize curve per preset; optionally enforce triangle caps for L1/L2/L3.
* **SEMI** Respect user LODGroups if specified; otherwise apply preset caps.
* **ADVISE** Warn on excessive materials per mesh (draw‑call multiplier).

**2.3 Lightmap UVs & Tangents**

* **AUTO** Generate/repair **Lightmap UVs** (min/max res, channel selection).
* **AUTO** Recompute normals/tangents per project policy (angle threshold).
* **ADVISE** Flag overlapping lightmaps, extreme texel densities.

**2.4 Collision & Complexity**

* **AUTO** Set simple collision type (box/sphere/convex) for props missing collision.
* **SEMI** Replace complex‑as‑simple with simple for non‑interactive assets.
* **ADVISE** List high‑poly colliders; promote UCX authoring tips.

**2.5 Stats & Collections**

* **AUTO** CSV per mesh: base tris, per‑LOD tris, materials, Nanite on/off, lightmap res.
* **SEMI** Create *Changed/Needs‑Attention* Collections (e.g., “No LODs”, “>X materials”).

**3) Skeletal Meshes & Animation — Concrete Actions**

* **AUTO** Apply **Animation Compression** settings per preset (bitwise/BVH/ACL if available).
* **AUTO** Enable **Update Rate Optimization** on components for distant actors.
* **SEMI** Strip/add LODs on skeletal meshes (bone reduction where rules allow).
* **ADVISE** Cap bones/influences for crowds; share anim instances; reduce tick rate off‑screen.

**4) Materials — Concrete Actions**

* **AUTO** Detect & remove dead **static switches** (compile‑time only) in dry‑run preview → confirm → apply.
* **AUTO** Identify duplicate texture samplers with identical params; suggest consolidation (SEMI to auto‑replace).
* **SEMI** Normalize normal map space naming (\_ogl/\_dx) → add note or switch sampler type where safe.
* **ADVISE** Flag high instruction count, heavy translucency, per‑pixel world‑position ops.

**5) Runtime (Scalability, Post, Lighting) — Concrete Actions**

**5.1 Scalability & Device Profiles**

* **AUTO** Apply per‑preset **Scalability** buckets (View/Shadow/Effects/Post/Foliage) via .ini.
* **SEMI** Switch **Device Profiles** (Desktop/Console/Mobile/VR) and copy tuned overrides.

**5.2 Global Illumination & Reflections**

* **SEMI** Toggle **Lumen** GI/Reflections on/off; choose HWRT vs Software; set quality levels.
* **ADVISE** Fall back to baked/static for mobile/archviz presets when appropriate.

**5.3 Shadows**

* **AUTO** Enable/disable **Virtual Shadow Maps**; adjust distances/cascades; set per‑quality defaults.
* **ADVISE** Reduce shadowed lights count; trim attenuation radii; prefer static for scenery.

**5.4 Post‑Processing & AA**

* **AUTO** Configure AA (TSR/TemporalAA/MSAA for VR) per preset; dial bloom/AO/DOF/motion blur.
* **SEMI** Enable Dynamic Resolution with bounds for consoles/PC Balanced.

**5.5 Streaming & Async Loading**

* **AUTO** Set r.Streaming.PoolSize hints; configure time‑slice for async loading.
* **SEMI** Enable **Fast Geometry Streaming** plugin and defaults for large worlds.
* **ADVISE** Organize streaming levels; avoid heavy blocking loads.

**6) World Partition, HLOD & Culling — Concrete Actions**

* **SEMI** Enable **World Partition** on new projects; set cell size templates (open world vs interior).
* **AUTO** Generate **HLOD** clusters with preset reduction; build proxies.
* **AUTO** Inject default **Cull Distance Volume** rules for common small props.
* **ADVISE** Place streaming volumes/triggers; cap on‑screen actor counts.

**7) Audio — Concrete Actions**

* **AUTO** Apply platform‑appropriate **compression** settings to SoundWaves (where unset).
* **SEMI** Set concurrency/priority limits per preset (mobile lower caps).
* **ADVISE** Stream long music; reduce simultaneous ambient loops.

**8) Packaging, Cooking & Size — Concrete Actions**

* **SEMI** Toggle PAK **compression on/off** (boot‑time vs disk size trade).
* **AUTO** Exclude Editor‑only content from cook; strip debug content by pattern.
* **SEMI** Set platform target texture formats (ASTC/ETC/BCx) and shader pipeline caches.
* **ADVISE** Preload hot levels; warm shaders; keep startup maps minimal.

**9) Safety & Governance — Concrete Actions**

* **AUTO** Dry‑Run default; enforce **Max‑Changes**; per‑change gate via *Apply‑Only* keys.
* **AUTO** Source‑control checkout; close editors; verify‑after‑write; save dirty packages.
* **SEMI** One‑click **Revert** using last snapshot; export/import action plan JSON.
* **ADVISE** Educate on content guidelines (naming, foldering, tags for rules).

**10) Reporting & Telemetry — Concrete Actions**

* **AUTO** Write TXT/CSV/JSON reports with counts, paths, reasons, memory deltas; emit compact summary to log.
* **AUTO** Create Collections for Changed/Skipped/Hot Referencers; open report folder.
* **SEMI** Baseline capture: run stat unit/gpu & MemReport and store next to audit.
* **ADVISE** Budget dashboards per map (future module).

**11) Presets — Concrete Actions**

* **AUTO** Load preset .json; merge project overrides; display deltas in UI.
* **AUTO** Apply preset to Texture/Mesh/Material/Runtime/World rules in one pass.
* **SEMI** Duplicate + edit preset into project config; lock presets for CI.
* **ADVISE** Per‑preset best‑practice tips inline (what engine can’t change).

**12) Extensibility Hooks**

* **SEMI** Per‑team rule packs (regex/tag → action) for custom pipelines.
* **SEMI** CLI/headless mode for CI (cook guards, budget checks).
* **ADVISE** Plugin API for third‑party reducers (Simplygon) and analytics.

**13) Coverage Map (AUTO vs SEMI vs ADVISE)**

| **Domain** | **AUTO** | **SEMI** | **ADVISE** |
| --- | --- | --- | --- |
| Textures | Compression, sRGB, mips, LODBias, VT off/on (safe cases), groups, reporting | Max size resample, group remap | Texel density, atlas, authoring |
| Static Mesh | Nanite toggle, LOD gen, screen sizes, lightmap UVs, simple collision, stats | Triangle caps, respect LODGroup, complex→simple collision | Materials per mesh, CAD prep |
| Skeletal | Anim compression, Update Rate Opt | LOD/bone reduction | Crowd budgets, shared BP |
| Materials | Dead static switches removal, sampler duplicates report | Auto‑replace duplicate samplers | Shader complexity guidance |
| Runtime | Scalability ini, VSM, AA/post defaults, streaming pool/time slice | Lumen mode, device profile, Dynamic Res, Fast Geo Streaming | Lighting strategy per platform |
| World | HLOD gen, cull distance defaults | Enable WP, cell sizes | Streaming layout, actor budgets |
| Audio | Compression defaults | Concurrency caps | Mix/atten rules |
| Packaging | Cook strip editor data | Pak compression, formats | Startup flow, shader warmup |
| Safety & Reports | Dry‑run, caps, verify, snapshots, collections | Revert, baseline capture | Governance tips |

**14) Next Step — Implementation Drill‑Down Index**

For each chapter above, we will create two sub‑documents:

1. **User/Plugin Controls** — exact UI toggles, preset bindings, safe defaults.
2. **Code Path** — module, APIs, property writes, .ini keys, Python/C++ flows, failure modes.

This Action Catalogue is the canonical scope; we’ll implement and verify chapter by chapter.