



THE UNREAL RADAR · ENGINE RADAR

# The Migration & Deprecation Radar

---

The mirror image of the feature radar: Epic's older systems and APIs that are being superseded — what to migrate off, and roughly when, judged against the current UE 5.7 release.

Engine Radar · Edition 2026.1

**2026-06-13**

Edited by Phil, MythicLemon

[mythicmon.com/radar](https://mythicmon.com/radar)

## Editor's note

---

Our companion **Feature Radar** answers "what should I build on?" This one answers the harder, less glamorous question: **what should I build off, and by when?** Every engine accumulates older systems that still work but are being quietly left behind — and the cost of ignoring them is rarely a crash today. It's a slow tax: a project that compiles on 5.7 but leans on a 2020-era assumption, then breaks on the upgrade that finally retires it.

We rate **Epic's own systems** here, the same as the Feature Radar — there is nothing of anyone else's to mark down, and nothing to sell you. The entries are Epic's APIs, subsystems and workflow assumptions, never a third-party plugin or a marketplace listing.

A word on the rings, because the distinction is the whole point. **Migrate now** means the old path is \*gone or going\* — Epic has removed it or set a clear sunset, and new work on it is wasted work.

**Migrate soon** means it's officially superseded and discouraged; it still runs, but it's the past tense of the engine and you should be planning the move this year. **Plan the move** means the replacement isn't fully ready yet (often experimental), so you can't migrate today — but you should understand the destination and avoid digging the old hole deeper. **Still fine** means: yes, it's the older way, but Epic still ships and supports it, and there's no urgency — don't let a migration zealot waste your sprint.

We have been deliberately conservative about the difference between **removed** and merely **discouraged** — they get treated as the same thing far too often, and the panic costs teams real time. Where we say something is gone, it's gone (PhysX); where it's only out of fashion but fully supported (Sound Cues), we say so plainly. Because this is the first edition, nothing carries a movement arrow yet. The matrix at the foot of the page lays out, blip by blip, exactly \*why\* each one sits where it does.

One commercial note that frames the whole radar. By the **MythicLemon Marketplace Index, our own compiled dataset** (as of 2026-05), **46.7%** of the version-tagged Fab catalogue is still UE4 — a genuine modernisation backlog sitting in published, for-sale content. Migration isn't an abstract engine-team concern; it's the difference between an asset that still installs cleanly in a buyer's 5.7 project and one that throws a wall of conversion warnings. Several blips below are exactly the assumptions baked into that backlog.

– Phil, MythicLemon

**This is our view, not a measurement.** The rings are a considered production-readiness opinion — not a quality score. We rate Epic's engine, asset categories and practices, never a named third-party product. Where a category overlaps something we sell, we say so.

# The radar at a glance

● Rendering & Lighting ● Worldbuilding & Streaming ● Gameplay & Input ● Animation, Audio & VFX

**Migrate now** Gone or going. Epic has removed it or set a clear sunset — new work on it is wasted work.

- 1 PhysX / APEX physics
- 12 Building on a single old engine version

**Migrate soon** Officially superseded and discouraged. It still runs, but it's the past tense of the engine — plan the move this year.

- 4 Baked lightmaps as a default
- 6 Persistent-level / sublevel streaming
- 3 Legacy input mappings (Action/Axis)
- 2 Cascade (legacy VFX)
- 10 UE4 SkeletalMesh / material authoring assumptions

**Plan the move** The replacement isn't fully ready (often experimental). You can't migrate today, but understand the destination and don't dig the old hole deeper.

- 7 Legacy foliage instancing
- 8 Heightfield Landscape as the only terrain option

**Still fine** The older way, but Epic still ships and supports it. No urgency — don't let a migration zealot waste your sprint.

- 5 Forward / mobile-style rendering as a desktop default
- 9 Sound Cues as the default audio path
- 11 Niagara module/script assumptions

## Removed vs merely discouraged — why each blip sits where it does

System / assumption	Actual engine status	We say	Why
PhysX / APEX	Removed (replaced by Chaos)	<b>Migrate now</b>	Not optional — the old physics is gone from the engine you build on today.
Single old engine version	N/A (a project decision)	<b>Migrate now</b>	The meta-debt under every other blip; adopt an upgrade cadence now, not a heroic jump later.
Cascade VFX	Legacy, superseded by Niagara	<b>Migrate soon</b>	Supported for now, but no investment and the clear past tense — never author new effects on it.
Legacy input (Action/Axis)	Superseded by Enhanced Input	<b>Migrate soon</b>	Still runs; Enhanced Input is the default and the migration is well-trodden.
Sublevel streaming	Superseded by World Partition	<b>Migrate soon</b>	Old maps still load; World Partition is the expected large-world model and source-control win.
Baked lightmaps as *default*	Supported technique; Lumen is the default	<b>Migrate soon</b>	The reflex, not the tool, is dated — start from Lumen, bake deliberately.
UE4 content assumptions	Converts, with warnings	<b>Migrate soon</b>	46.7% of the tagged catalogue is still UE4; conversion warnings are the cost of not re-authoring.
Legacy foliage	Supported; Nanite Foliage is experimental	<b>Plan the move</b>	No mature destination yet — understand the path, don't rebuild on experimental code.
Heightfield-only terrain	Supported; Mesh Terrain is 5.8-preview	<b>Plan the move</b>	Nothing production-ready to move to — keep the architecture open, watch 5.8.
Sound Cues	Supported, not deprecated	<b>Still fine</b>	MetaSounds for new work, but no urgency — don't rip out working cues.
Forward rendering	Supported platform choice	<b>Still fine</b>	Not deprecated at all — the mistake is migrating *off* it where it's the right call.
Older Niagara patterns	Modern system, evolving internals	<b>Still fine</b>	You're already on the right system — this is version-bump hygiene, not migration.

The whole point of this radar: 'removed' and 'discouraged' are not the same thing. Status is judged against UE 5.7 (current stable); 5.8 is in preview. "We say" is our migration-urgency opinion, not a measurement.

## The evidence (live online)

AS OF MAY 2026

**New supply has moved to UE5; the back-catalogue is the migration backlog.** The backlog this radar keeps returning to: by the MythicLemon Marketplace Index (our own compiled dataset, as of 2026-05), UE5 is the overwhelming majority of *\*new\** releases, yet UE4 still accounts for 46.7% of the version-tagged catalogue. The gap between the two is exactly the migration debt the blips above describe.

↗ Live, interactive figure (with downloadable data): <https://www.mythiclemon.com/data/dashboards/engine-migration>

# The verdicts

## ■ Rendering & Lighting

The drawing and lighting paths. The corner with the most genuinely-removed code (PhysX-era assumptions, legacy lighting) and the most upgrade-time surprises.

### 1 PhysX / APEX physics MIGRATE NOW

*Already removed. Chaos is the engine's physics — there is nothing left to migrate to.*

This one isn't a recommendation, it's a fact: **PhysX and APEX were replaced by Chaos and are gone** from modern Unreal Engine. There is no compatibility toggle to flip back, no setting to keep the old simulation alive. If a codebase still carries PhysX-era assumptions — direct PhysX API calls, APEX cloth/destruction assets, `bUseChaos`-style branches` — they are dead paths on 5.7.

We place it firmly in **Migrate now** because the migration is not optional and not deferrable: the engine you build on today already made the decision for you. In practice the work is mechanical — re-author cloth and destruction on Chaos equivalents, and delete the dead branches. For most projects this was done releases ago; if it wasn't, it's the first thing to clear, because nothing else compiles around it.

Sources: [Epic · UE 5.7 release notes](#)

### 4 Baked lightmaps as a default MIGRATE SOON

*Fully-static, lightmap-only lighting as your \*default\* — superseded for most new work by Lumen, still valid where you need it.*

This blip is about a *\*default assumption\**, not a removed feature. Static lightmapping still exists and is supported, and it remains the correct choice for fixed-hardware, performance-critical scenes where you can afford no dynamic GI cost. What's changed is that **Lumen** is now the production-ready dynamic-lighting default in UE5, and reaching reflexively for a fully-baked pipeline on a new project is increasingly the legacy instinct.

We say **Migrate soon** at the level of the default, not the technique. If you start every new scene by building lightmaps out of habit, that habit is dated — start from Lumen and fall back to baking deliberately where the budget demands it. Note that the old CPU lightmass path has long since given way to the GPU Lightmass for teams that do bake. Don't tear baked lighting out of a shipping project that relies on it; do stop treating it as the automatic starting point.

Sources: [Epic · UE 5.7 release notes](#)

## 5 Forward / mobile-style rendering as a desktop default STILL FINE

*The forward renderer is a deliberate platform choice, not a deprecated path — keep it where it belongs.*

We include this one specifically to *stop* a migration mistake. The deferred renderer is the default and the home of Nanite, Lumen and Virtual Shadow Maps; the **forward renderer is not deprecated** — it's the right, supported choice for VR and many mobile and performance-bound targets where the deferred path's costs don't fit. Treating 'forward' as a thing to migrate away from is a category error.

So we place it in **Still fine**, with a caveat that runs the opposite direction to the rest of this radar. The legacy instinct to watch for isn't *using* forward rendering — it's *assuming* it for a desktop or current-gen-console project that would be better served by the deferred, Nanite-and-Lumen pipeline. Choose your renderer by your platform and budget, not by habit, and don't migrate off forward just because a checklist told you it was old.

Sources: [Epic · UE 5.7 release notes](#)

## ■ Worldbuilding & Streaming

Level structure, terrain and foliage. Mostly 'plan the move' territory — the replacements (World Partition, Nanite Foliage) are at very different stages of readiness.

## 6 Persistent-level / sublevel streaming MIGRATE SOON

*The old sublevel workflow for large worlds — superseded by World Partition, still loadable.*

Before World Partition, large UE worlds were assembled from a persistent level plus streaming sublevels, hand-managed by streaming volumes and level blueprints. **World Partition** — with Data Layers, One File Per Actor and automatic grid streaming — is now the production-ready, expected approach, and Epic's own large-world tooling is built around it. The old sublevel maps still load and run, so existing projects aren't broken.

We say **Migrate soon** for new large worlds and most actively-developed ones. The win isn't just streaming behaviour; it's the source-control and collaboration model — One File Per Actor removes the single-level merge-conflict bottleneck that made big-team level work miserable. The migration has a genuine learning curve and changes how the team works day to day, so it's a planned project, not a flip — but the destination is settled and the old workflow is the past tense.

Sources: [Epic · UE 5.7 release notes](#)

## 7 Legacy foliage instancing PLAN THE MOVE Epic: Experimental

*Today's hierarchical instanced foliage is fine; the Nanite-density future is only experimental — so plan, don't migrate yet.*

Vegetation has always been Nanite's hardest case, and 5.7 added an **experimental Nanite Foliage** path plus a Procedural Vegetation Editor aiming to bring virtualised-geometry density to foliage. The direction is clearly right and the early results are exciting. But experimental means experimental — it is not something to rebuild a shipping vegetation pipeline on this release.

That's why this sits in **Plan the move** rather than 'soon'. Your current hierarchical-instanced-static-mesh foliage is not deprecated and works fine; there is no mature destination to migrate \*to\* yet. The right posture is to understand where the foliage pipeline is heading, prototype against the experimental path to learn its shape, and avoid over-investing in elaborate legacy foliage tooling you'll only have to unwind. Revisit the urgency when the Nanite Foliage path leaves experimental.

Sources: [Epic · UE 5.7 release notes](#)

## 8 Heightfield Landscape as the only terrain option PLAN THE MOVE Epic: Experimental

*The classic heightfield Landscape still coexists and is supported; mesh-based terrain is preview-only — so plan, don't migrate.*

The classic heightfield **Landscape** system is not going anywhere yet — it coexists with newer terrain work and remains supported. What's on the horizon is an experimental **Mesh Terrain** system in the **5.8 preview**, building large landscapes from a 3D mesh rather than a heightfield. If it matures, it's a meaningful rethink of the terrain pipeline; today it exists only in a preview build.

So the verdict is **Plan the move**, and barely on the board at that. There is nothing production-ready to migrate to, so don't. The legacy assumption worth questioning is the architectural one — designing a project as though heightfield Landscape is the \*only\* terrain model the engine will ever offer. Keep the option open, watch 5.8, and don't pour a fortune into bespoke heightfield-only tooling. Revisit when Mesh Terrain ships and earns a status.

Sources: [Epic · UE 5.7 release notes](#)

## ■ Gameplay & Input

Input, abilities and the project-version decision itself. Old defaults that still work but quietly date a project.

### 3 Legacy input mappings (Action/Axis) MIGRATE SOON

*Project-settings action/axis mappings. Enhanced Input is the default now — superseded, still runs.*

The old project-settings **Action and Axis Mappings** have been superseded by **Enhanced Input**, which is the production-ready, expected input system in UE5 — input actions, mapping contexts and modifiers, and the assumed target of nearly all current tutorials and templates. The legacy bindings still function, so a project on them keeps working, but you're building on yesterday's default.

We place this in **Migrate soon**. There is no reason to author new input on the old system, and the migration is well-trodden and well-documented. For an existing project the move is bounded and worth scheduling; for new work it shouldn't even be a question. This is one of the cleaner 'new system has clearly won' transitions in the engine.

Sources: [Epic · UE 5.7 release notes](#)

### 12 Building on a single old engine version MIGRATE NOW

*The meta-migration: anchoring a project to one ageing UE version is the assumption underneath every other blip on this radar.*

The single most consequential migration decision isn't any one subsystem — it's whether you let a project ossify on an old engine version at all. Every other entry here becomes painful in proportion to how long you defer the upgrade: PhysX assumptions, Cascade libraries, legacy input, UE4-era content all compound the further behind you fall. Building indefinitely on one ageing version isn't 'stability', it's debt accruing interest, and it's why we put this in **Migrate now** — the action is to adopt a deliberate upgrade cadence today, not to perform one heroic jump in three years.

The marketplace makes the stakes concrete. By the **MythicLemon Marketplace Index, our own compiled dataset** (as of 2026-05), UE5 is **93.2%** of new monthly releases yet only **53.3%** of the version-tagged catalogue — meaning **46.7%** of tagged listings are still UE4, a standing modernisation backlog of content anchored to the old version. New supply has overwhelmingly moved to UE5; the back-catalogue hasn't followed. For a publisher, that gap is both a warning (don't add to the stale pile) and an opportunity (clean, current-version content stands out against it). Pick a cadence, keep moving, and you'll never face the others on this radar as a crisis.

Sources: [Epic · UE 5.7 release notes](#) · [MythicLemon · Engine migration dashboard](#)

## ■ Animation, Audio & VFX

The content pipelines: particles, sound and skeletal/material authoring. Where the clearest 'old system, new system' replacements live — at very different urgencies.

## 2 Cascade (legacy VFX) MIGRATE SOON

*The old particle system. Superseded by Niagara — supported for now, but the past tense of UE VFX.*

Cascade is the original Unreal particle editor, and **Niagara is its successor** — the production VFX system across the whole 5.x line. Cascade still opens and still plays back, so an existing effect won't break tomorrow, which is exactly why so much legacy content still ships on it. But it is legacy in the plainest sense: no new investment, no new features, and every piece of current learning material assumes Niagara.

We say **Migrate soon** rather than 'now' because it isn't removed and a shipped effect that works is not an emergency. The line we'd draw is on *\*new\** work: never author a new effect in Cascade, and convert your reusable, high-traffic systems to Niagara on a planned cadence rather than in a panic. The longer a Cascade library lives, the more it costs to move, and the day Epic finally retires the editor is a question of when, not if.

🗨️ We sell Niagara VFX packs on Fab, so we benefit commercially when teams move to Niagara. The verdict here reflects Epic's own positioning — Niagara is the production VFX system and Cascade is legacy — not our catalogue; but you should know the overlap.

Sources: [Epic · UE 5.7 release notes](#)

## 9 Sound Cues as the default audio path STILL FINE

*MetaSounds is the modern source — but Sound Cues are still fully supported, so this is 'soon-ish for new work', not an emergency.*

**MetaSounds** is the production-ready, node-based audio system and the modern way to author sound in UE5, with sample-accurate procedural control that the older **Sound Cue** workflow can't match. That makes MetaSounds the clear default for new audio work. But — and we're being deliberately careful here — Sound Cues are **not removed and not formally deprecated**; they remain supported, and plenty of perfectly good audio ships on them.

We place this in **Still fine** rather than 'Migrate soon' precisely to resist over-stating the urgency. The honest call: author *\*new\** sources in MetaSounds, and don't bother ripping out a working Sound Cue setup that does its job. Migrate the pieces where MetaSounds' procedural power actually buys you something — dynamic, parameter-driven audio — and leave simple, static cues alone until you're touching that area anyway. This is the opposite of the Cascade story: same 'old vs new editor' shape, much lower temperature.

🗨️ We sell audio packs on Fab, so we have a commercial interest in modern audio adoption. We've rated this conservatively as 'Still fine' *\*against\** that interest — Sound Cues genuinely aren't deprecated — but the overlap is worth disclosing.

Sources: [Epic · UE 5.7 release notes](#)

## 10 UE4 SkeletalMesh / material authoring assumptions MIGRATE SOON

*Content authored to UE4-era skeletal and material assumptions converts but carries warnings — the marketplace's UE4 backlog lives here.*

This blip is about \*content authored to UE4-era assumptions\* rather than a single API: skeletal-mesh setups and material graphs built before UE5's rendering, skin-weight and material changes. Such content usually opens in 5.7 via the conversion pipeline, but 'opens' and 'is clean' are different things — conversion warnings, shading-model mismatches, and material nodes that behave differently under the modern renderer are the day-to-day reality of dragging UE4-era assets forward.

We say **Migrate soon**, and this is where the marketplace numbers bite. By the **MythicLemon Marketplace Index, our own compiled dataset** (as of 2026-05), **46.7%** of the version-tagged Fab catalogue is still UE4. A large share of that is exactly this kind of content — assets that technically install in a buyer's 5.7 project but greet them with a conversion wall. If you publish, re-authoring to UE5 assumptions is the difference between a clean install and a one-star surprise; if you consume, budget for the conversion rather than assuming a tagged-UE4 asset will land tidily. The newer **Substrate** material framework (production in 5.7) is the longer-horizon destination for material work — but its ecosystem is still catching up, so we'd not force that part of the move this release.

Sources: [Epic · UE 5.7 release notes](#)

## 11 Niagara module/script assumptions STILL FINE

*Niagara is the modern system — but its own internals evolve, so older module/script patterns are 'keep an eye on', not a migration target.*

A subtle one, and we include it so the radar isn't only about leaving old \*systems\*. Niagara itself is the production VFX system and the right place to be — but it's an actively-developed one, and patterns from earlier 5.x versions (deprecated modules, older scratch-pad conventions, emitter setups Epic has since reworked) can throw deprecation notices or behave differently as you move up engine versions. This is intra-Niagara drift, not a Cascade-style 'whole system superseded' story.

We place it in **Still fine**: there's nothing to migrate \*off\* — you're already on the right system — so the posture is hygiene, not urgency. When you bump engine versions, read Niagara's deprecation notices, swap flagged modules for their current equivalents, and keep reusable emitter libraries tidy so the drift doesn't compound. Treat it as routine upkeep on a healthy system, not a project.

🔗 We sell Niagara VFX packs on Fab and maintain reusable Niagara modules, so keeping pace with Niagara's internal changes is part of our own product upkeep. The 'Still fine' verdict reflects that this is maintenance, not migration — but the overlap is worth disclosing.

Sources: [Epic · UE 5.7 release notes](#)

# FAQ

---

## Which engine version is this edition based on?

Unreal Engine 5.7, the current stable release as of June 2026, with notes on the 5.8 preview where a replacement system is about to land. We won't tell you to migrate onto a feature that only exists in a preview build — that's why the experimental replacements sit in 'Plan the move' rather than 'Migrate soon'.

## Why isn't everything old in 'Migrate now'?

Because 'removed' and 'discouraged' are different, and conflating them wastes teams' time. PhysX is genuinely gone, so it's 'Migrate now'. Sound Cues and forward rendering are fully supported, so they're 'Still fine'. The ring reflects real engine status and real urgency, not a blanket 'newer is better' reflex.

## Where does the 46.7% UE4 figure come from, and is it current?

It's the share of the version-tagged Fab catalogue still on UE4, from the MythicLemon Marketplace Index — our own compiled dataset — as of 2026-05. It's a supply / engine-version measure, stamped 'as of' that month, not a live ticker. It's the standing modernisation backlog several blips here describe.

## Why don't you rate any plugins or marketplace assets here?

We sell on Fab, so rating rival products would be marking our own homework. This radar rates Epic's own engine systems, where we have no commercial stake. Where a system overlaps something we sell — Niagara, audio — we disclose it on the blip.

## How often does this come out?

Roughly twice a year, alongside the companion Feature Radar and tracking Epic's major releases. From the next edition, movement arrows will show where we've changed our migration-urgency view — distinct from where Epic changed an actual status.

### HOW TO CITE

MythicLemon, "The Migration & Deprecation Radar", edition 2026.1 (2026-06-13). The Unreal Radar. <https://www.mythic lemon.com/radar/migration-radar-2026-1>