Authoring for the new MTM2 engine - HD art, long names, and pod options

For track and truck authors. Written 2026-08-09, completed 2026-08-11. Covers everything the engine gained in the asset-pipeline work, plus the three authored features that came with it: the **glass** and **foliage** materials (§5) and the **light** object type (§6).

One caveat before you rely on any of it: §5 and §6 are verified on DX11 and Vulkan. DX9 gets its catch-up pass later (see `ROADMAP.md`), so if you author for DX9 today, check your work there rather than assuming parity.

The engineering contracts live in `ENGINE_LIMITS.md`, `FILENAME_CONVENTIONS.md` and `TRACK_VERSIONING.md`. This file is the version you hand to someone who just wants to make art.

---

1. HD textures - PNG and TGA

Drop a `.PNG` or `.TGA` next to (or instead of) the `.RAW` and the engine uses it. Same stem, new extension: `ART\MYROCK.RAW` `ART\MYROCK.PNG`.

Requirements: square, power of two, **32 to 1024** pixels.

Anything else still loads - it is resampled to the nearest supported size and a line appears in `hdtex.txt` saying so. It is a warning, not a rejection. But you chose the resample by accident, so it is worth fixing.

what you supply	what happens
1024x1024 PNG	used at 1024, full quality
700x700 PNG	resampled to 1024, warned
640x480 PNG	resampled to square, warned - **you are losing your aspect ratio**
2048x2048	resampled down to 1024
JPEG renamed to `.PNG`	**refused**, named in `hdtex.txt` (see §8)

Where the size limit really bites

Textures are held in a budget (`[Graphics] hdTextureBudgetMB`, default **1024 MB**). A 1024x1024 costs about **4 MB**, so the budget holds roughly 256 of them.

Exceeding it is **not** an error and nothing is said in-game - the least-recently-drawn textures quietly drop back to their 256x256 versions. On screen that looks like textures softening or popping mid-race. If you see that, check the `(arena)` line in `hdtex.txt` for a non-zero `evicted` count.

---

2. The pure-black trap - read this one

Legacy `.RAW` art takes its transparency from palette index 0. PNG and TGA do not: **alpha is authoritative and no colour is magic.** That is deliberate - it is what stops legitimately dark texels going see-through.

These two rules collide in one very common way:

> Convert a legacy texture to PNG by decoding its indices through its `.act`, and index 0 comes out as whatever colour the `.act` holds there - **almost always black** - at **full opacity**. Every texel that used to be a hole is now solid black paint.

So a fence that used to have gaps becomes a solid black slab. A backdrop that used to show sky through it gets a black band.

The engine watches for this. If a texture is fully opaque **and** more than 2% pure black, you get a line in `hdtex.txt` naming the file. **If you see that line, re-export with real alpha** where you want see-through.

Upscalers are the usual culprit: most output opaque RGB and throw the alpha away.

---

3. Why your PNG becomes `.RAW` in the pod

This surprises everyone. Import a PNG in Traxx and the pod ends up with `MYROCK.RAW` and `MYROCK.ACT` - your PNG appears to have vanished.

It has not. Both are packed:

- `MYROCK.PNG` - your real art, what the new engine draws.
- `MYROCK.RAW` + `MYROCK.ACT` - an 8-bit **fallback**, generated automatically, so the track still

works on an original 1998 install and in the software renderer.

The `.RAW` is the *companion*, not a replacement. The engine prefers the PNG whenever it exists.

Model records name the `.RAW` on purpose. Open a `.bin` in a hex viewer and you will see `MYROCK.RAW`, never `MYROCK.PNG`. That is what keeps older engines working; the new one swaps the extension when it looks for art.

The "do not write the legacy fallback" option

Traxx can skip the `.RAW`/`.ACT` pair. It saves a lot - the fallback is typically **3-11x the size of the HD source**, and can be ~90% of a texture-heavy pod.

The cost: the track becomes **invisible** to an original 1998 MTM2. Not broken - invisible. It does not appear in the track list at all.

That is deliberate. A track that loads with missing textures looks like a bug and generates reports; a track that is simply absent explains itself. See `TRACK_VERSIONING.md` if you want the mechanism.

The current engine is unaffected either way - it reads PNG on every renderer, including DX9 and software.

---

4. Sky and weather art

Sky art is **global** - it applies to every track, not just yours.

The engine names six pieces itself:

file	used for
`CLOUDY2.RAW`	the default sky, MTM1 tracks, and the fallback
`CCLOUDS.RAW`	rain
`DUSKSKY.RAW`	dusk
`NITESKY.RAW`	night
`SUNCRAM.RAW`	**the sun** - a 2x2 atlas of four styles; only the top-left is drawn
`MOON.RAW`	the moon

Cloudy weather has no art of its own. It takes `CLOUDY2` and converts it to greyscale at load. So supplying a cloudy sky means replacing `CLOUDY2`.

`WeatherArt\` *- put your sky packs here*

A loose folder beside the game, checked **before** the pod for those six stems.

Use it rather than `ART\`: extracting a track's art into `ART\` is a routine authoring step and will clobber a sky override sitting there. Nothing ever writes into `WeatherArt\`.

Priority: `WeatherArt\<stem>.PNG` pod `ART\<stem>.PNG` `ART\<stem>.RAW`.

The sky still needs its `.ACT`. The palette band is read separately from the image and is used for the gouraud horizon, so keep the `.ACT` even when supplying a PNG.

Upscaling the sun: `SUNCRAM` is a 2x2 atlas. Keep the layout and the quadrant boundaries intact or the sun will sample the wrong region.

---

5. Materials - glass and foliage

Until now a facet's look came from **which opcode** it was written as, and the opcodes are fixed *combinations*. That is why a windshield could be transparent **or** shiny but never both - the two lived in different families, so the combination simply did

not exist.

Facets can now carry a **material** instead, which is a set of independent switches. You author these in **BinEdit**, as **presets** - you pick "glass" or "foliage", not ten checkboxes.

> **A model using a material needs the new engine.** Older builds cannot read the material records and will not load the model at all. Keep a plain version if you need one to work on a 1998 install. This is the same trade as §3, just less forgiving: there is no automatic fallback.

Glass - transparent and shiny

For windshields and anything that should reflect while you see through it.

- **Tinted glass works** - set the colour and it tints both what you see through it and the

reflection.

- Clear glass is the default; the tint costs nothing when unused.
- It is a material, so it applies to any facet, not only trucks - but trucks are what it was built

for and where it is tested.

Mesh - a grille with real glass behind it

For a window guard, a mesh grille, a tint band across a windscreen, a decal printed on a side window: **one face where part of the surface is solid and the rest is glass.**

Until this preset you could not author that at all, because the solid part and the see-through part are a **pattern in the texture, not separate geometry** - there is nothing to select and give a second material to. Alpha-test gave you open holes with no glass; the Glass preset made the bars translucent along with everything else.

The alpha channel decides. Paint what should be SOLID as opaque texels, and what should stay GLASS as transparent ones.

A legacy `.RAW` mesh texture needs no re-authoring. Transparency there is the black colour key, so a texture that already works as a cut-out works here unchanged - the holes simply become glass instead of holes.

Never put this on a plain window. An ordinary window texture is opaque everywhere, and this preset reads opaque as "solid", so the whole pane goes solid and the glass vanishes. A window that should be glass all over is the **Glass** preset.

On a `.RAW` the glass half is always dark. Colour and alpha are the same colour key there, so a transparent texel *must* be black. Over a dark cab that reads fine; clear or coloured glass in the gaps needs a **PNG with a real alpha channel**, where a texel can be any colour AND transparent.

Small things only. The engine draws a Mesh face twice - once for the glass, once for the solid parts, which is what lets the solid parts sit properly in front of what is behind them. One extra draw and double the fill for that face: nothing for a grille, not something for a whole body.

Foliage - leaves that behave like leaves

The foliage preset turns on four things at once: lit, alpha-tested, **two-sided** and **translucent**.

- **Two-sided** is the one authors notice. Leaf cards used to vanish when seen from behind, so trees

had to be built as crossed pairs or accept disappearing branches. A foliage facet is drawn from both sides.

- **Translucent** lets light through the leaf rather than treating it as a wall.

Use it on leaf cards, fronds, grass tufts, and billboards attached to a 3D trunk. It is not restricted to trees, but that is the case it is tuned for.

The black-key trap again - this bites foliage hardest

§2 explains why converting legacy `.RAW` art to PNG turns transparent holes into solid black paint. Foliage is where it hurts most, because leaf art is *mostly* holes.

There is a second, subtler half you should know about:

> **In legacy art, transparency is keyed on the colour pure black - RGB(0,0,0) - not on palette index 0.** Those are usually the same texel, which is why "index 0" is the folklore. They are not the same rule, and the engine follows the colour.

So a leaf texture with genuinely black shading in it has holes where you painted shadow. If your foliage looks moth-eaten, that is why: **darken to near-black, never to black**, or supply real alpha via PNG and stop relying on the key entirely. The second is strongly preferred for new art.

(A related artefact - distant trees showing a dark rim - was an engine bug in how mip levels averaged the black key into its neighbours. That is fixed; you do not need to author around it.)

5b. Normal maps - bump without touching the model

A normal map adds surface relief: rivets, planking, brick, tread. It changes how light falls across a surface without adding a single polygon.

The whole rule: name it `_N`

Put a file called `<texture>_N.PNG` (or `.TGA`) beside the texture it belongs to.

```
ART\ROCK.RAW          <- the texture a model or track already uses
ART\ROCK_N.PNG        <- its normal map. That is the entire setup.
```

Nothing else. No model edit, no re-export, no setting to tick. A truck or a building authored in 1998 gains a normal map the moment you drop the file next to its art. That is deliberate: the model never names the map, so old models are not excluded.

- The base texture can be **anything** - legacy `.RAW` with its `.ACT`, or a modern `.PNG`. The map is

always a `.PNG` **or** `.TGA`, because it is data rather than art.

- **Never make a `.RAW`/`.ACT` for a normal map.** Squeezing one into 256 palette entries destroys

it - those are *vectors*, not colours. Traxx refuses if you try to add one as a texture, and names the base to add instead.

- `_N` **is reserved.** Do not name an ordinary texture `SOMETHING_N` - it will be read as the

normal map of `SOMETHING`.

It travels with the track by itself

Traxx carries `<texture>_N.PNG` into the pod automatically, beside the texture it belongs to. You do not add it to the texture list and it never appears there - it is a *property* of a texture, not a texture. That also means it can never be offered for deletion as "unused art", and if you remove the base texture the map simply stops being carried.

Strength

There are two controls, and they multiply:

where	what it is for
`[Game] normalStrength` in `monster.ini`, a percentage	the player's/global knob. 100 = as authored
Per material, authored in BinEdit	one material subtler or stronger than the rest

Author at full strength and turn it down, rather than the reverse. A map that looks right at 100 can be softened for free; one authored weak cannot be strengthened without re-exporting.

Where it works, and where it does not

Models, trucks, buildings	yes
Static props (fences, cranes, signs)	yes - they keep their hardware instancing
Terrain	**HD terrain art only** (see §1). Legacy 64x64 ground tiles cannot carry one
Camera-facing trees / billboards	**refused deliberately** - a flat card that always faces you has no real surface for relief to sit on, and forcing it blows the card out white

Night terrain	no directional sun to modulate, so nothing to bump
DX9	not implemented - parked. DX11 and Vulkan only

What good normal-map art looks like here

- **Same dimensions rules as any HD texture** - square, power of two, 32-1024 (§1).
- **Flat is** `(128,128,255)` - the familiar lavender. A flat map is a no-op, which is the correct

result, not a bug.

- **Relief is a redistribution of light, not extra brightness.** The engine clamps the effect so a

bump can darken freely and brighten only up to full diffuse - it can never manufacture a highlight. If you want something to *glow*, that is emissive, not a normal map.

- Very fine, high-contrast detail reads as noise at distance. Prefer relief you can see from where the

player actually is.

What the channels mean - this bites people from other engines

channel	this engine reads it as
R	tangent X - tilt across the texture
G	bitangent Y - tilt down the texture
B	**surface Z**. Export normally; reused internally for occlusion - see below.
A	**specular mask** - dims the shiny highlight per texel. Optional; no alpha channel = unchanged behaviour.

RGB is decoded as `tex * 2 - 1`.

Blue is Z - and you never touch it

If you have worked in an engine that packs **roughness into blue**, or that uses a **DXT5nm**-style layout, **neither applies here.** Export a normal map the way your baker exports one: R, G and B carrying X, Y and Z. That is all that is asked of you.

Behind the scenes the game reconstructs Z from X and Y (a normal is a unit vector, so Z is redundant) and uses the freed blue channel for occlusion. **That is an internal detail and not a format you author against.** Do not pre-pack it yourself - ship the two files below and the game does the merge, correctly, every time.

Occlusion: `<stem>_AO`

> **In short - everything you need to ship one** - Ship `<stem>_AO` beside your diffuse. **Greyscale: white = open, black = fully occluded.** No file = no occlusion. - **Separate file, not packed.** Export the normal map and the AO exactly as your baker gives them. The engine merges them internally - do not pre-pack anything. - Works **with or without** a `_N`. Occlusion alone is fine. - **Size needn't match** the `_N` or the diffuse. Matching is fine too. - Bake it **from the MODEL'S GEOMETRY**, into **the same UV layout as the diffuse** - then it lines up by itself. Do NOT derive it from the diffuse image by desaturating or darkening it: that is not occlusion, it just re-shades what your skin already paints, and reads as mud. **Two things that otherwise look like bugs:** 1. **It darkens ambient light only, never direct sun** - so it is strongest in shade and on overcast/dusk, and subtlest in bright sun. **Check it in shade before judging it.** 2. **Use it on unique skins, not tiling detail textures** - a tiled texture gets identical occlusion everywhere it appears.

Working now. A plain **greyscale** map beside the diffuse, same naming as `_N`:

value	meaning
white (255)	fully open - no occlusion
black (0)	fully occluded
no file	no occlusion, exactly as before

Bake it from your high-poly source the way you normally would. Nothing needs packing, and it works **with or without** a `_N` - ship occlusion alone if that is all you want.

It darkens AMBIENT light only, never direct sun. A crevice reads dark because the sky cannot reach into it; a surface the sun is actually hitting stays lit. That is why this looks like shading rather than like dirt painted on your texture - and it is also why the effect is strongest in shade and on overcast/dusk weather, and subtlest in full sun. That is correct, not a weak map.

Size need not match the `_N` or the diffuse. Occlusion is low-frequency, so a 256 AO beside a 512 normal is fine and costs nothing in quality. Matching sizes is equally fine - there is no reason to go out of your way either way.

`_AO` **BELONGS TO A TEXTURE, NOT TO A PART OF THE MODEL** - and that decides where it is worth using. It is per-texel, but it is attached to the texture, so it lands wherever that texture's UVs land. Bake it into **the same UV layout as the diffuse** (which is what your baker already produces beside the normal map) and it lines up by itself.

SO USE IT ON UNIQUE SKINS, NOT ON TILING DETAIL TEXTURES. A truck body skin is unique - each part of the image maps to one place on the model - so baked occlusion means something there. A tiling texture is the opposite: if the same small metal panel is used on twenty surfaces, one `_AO` repeats the identical occlusion on all twenty, and real occlusion depends on where a surface actually sits. It will be wrong nearly everywhere, and it reads as a repeating stain rather than as shading.

This bites harder than it does for normal maps. A tiled *normal* map still looks like plausible surface detail when it repeats; tiled *occlusion* looks like dirt in impossible places.

When the sizes differ, the AO is resampled to the normal map's grid **nearest-neighbour**. For a baked map that is invisible, because occlusion is smooth by nature. If you have hand-painted an AO with **deliberately hard edges** - a crisp mask rather than a bake - those edges will look slightly chunky at a mismatched size. Author it at the `_N`'s size and the question does not arise.

The alpha channel: a specular mask (2026-08-23)

Alpha dims the shiny highlight per texel, on Gouraud (shiny) faces.

alpha	result
no alpha channel	full highlight - exactly as before. Every existing map is unchanged.
black (0)	full highlight, same as no channel
shades between	darker = shinier, lighter = flatter
white (255)	no highlight - reads like a flat-shaded face

Bumpiness is unaffected. RGB and alpha are read independently, so the relief is identical whether or not you paint alpha.

Flat-shaded faces ignore it - they have no highlight to dim, and their bump still shows.

Black and "no channel" mean the same thing on purpose, so a legacy map cannot lose its highlight by accident. To flatten a whole surface, paint alpha **white**.

Export green-down (DirectX), not green-up (OpenGL)

This engine reads the **DirectX / green-down** convention: green is high at the **bottom** of a bump. Your tool almost certainly has this as a checkbox - "invert green", or a DirectX/OpenGL dropdown.

Nothing warns you if it is wrong, because both are perfectly valid maps. The relief just comes out **inverted** - rivets become dimples, planking becomes grooves. It is a common mistake and a genuinely hard one to spot, so set it once in your export preset and forget it.

A greyscale image is a height map, and it will not work

This is the single commonest mistake. A normal map stores *vectors* and looks predominantly lavender-blue; a height map stores *depth* and looks grey. They are not interchangeable, and the engine does not convert.

Grey `102` decodes to `(-0.2, -0.2, -0.2)` - a negative Z, which is not a normal at all. Any texel with **Z <= 0 is ignored and treated as flat**, so a height map renders as *no bump* rather than as a wrong one. Convert it first; most tools have a height-to-normal step.

> **Check before you ship a map:** blue should be **128 or above everywhere**, averaging north of 200. If the file looks grey rather than lavender, it is a height map.

That same rule quietly protects you elsewhere: if you generate a `_N` from a colour texture that has **transparent areas**, those come through **black**, and black decodes to a normal tilted 125 deg pointing *behind* the surface. Before this was guarded it turned material glass into a solid pane. Now such regions are simply flat - but they are still not doing anything, so paint them properly.

Any texture size can carry a normal map - including 64x64

Tyres, axles, shocks and small parts are usually 64x64, and those work. So does the 256x256 art on a truck body. There is no size prerequisite and nothing to add beside the texture but the `_N` itself.

> **Changed 2026-08-21, and worth knowing if you have read older advice.** Small textures are packed sixteen to a shared page, and a map could not previously be addressed on one - so wheels silently got no bump, and the workaround was to drop a 128px+ **colour** texture beside the legacy one to move the slot onto a page of its own. **That workaround is no longer needed.** A bigger colour texture is still worth having for its own sake; it is simply not a condition for bump any more.

Two surfaces still do not take a map, by design: WATER (its shader has no bump block) and SNOW (it draws from a pre-built atlas with baked coordinates, so nothing could bind one). Dropping an `_N` beside either is not harmless - it takes memory from the shared HD budget and can evict art that is actually being used.

Resolution is independent of the colour texture

The map is registered at **its own** size. A 64x64 skin can carry a 512x512 normal map at full 512, so do not down-res a map to "match" the diffuse - tread and panel lines are exactly the high-frequency detail that benefits from the extra resolution.

6. Lights - the `Light` object type

Tracks can place **real lights** that illuminate the ground and nearby objects, instead of faking a lamp with a glowing texture. In Traxx these are a scenery object whose type is **Light**.

They are **night-time features**: a light shows when the weather and time of day are such that headlights would show. In daylight it costs nothing and does nothing.

The four knobs, and what each actually does

They are deliberately independent - none of them secretly consumes another.

you want	change	notes
the lit area **wider**	**Radius**	this is the pool radius **on the ground**, in feet - it is not a distance from the bulb
the ground **brighter or dimmer**	**Brightness**	0.25 - 3.75. **1.0 = a truck headlight**
the lamp **higher or lower**	**Height**	free - the pool keeps its size *and* its strength at any height
the pool **somewhere other than under the lamp**	**Aim offset**	throws it along the way the object faces, to fake a light striking at an angle

Start at Brightness 0.25, not 1.0

This is the single most useful thing on this page. On a night track a **1.0** light is already saturated across roughly half its radius - it reads as a flat white disc, and raising it further mostly widens the disc. **0.25 is the only setting with no saturated area at all**, so it falls off smoothly from the middle to the rim and actually looks like a light.

Treat 1.0 as "floodlight" and work *down* from it. The range goes to 3.75, but most of the useful authoring is below 1.

The visible bulb is separate from the light

Glow mode controls what you *see*, independently of what the light *does*:

mode	you get
0	no visible source - it lights the ground and draws nothing. **Mode 0 is not "off".**
1	a glow sprite at the lamp head

mode	you get
2	glow **+** a light cone

Mode 0 is the default, and it is the right choice when your model already has a lit-looking lamp head - otherwise the engine draws a bright blob on top of the art you made.

The cone (mode 2) is aimed by rotating the object. **Aim offset runs along that same direction**, so rotating the prop turns the cone and the pool together.

What Traxx shows you

- a **cyan ring** on the ground - the pool footprint, exactly where the light will land, moving live

as you type;

- a **stem** at the post showing the height;
- an **arrow** showing the facing, which is both the cone direction and the aim-offset direction.

The ring means the footprint and *only* the footprint. It deliberately does not try to draw brightness falloff.

Two behaviours worth expecting

- **A light with a non-zero aim offset stops lighting its own post**, because the post is no longer

inside the pool. That is intended - light shines away from its source.

- **Lights are capped at 128 illuminating at once**, shared with truck headlights. Past that the

engine keeps the **nearest** ones rather than failing; a busy track simply loses its most distant lights, and says so once rather than refusing to load.

While you are in the object type list: `Moving`

A second new type, **Moving**, is movement without the rest of the train: no horn, no train engine sound, no headlight. Use it for anything that should travel a path but is not a train. (`Train` still behaves exactly as it always did.)

---

7. File names - how long can they be?

Longer than 8.3, but the limits differ by *what the name is for*, and two of them are permanent.

what	limit	why
Pod entries in `ART\`	23 chars incl. extension	pod directory field is 31 total, minus the folder prefix
Pod entries in `DATA\`	22	"
Pod entries in `WORLD\`	21	"
Pod entries in `MODELS\`	20	"
Texture names **inside a model**	63	needs the new record; older engines cannot read those models
Models named by a **keyframe animation**	**15 total, incl. extension**	permanent - the record size is fixed in the file format

The pod budget is per folder, because the folder prefix is stored in the same field as the name. `MODELS\` costs more than `ART\`, so the same stem may fit one and not the other.

Truck LOD models: the engine writes the detail-level digit at a fixed position for stems longer than 7 characters. Keep LOD-bearing truck model stems short until that changes.

Two ways a name can collide

Longer names removed a length limit. They did **not** remove two places where names are matched by something *other* than the whole name, and both fail **silently** - you get someone else's art rather than an error. Neither is being redesigned: the schemes are shared with every pod authored since 1998, and changing them would break content nobody can re-release. So they are rules for you, not bugs to wait on.

1. A texture is its STEM, not its extension.

`MYSKIN.PNG` and `MYSKIN.RAW` are not two textures. They are two files for **one** texture, and the engine loads exactly one of them - the HD file when it is there, the `.RAW` when it is not (§3).

So you cannot ship two *different* images that happen to share a stem. If you do:

- the engine binds faces to whichever one the probe resolves - normally the `.PNG`;
- the other one is then referenced by nothing;
- in **BinEdit** it shows a face count of zero and is offered for stripping on save, which reads as

the editor deleting art you are still using. It is not: the file on disk is untouched, and the saved model still resolves to the `.RAW` if you remove the `.PNG`. But the model now names the `.PNG`, and the two images have quietly become one.

Give every distinct image a distinct stem. Reuse a stem **only** when the two files are the same picture at two qualities - which is exactly what the HD path is for.

2. Two track stems that share their first 7 characters can share UI bitmaps.

The track logo and map are named `UI\<stem>S.BMP` and `UI\<stem>L.BMP`, and the 8.3 era truncated that stem to **7** so the suffix letter would still fit. `ALPINEVALLEYNORTH` and `ALPINEVALLEYSOUTH` both reduce to `ALPINEV` - one track's logo, on both tracks, with nothing reported.

At 8-character stems this was rare enough to ignore. At 20 it is easy to hit, because long names tend to share a prefix precisely *because* they are descriptive. The good news is that long names also make it trivial to avoid: **make the first 7 characters distinct.** `AVNORTH_Alpine` and `AVSOUTH_Alpine` carry the same meaning and cannot collide.

---

8. When art does not load: read `hdtex.txt`

It sits next to `monster.exe` and names every file it refused and why.

line	meaning
`source LEGACY .RAW - no .PNG/.TGA found for this name`	the engine never found your file - check the name and folder first
`not a readable PNG/TGA`	the file is damaged or truncated
`actually a JPEG` (and similar)	renaming a `.jpg` to `.png` does not convert it - re-export properly
`is not a square power of two in 32..1024 - resampled`	it loaded, but not at the size you intended
`no alpha channel and N% of it is pure black`	**§2**, and **§5** if it is foliage - your transparency is gone
`(arena) ... N evicted`	you are over the texture budget
`(sky) <name> : source PNG/TGA 1024x1024 -> TAKEN`	it worked

The commonest cause of "it did not load" is not the engine refusing it - it is the probe never finding the file. The `source` half of each line tells those apart.

---

9. Quick checklist

Textures

- [ ] Square, power of two, 32-1024.
- [ ] Real alpha where you want see-through - **not** black.
- [ ] Exported as a genuine PNG/TGA, not a renamed JPEG.
- [ ] Sky art in `WeatherArt\`, with its `.ACT`.
- [ ] Names inside the per-folder budget; keyframe-referenced models <= 15.
- [ ] No two *different* images sharing a stem (§7) - `FOO.PNG` and `FOO.RAW` are one texture.
- [ ] Track stems distinct in their **first 7 characters** (§7), or two tracks share a logo.

- [ ] Decided whether to ship the legacy `.RAW` fallback (§3) and understood what omitting it means.
- [ ] Checked `hdtex.txt` after a test run - it is silent when everything worked.

Materials (§5)

- [ ] Understood that a model using a material **will not load on an older engine** - kept a plain version if you need one.
- [ ] Foliage art shaded to *near*-black, never pure black, or given real alpha.
- [ ] Checked glass and foliage on the renderer you actually ship for - DX9 is not yet at parity.

Lights (§6)

- [ ] Started at **Brightness 0.25** and worked up, not down from 1.0.
- [ ] Chose the glow mode deliberately - **0 is not "off"**, it means "no visible bulb".
- [ ] Rotated the object to aim the cone *and* the aim offset; checked the cyan ring is where you want the pool.
- [ ] Remembered that height is free, so mount it where it looks right.

Normal maps (§5b)

- [ ] Named `<texture>_N.PNG` and dropped beside the texture - nothing else done to the model.
- [ ] Saved as **PNG or TGA**, never `.RAW`/`.ACT`.
- [ ] No ordinary texture accidentally named `<something>_N`.
- [ ] Terrain maps only on **HD** ground art - legacy 64x64 tiles cannot carry one.
- [ ] Did not expect one on a camera-facing tree or billboard; those are refused by design.
- [ ] Authored at full strength, then turned down with `[Game] normalStrength` if needed.
- [ ] Checked on **DX11 or Vulkan** - DX9 has no normal maps.