Terrain detail normals and the painted mask

For track authors. Written 2026-08-30. Covers the world-space terrain detail normal and the painted mask that controls where it appears and how strongly.

Verified on DX11 and Vulkan. DX9 does not have this feature at all and is not scheduled to get it - a track authored with a mask simply renders without detail there, exactly as it does today.

---

1. What it does

The engine can overlay a fine tiling **detail normal** on terrain - surface relief that catches the sun, at a scale far below what the terrain textures themselves carry. The engine generates one, so a track needs no art at all; a track can also **supply its own** (§4), which is where most of the visual gain is - generic noise is the weakest form of this.

This is separate from, and stacks with, the per-texture `_N` normal maps described in `AUTHORING_HD_ART.md`. Those give a texture its own material character; this adds relief below the texture's own scale, wherever you have painted for it. Neither affects the other, and per-texture normal maps work on tracks with no mask at all.

It is **off per track.** A track only gets detail if it carries a painted mask - no mask means no detail, which is why every existing track is unaffected. That is the switch, so `terrDetail` does not need to be set to try this: paint a mask and it works.

The detail is **relief and ambient occlusion together** - the surface catches the sun, and its dips and gaps darken. Both come from the same map and are driven by the same strength.

The mask is a greyscale top-down image of the track. Where you paint it brighter, the relief gets stronger. It is addressed in world space, so it lines up with the ground itself and does not move with the camera.

---

2. The workflow

Step 1 - export a reference. Turn on **Options Game Export Terrain Mask Reference**, or set it by hand:

```
[Graphics]
terrMaskExport=1
```

Load the track once. The engine writes `MaskArt\<track>_MASK_ref.png` - a top-down picture of the level built from its **actual terrain art**, tile by tile and honouring each tile's rotation, so road markings, ruts and surface changes all read exactly as they do in the world. Turn the switch back off afterwards; it re-exports on every load.

`terrMaskScale` (default 4) sets texels per ground tile - a tile is 32 feet, so 4 gives 8 feet per texel. **Raise it to 8 or 16 for this.** The default was sized for the old flat-colour reference; real art at 4 texels a tile throws away most of what makes it worth having. A 128x128-tile track at 16 is a 2048x2048 canvas.

It costs some load time, which is why the switch exists and why it is off by default - you only pay it while authoring.

Step 2 - paint. Put your paint on a **new layer above** the reference.

Step 3 - save. Hide or delete the reference layer, then save as `MaskArt\<track>_MASK.PNG` (or `.TGA`), same size and orientation as the reference.

The reference must not be in the saved file. If you flatten your paint onto the coloured map, the engine reads those colours as mask values and every unpainted tile gets a strength you never asked for. It looks fine in the editor and wrong in the game. The engine checks for this and writes a warning to the log if the mask is not greyscale - enable `hdLog=1` to see it.

---

3. The numbers

`terrDetail` in `[Graphics]` is the **floor** - the strength where the mask is black. A white mask reaches **five times** that value. Everything in between is linear.

At the default `terrDetail=10`:

paint (RGB)	multiplier	strength
0 (black)	1.0x	10

paint (RGB)	multiplier	strength
32	1.5x	15
64	2.0x	20
96	2.5x	25
128 (mid grey)	3.0x	30
160	3.5x	35
191	4.0x	40
223	4.5x	45
255 (white)	5.0x	50

Every +32 in the 8-bit value is +5 strength. Going the other way, `RGB = (strength – 10) x 6.375`.

You do not have to hit these exactly - the scale is continuous and filtered smoothly, so a soft brush gives a smooth ramp and eyeballing works.

The same number drives ambient occlusion. Strength is not relief-only: AO is applied at 0.6x whatever the relief strength works out to, so raising `terrDetail` - or painting the mask brighter - deepens the shading in the surface's dips at the same time. They are two properties of one surface and there is deliberately no way to separate them; if the ground looks too dark rather than too bumpy, turn the strength *down* rather than looking for an AO setting.

Picking exact greys in Photoshop

Type the value into the **RGB fields**, the same number in R, G and B.

- **Not the K field.** In Grayscale and CMYK mode K is inverted - K 100% is *black*. "50% grey"

typed as K50 is not RGB 128.

- **HSB's B works** and is quicker: for a neutral grey, B% x 2.55 ~ the RGB value, so B 50 ~ 128.

Keep S at 0 or the mask is no longer neutral and the greyscale check will flag it.

A Solid Color fill layer set to an exact RGB, masked to a region, is easier to adjust later than brushwork. Use a soft brush for the transitions between regions.

---

4. Supplying your own detail normal

Drop `MaskArt\<track>_DTL.PNG` (or `.TGA`, or ship it in the pod as `ART\<track>_DTL.PNG`) and the engine uses it instead of the generated noise. No mask changes are needed - the mask still controls *where* and *how strongly*, this controls *what the relief looks like*.

Size: 256x256 is the recommended size - that is what the generated map is. Square, a power of two, between 32 and 1024.

Anything over 1024 is **refused** and the generated map is used instead: an RGBA map with mips costs four bytes a texel and a 4096-square one is around 64 MB of video memory for something that repeats every few dozen feet. Odd shapes (non-square, not a power of two) still load, with a note in the log - they tile correctly, they just mip less cleanly. Check the log if the result looks wrong.

It must tile. The image repeats every `terrDetailRep` feet, so a seam shows up as a grid across the whole track.

It is a normal map with AO, in the same encoding the per-texture `_N` maps use: red is the normal's X, green its Y, and **blue is ambient occlusion** - 255 unoccluded, 0 fully occluded. Z is reconstructed in the shader, which is what frees blue for AO. Alpha is ignored.

A flat, unoccluded surface is RGB (128, 128, 255). **Leave blue at 255 if you do not want AO** - a blue channel left at 0 by accident reads as fully occluded and paints that ground dark.

The engine's generated map computes its own AO as a cavity map, so pebble gaps read as gaps. AO strength follows the detail strength rather than having its own setting, since the two describe the same surface; the mask scales both together, so ground painted black gets neither.

A track without one silently uses the generated map, so this is purely opt-in and nothing breaks if you leave it out.

---

5. Where the files live

file	purpose
`MaskArt\<track>_MASK_ref.png`	the exported reference - **not** read by the game
`MaskArt\<track>_MASK.PNG`	your painted mask, loose. Overrides the pod copy
`ART\<track>_MASK.PNG`	the mask shipped inside a pod
`MaskArt\<track>_DTL.PNG`	your own tiling detail normal, loose. Overrides the pod copy
`ART\<track>_DTL.PNG`	the detail normal shipped inside a pod

A loose file in `MaskArt\` always wins over the pod copy, so a mask can be overridden without repacking. `MaskArt\` is its own folder rather than `art\` for the same reason `WeatherArt\` is: `art\` is a working directory, and extracting a track's art into it would clobber anything parked there.

`<track>` is the level stem - `SFARE.LVL` `SFARE_MASK.PNG`.

Track-name length. The suffix counts against the pod's 31-character name field, so a long track name plus a long suffix can overrun. `_DTL` is abbreviated for exactly that reason - `_DETAIL` cost three more characters and was reported overrunning on real track names. `_MASK` fits.

If a name does overrun, nothing errors: the file simply stops resolving and the feature quietly does nothing. Check the log (`hdLog=1`) if a mask or detail map is ignored on a long-named track.

---

6. Settings

key	default	meaning
`terrDetail`	10	strength floor, percent - drives **both** relief and AO (AO at 0.6x). 0 disables detail entirely, mask or not
`terrDetailRep`	64	**feet per repeat** of the detail pattern. Keep it a whole number that divides 32
`terrMaskExport`	0	1 = write the painting reference on level load. Also **Options Game**
`terrMaskScale`	4	reference texels per 32-foot tile
`terrDetailDbg`	0	1 = terrain goes white where detail runs; 2 = paint the detail normal as colour

`terrDetailRep` must divide the 32-foot tile (16, 32, 64, 128…). The terrain grid re-centres on the camera in whole tiles, so a repeat length that does not divide one makes the pattern slide as you drive.

---

7. Things that will catch you out

The reference is mirrored relative to how you see the track in game. That matches the level's own map data, so it is expected. You can flip it to work on it, but save the mask back in the reference's orientation or it will be mirrored in the world.

Hard mask edges can shimmer at distance on Vulkan but not DX11 - Vulkan does not build a mip chain for the mask. Feathering the edges avoids it, and looks better on both.

Changing `terrDetail` **needs a track reload**, not just a menu return.

`terrDetail=0` **disables the mask too.** The mask scales the detail, so with no detail there is nothing for it to scale. That is deliberate: someone setting the strength to zero wants the relief off.