

SKYDEFS formal specification 0.99.1

A way of defining custom skies without using transfer lines has not existed in most Doom source ports. This specification documents a data format and corresponding features that replicates and expands Doom's sky rendering functionality.

Scope of features

This specification has been written to handle sky rendering as present in Doom and Doom II: Hell on Earth, as well as expanding the featureset to include Hexen layered skies and Playstation Doom/Doom 64 fire skies. Tall skies are not handled by this specification, and until such time that any future revisions do any implementation (including the style used in Heretic) is to be considered "implementation defined" and not a part of any standard.

JSON lump

This specification uses the JSON Lump 1.0.0 formal specification as the root of its data storage, with a type of "**skydefs**" and a version of "**1.0.0**".

Minimum engine featureset

This specification applies to any **limit-removing** featureset or greater. While it is optional for other valid featuresets, it is a requirement of the **ID24** featureset.

Only one lump

Skydefs are expected to be defined with the **SKYDEFS** lump. If this lump does not exist, then the original behaviour for skies is to be used. For WAD resolution purposes, this lump follows the standard rules and resolves to the last lump found in the WAD dictionary with that name.

Default skies

When **SKYDEFS** are enabled, the flat **F_SKY1** is permanently reserved for default sky behavior. The texture used for this sky is always retrieved for the current level by the appropriate defaults (be they built-in Doom behaviors or skies defined by **UMAPINFO** etc).

A sky defined by a transfer line or via the level defaults always uses the default sky behavior. This is defined as:

- Wrapping around the full 360 degree Z axis where 0-359 degrees maps to 0-1023 X texels.
- Using the horizontal view angle of a target render column to project the sky horizontally.
- Using a non-skewed renderer to render vertically (ie the column renderer).
- Using a screen mid-point of 100 texels for the rendered texture.
- Using a scale that results in rendering 100 texels up from the screen mid-point.
- Applying scrolling values as defined by any transfer lines associated with the rendered sky's sector.

Sky textures and flats

A texture declared for a sky must be found in the **TEXTURE1/2** entries; likewise, a flat must be found in the WAD dictionary between the designated flat markers.

Sky types

There are currently three sky types supported by **SKYDEFS**:

- **Normal** - Just a texture as seen in Doom and Doom II.
- **Fire** - The fire effect as seen in Playstation Doom/Doom 64.
- **With Foreground** - Hexen style skies, where a foreground sky is rendered on top of a background sky and using palette index 0 to indicate a 100% transparent foreground texel (every other palette index is a 100% opaque foreground texel).

Sky parameters

A sky at a minimum must have the following values specified:

- **Type** - Specifies if this is a **Normal**, **Fire**, or **With Foreground** sky.
- **Texture Name** - the texture found in TEXTURE1/2 to be rendered for this sky. This is also used to resolve the sky definition itself.
- **Mid Texel** - the Y texel row that is rendered at the vertical center of the screen.
- **X Scroll Rate** - how many texels per second gets added to the base lookup value during the spherical projection phase.
- **Y Scroll Rate** - how many texels per second gets added to the base lookup value during the vertical rendering phase.
- **X Scale** - how wide an X texel is. This in effect modifies the texel range to $0 - (1024 * (1 / x \text{ scale}))$ during the spherical projection phase.
- **Y Scale** - how tall a Y texel is. This in effect modifies the vertical scale to render $100 * (1 / y \text{ scale})$ texels during the vertical rendering phase.

Animated skies

A sky texture when rendering will always resolve the target texture defined by Doom's animation systems. This resolved texture will use the current sky's properties instead of any additional properties that may or may not be defined as a sky entry with that texture's name.

Fire skies

Fire skies replicate the functionality of the Playstation Doom/Doom 64 fire skies. The reference documentation for this feature [is found on Fabien Sanglard's website](#).

Fire skies have the following parameters:

- **Palette** - a linear series of palette indices, representing the palette index that the fire decays to (and is cleared to) to the paletteindex that the fire starts at in sequential order.
- **Update Time** - how many seconds it takes for the fire to perform the next step of the simulation. This must be a minimum of one rendered frame.

Fire skies are to overwrite the texture data for the target texture. This will result in the texture always displaying the generated fire instead of the original texture data. It is recommended as a result that you use unique textures for your fire targets. Note that this requirement means that a fire sky texture can be inside an **ANIMATED** definition and show up in another sky definition as a result.

Fire skies are expected to be primed with both the bottom row set to the last palette index; and with a minimum of 64 iterations performed to populate the target image with some fire to begin with.

Hexen MAPINFO

When reading Hexen **MAPINFO** structures, the sky layers and scroll rates are to be considered a part of the per-map default sky definition and do not create unique sky entries in the runtime code.

Flat definitions

The flat definitions exist to map the standard sky functionality to any flat found in the WAD dictionary, and resolve to the defined texture as found in **TEXTURE1/2**. This is a way to define skies globally without having to implement sky transfer lines, mapinfo definitions, or any other method.

Data type definitions

root

Name	Type	Description
skies	array of sky	An array of skies. Can be null; if not null, it must contain a minimum of one entry. Any other value is invalid and an error condition.
flatmapping	array of flatmap	An array of flat definitions. Can be null; if not null, it must contain a minimum of one entry. Any other value is invalid and an error condition.

flatmap

Name	Type	Description
flat	string	A string representing a valid flat as found in the WAD dictionary.
sky	string	A string representing a valid texture as found in TEXTURE1/2.

sky

Name	Type	Description
type	integer	An enumeration with the following valid values: <ul style="list-style-type: none"> • 0 - Normal • 1 - Fire • 2 - With Foreground Any other value is invalid and an error condition.
name	string	A string representing a valid texture as found in TEXTURE1/2.
mid	number	A value representing the Y texel row that is rendered at the vertical center of the screen.
scrollx	number	A value representing how many texels per second gets added to the base lookup value during the spherical projection phase.
scrolly	number	A value representing how many texels per second gets added to the base lookup value during the vertical rendering phase.
scalex	number	A value representing how wide an X texel is.
scaley	number	A value representing how tall a Y texel is.
fire	fire	An object containing the parameters for a fire sky. If type is not Fire , then it is considered an error condition if not null.
foregroundtex	foretex	An object containing the parameters for a foreground texture. If type is not With Foreground , then it is considered an error condition if not null.

fire

Name	Type	Description
palette	array of integers	An array of fire palette indices. Must contain at least one value; zero values or null is invalid and an error condition.
updatetime	number	A value representing how many seconds it takes for the fire to perform the next step of the simulation. Must be greater than zero; it is considered an error condition otherwise.

foretex

Name	Type	Description
name	string	A string representing a valid texture as found in TEXTURE1/2.
mid	number	A value representing the Y texel row that is rendered at the vertical center of the screen.
scrollx	number	A value representing how many texels per second gets added to the base lookup value during the spherical projection phase.
scrolly	number	A value representing how many texels per second gets added to the base lookup value during the vertical rendering phase.
scalex	number	A value representing how wide an X texel is.
scaley	number	A value representing how tall a Y texel is.

Reference implementation details

Sky transparency

A custom render path is not required for the foreground sky to render transparently. All that is needed is a correctly cooked transparency map, which results in the backbuffer texel being used when the texture texel is a 0 index. While this map is generated at compile time for our implementation, a copy of its contents can be found as SKYTRAN.Imp in the bundled resources.

Note that as fire skies use a texture from TEXTURE1/2 as a target, that the result of a fire sky can be used as a foreground texture with all features available (including transparency).