

Translations formal specification 0.99.1

Translations from one palette range to another have always been hardcoded into the Doom engine, but only specifically for players 2-4 in a multiplayer match. This specification details a way of defining translation tables through data, as well as a few other handy things to have when dealing with multiplayer matches.

JSON lump

This specification uses the JSON Lump 1.0.0 formal specification as the root of its data storage, with a type of “**translation**” 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.

The translation table

The table field is a literal lookup table from one palette index to another. As such, the table must be exactly 256 entries or else it is considered an error condition.

Built-in translations

To replicate vanilla functionality, the following built-in translations exist for player sprites:

- T_GREEN
- T_INDIGO
- T_BROWN
- T_RED

These can be overridden by a lump found in the WAD dictionary following normal lump resolution rules.

In addition, four extra translations are built-in. While matching the ranges defined for these translations is not a requirement of this specification, for reference the ranges they remap to are included below:

- T_YELLOW - index 160 to 167
- T_BLUE - index 196 to 207
- T_NAVY - index 240 to 247
- T_MAGENTA - index 250 to 255

The default tables define the **sbarback**, **interback**, and the **table** fields to their vanilla equivalents (with the extra translations using the T_GREEN background and translated with the defined table at render time), with the order of entries defined above corresponding to both player numbers and the order of lumps resolved for multiplayer purposes.

Translation initialisation order

Translation lumps are loaded in at renderer initialisation. They are added to the translation lookup table in the following order:

- Lumps defined by **Translation** entries in a thing's Dehacked entry
- Translations defined in the **playertranslations** field in GAMECONF

- Built-in tables that have not been previously loaded as a lump

Player translations

Player translations are defined by the GAMECONF **playertranslations** table. If this is undefined by data, a default table comprised of the built-in translations in the order specified above will be inserted into the active GAMECONF.

All player things will always have a valid translation table. A default implementation can be achieved with the following formula:

```
translation = playertranslations[ playernum % <playertranslations count>
];
```

This default implementation can be overridden by however a port chooses to implement player preferences.

Thing flags indicating translations

Any thing that has the MF_TRANSLATION field set in its mobjinfo_t flags and does not have its **Translation** field set must resolve its translation from the **playertranslations** table.

Guidelines for editor authors

It is anticipated that users will want to use [ZDoom style translations](#) when creating their translation lumps. However, all lump specifications in the RNR24 feature suite have been designed to be reflections of data as opposed to command lists. As such, it is recommended that editors use the metadata field of a JSON lump to store ZDoom translation information and apply it to the table field on modification/save.

Note that being palette based, any true colour translations should map to the closest colour found in the palette.

Data type definitions

root

Name	Type	Description
name	string	A dehashed mnemonic representing the name of this translation.
sbarback	string	A lump representing the face background on the status bar. Can be null. If this lump does not exist, it is an error condition.
sbartranslate	bool	Determines if sbarback should use the translation table to render.
interback	string	A lump representing the player background on the interlevel screens. Can be null. If this lump does not exist, it is an error condition.
intertranslate	bool	Determines if interback should use the translation table to render.

table	array of integers	An array of translation palette indices. Must be exactly 256 elements; any other length is an error condition.
--------------	-------------------	--