

ID24HACKED hashing formal specification 0.99.1

In order to eliminate undefined behavior, **ID24HACKED** has the ability to hash a DeHackEd state and require additional DeHackEd patches to specify a valid hash in order to successfully load. This document describes everything required to implement DeHackEd Hashing.

A DeHackEd patch is always assumed to work with just the built-in tables, and as such a DeHackEd patch will always be processed when the engine is in its default state of zero patches loaded. If the engine has already parsed a DeHackEd patch, any further patches will fail to load unless they specify a valid hash value that matches the engine's current state.

DeHackEd patch loading order

As **ID24** is required to eliminate undefined behavior, the loading order of DeHackEd patches has changed from what was defined by MBF. This behavior retroactively applies to all previously published specifications.

DeHackEd patches must be parsed in the following order:

- IWAD DEHACKED entry (last found)
- For each PWAD listed in order
 - PWAD DEHACKED entry (last found)
- Each DEH file listed in order

This ensures both that the hashing function described in this document works as intended, and that any mod shipping with a DeHackEd patch doesn't have weird bugs introduced by a user-defined DeHackEd patch.

The hash function used

FNV-1a with 64-bit values is used to hash each required value. An initial value of 14695981039346656037 is used to begin the hashing, and a prime of 1099511628211 is used in the transformation operation.

All bytes of an integer are passed into the hash function in little endian format.

Any undefined or zero-length string values must be hashed using the value "**null**" instead of a null pointer or a zero value.

Reading hashes from a DeHackEd file

A new field exists at the root level of a DeHackEd document: **valid hashes**. This is a comma-separated list of hash values that a DeHackEd patch is allowed to load on top of. Each hash is a 64-bit unsigned integer encoded as uppercase hexadecimal. If no patches are loaded or if the loaded hash is equal to one found in the **valid hashes** entry, then a DeHackEd patch is considered valid and can be loaded.

NOTE: At the time of publishing, the Thing and Frame tables are incomplete in the officially released Doom and Doom II applications. As such, it is impossible for community generated hash values to match any hash provided. This has the effect of limiting the user to loading one DeHackEd patch at a time. This limitation will be removed in a patch in the near future.

Data types

The following datatypes are considered in the hashing algorithm:

- Miscellaneous values
- Thing
- Frame
- Sprite
- Sound
- Weapon
- Ammo

Each item is parsed from the **in-order tables** in exactly the order they were allocated in. As objects are resolved as they are referred to in **ID24HACKED**, this means the allocation order matches the first time their index is referenced in a DeHackEd patch rather than the first time their complete definition is encountered.

Strings, cheats, and par times are not considered for hashing as these are values that by definition cannot result in undefined behaviour.

Code pointers in a Frame entry hash using their lookup mnemonic; or “**null**” if no code pointer is defined

Any undefined or zero-length string values must be hashed using the value “**null**” instead of a null pointer or a zero value.

When the **feature level** of the previous and currently loaded DeHackEd patches does not require items added at higher feature levels, these items are not considered when hashing. DeHackEd always starts off at a **Doom 1.9** feature level and will not consider feature levels defined outside of the DeHackEd parser (such as through the command line or a GAMECONF lump).

Data processing orders

Miscellaneous values

Field	Condition
Initial Health	Always
Max Health	Always
Max Armor	Always
Green Armor Class	Always
Blue Armor Class	Always
Max Soulsphere	Always
Soulsphere Health	Always
Megasphere Health	Always

God Mode Health	Always
IDFA Armor	Always
IDFA Armor Class	Always
IDKFA Armor	Always
IDKFA Armor Class	Always
BFG Cells/Shot	Feature level < MFB21
Initial Bullets	Feature level < ID24

Thing

Field	Condition
Thing index	Always
ID #	Always
Initial frame	Always
Hit points	Always
First moving frame	Always
Alert sound	Always
Reaction time	Always
Attack sound	Always
Injury frame	Always
Pain chance	Always
Pain sound	Always
Close attack frame	Always
Far attack frame	Always
Death frame	Always
Exploding frame	Always
Death sound	Always
Speed	Always
Width	Always
Height	Always

Mass	Always
Missile damage	Always
Action sound	Always
Bits	Always
Respawn frame	Always
MBF21 Bits	Feature level >= MFB21
Infighting group	Feature level >= MFB21
Projectile group	Feature level >= MFB21
Splash group	Feature level >= MFB21
Fast speed	Feature level >= MFB21
Melee range	Feature level >= MFB21
Rip sound	Feature level >= MFB21
ID24 Bits	Feature level >= ID24
Min respawn tics	Feature level >= ID24
Respawn dice	Feature level >= ID24
Dropped item	Feature level >= ID24
Pickup ammo type	Feature level >= ID24
Pickup ammo category	Feature level >= ID24
Pickup weapon type	Feature level >= ID24
Pickup item type	Feature level >= ID24
Pickup bonus count	Feature level >= ID24
Pickup sound	Feature level >= ID24
Pickup message	Feature level >= ID24
Translation	Feature level >= ID24

Frame

Field	Condition
Frame index	Always
Code pointer	Always

Sprite number	Always
Sprite subnumber	Always
Duration	Always
Next frame	Always
Unknown 1	Always
Unknown 2	Always
Injury frame	Always
MBF21 Bits	Feature level >= MBF21
Args1	Feature level >= MBF21
Args2	Feature level >= MBF21
Args3	Feature level >= MBF21
Args4	Feature level >= MBF21
Args5	Feature level >= MBF21
Args6	Feature level >= MBF21
Args7	Feature level >= MBF21
Args8	Feature level >= MBF21
Tranmap	Feature level >= ID24

Sprites

Field	Condition
Sprite index	Always
Sprite name	Always

Sounds

Field	Condition
Sound index	Always
Sound name	Always

Weapons

Field	Condition
Weapon index	Always
Ammo type	Always
Deselect frame	Always
Select frame	Always
Bobbing frame	Always
Shooting frame	Always
Firing frame	Always
Ammo per shot	Feature level >= MBF21
MBF21 Bits	Feature level >= MBF21
Slot	Feature level >= ID24
Slot Priority	Feature level >= ID24
Switch Priority	Feature level >= ID24
Initial Owned	Feature level >= ID24
Initial Raised	Feature level >= ID24
Carousel icon	Feature level >= ID24
Allow switch with owned weapon	Feature level >= ID24
No switch with owned weapon	Feature level >= ID24
Allow switch with owned item	Feature level >= ID24
No switch with owned item	Feature level >= ID24

Ammos

Field	Condition
Ammo index	Always
Per Ammo	Always
Max Ammo	Always

Initial ammo	Feature level >= ID24
Max upgraded ammo	Feature level >= ID24
Box ammo	Feature level >= ID24
Backpack ammo	Feature level >= ID24
Weapon ammo	Feature level >= ID24
Dropped ammo	Feature level >= ID24
Dropped box ammo	Feature level >= ID24
Dropped backpack ammo	Feature level >= ID24
Dropped weapon ammo	Feature level >= ID24
Deathmatch weapon ammo	Feature level >= ID24
Skill 1 multiplier	Feature level >= ID24
Skill 2 multiplier	Feature level >= ID24
Skill 3 multiplier	Feature level >= ID24
Skill 4 multiplier	Feature level >= ID24
Skill 5 multiplier	Feature level >= ID24