

## Contents

<b>GameDB Documentation</b>	<b>1</b>
YAML Game Format . . . . .	1
A Note on Case Sensitivity . . . . .	3
Compatibility . . . . .	3
Rounding Modes . . . . .	3
Options for rounding . . . . .	3
Clamping Modes . . . . .	4
eeClampMode . . . . .	4
vuClampMode . . . . .	4
Game Fixes . . . . .	4
Options for Game Fixes . . . . .	4
SpeedHacks . . . . .	6
Options for SpeedHacks . . . . .	6
Memory Card Filter Override . . . . .	6
Patches . . . . .	6

## GameDB Documentation

### YAML Game Format

The following is an annotated and comprehensive example of everything that can be defined for a single Game entry.

```

SERIAL-12345: # !required! Serial number for the game, this is how games
↳ are looked up. Case insensitive
  name: "A Sample Game" # !required!
  region: "NTSC-U" # !required!
  compat: 0
  roundModes:
    eeRoundMode: 0
    vuRoundMode: 3
  clampModes:
    eeClampMode: 0
    vuClampMode: 3
  # If a GameFix is included in the list, it will be enabled.
  # If you'd like to temporarily disable it, either comment out the line,
  ↳ or remove it!

```

```

gameFixes:
  - VuAddSubHack
  - FpuMulHack
  - FpuNegDivHack
  - XGKickHack
  - IPUWaitHack
  - EETimingHack
  - SkipMPEGHack
  - OPHFlagHack
  - DMABusyHack
  - VIFFIFOHack
  - VIF1StallHack
  - GIFFIFOHack
  - GoemonTlbHack
  - IbitHack
  - VUKickstartHack
  - VUOverflowHack
# The value of the speedhacks is assumed to be an integer,
# but at the time of writing speedhacks are effectively booleans (0/1)
speedHacks:
  mvuFlagSpeedHack: 0
  InstantVU1SpeedHack: 0
memcardFilters:
  - "SERIAL-123"
  - "SERIAL-456"
# You can define multiple patches, but they are identified by the CRC.
patches:
  default: # Default CRC!
    author: "Some Person"
    content: |- # !required! This allows for multi-line strings in YAML,
→ this type preserves new-line characters
    comment=Sample Patch
    patch=1,EE,00000002,word,00000000
  crc123: # Specific CRC Patch!
    author: "Some Person"
    content: |-
    comment=Another Sample
    patch=1,EE,00000001,word,00000000

```

Note that quoting strings in YAML is optional, but certain characters are reserved like "\*" and require the string to be quoted, be aware / use a YAML linter to avoid confusion.

## A Note on Case Sensitivity

Both the serial numbers for the games, and the CRC patches are at the moment not case-sensitive and will be looked up with their lowercase representations. **However, stylistically, uppercase is preferred and may be enforced and migrated to in the future.**

For example:

- SLUS-123 will be stored and looked up in the GameDB as slus-123
- Likewise, a CRC with upper-case hex 23AF6876 will be stored and looked up as 23af6876

However, YAML is case-sensitive and will allow multiple serials that only differ on casing. To prevent mistakes, this will also throw a validation error and the first entry will be the one that wins.

**Everything else can be safely assumed to be case sensitive!**

## Compatibility

compat can be set to the following values:

- 0 = Unknown Compatibility Status
- 1 = Nothing
- 2 = Intro
- 3 = Menu
- 4 = In-game
- 5 = Playable
- 6 = Perfect

## Rounding Modes

The rounding modes are numerically based.

These modes can be specified either on the **EE** (eeRoundMode) or **VU** (vuRoundMode)

### Options for rounding

- 0 = **Nearest**
- 1 = **Negative Infinity**
- 2 = **Positive Infinity**
- 3 = **Chop (Zero)**
- The is the common default

## Clamping Modes

The clamp modes are also numerically based.

- `eeClampMode` refers to the EE's FPU co-processor and COP2
- `vuClampMode` refers to the VU's in micro mode

### eeClampMode

- 0 = **Disables** clamping completely
- 1 = Clamp **Normally** (only clamp results)
- 2 = Clamp **Extra+Preserve Sign** (clamp results as well as operands)
- 3 = **Full Clamping** for FPU

### vuClampMode

- 0 = **Disables** clamping completely
- 1 = Clamp **Normally** (only clamp results)
- 2 = Clamp **Extra** (clamp results as well as operands)
- 3 = Clamp **Extra+Preserve Sign**

## Game Fixes

These values are case-sensitive so take care. If you incorrectly specify a GameFix, you will get a validation error on startup. Any invalid game-fixes will be dropped from the game's list of fixes.

### Options for Game Fixes

- `VuAddSubHack`
  - Tri-ace games, they use an encryption algorithm that requires VU ADDI opcode to be bit-accurate.
- `FpuMulHack`
  - Tales of Destiny hangs.
- `FpuNegDivHack`
  - Gundam games messed up camera-view. Dakar 2's sky showing over 3D. Others...

- XGKickHack
  - Use accurate timing for VU XGKicks (Slower). Fixes graphical errors on WRC, Erementar Gerad, Tennis Court Smash and others.
- IPUWaitHack
  - FFX FMV, makes GIF flush before doing IPU work. Fixes bad graphics overlay.
- EETimingHack
  - General purpose timing hack.
- SkipMPEGHack
  - Finds sceMpegIsEnd pattern in games and then recompiles code to say the videos are finished.
- OPHFLagHack
  - Bleach Bankais and others.
- DMABusyHack
  - Mana Khemia, Metal Saga. Denies writes to the DMAC when it's busy.
- VIFFIFOHack
  - Transformers Armada, Test Drive Unlimited. Fixes slow booting issue.
- VIF1StallHack
  - SOCOM II HUD and Spy Hunter loading hang.
- GIFFIFOHack
  - Enables the GIF FIFO. Needed for Wallace & Grommit, Hot Wheels, DJ Hero.
- GoemonTLbHack
  - Preload TLB hack to avoid tlb miss on Goemon.
- IbitHack
  - VU I bit Hack avoid constant recompilation in some games (Scarface The World Is Yours, Crash Tag Team Racing).
- VUKickstartHack
  - Let the VU's both run ahead of the EE to fix some timing issues.
- VUOverflowHack
  - VU Overflow hack to check for possible float overflows (Superman Returns).

## SpeedHacks

These values are in a key-value format, where the value is assumed to be an integer.

### Options for SpeedHacks

- `mvuFlagSpeedHack`
- Accepted Values - 0 / 1
- Katamari Damacy have weird speed bug when this speed hack is enabled (and it is by default)

<!-- [list-item-spacing] Missing new line after list item-->

- `InstantVU1SpeedHack`
- Accepted Values - 0 / 1
- Games such as Parappa the Rapper 2 need VU1 to sync, so you can force disable the speedhack here

## Memory Card Filter Override

By default, the `FolderMemoryCard` filters save games based on the game's serial, which means that only saves whose folder names contain the game's serial are loaded.

This works fine for the vast majority of games, but fails in some cases, for which this override is for. Examples include multi-disc games, where later games often reuse the serial of the previous disc(s), and games that allow transfer of savedata between different games, such as importing data from a prequel.

Values should be specified as a list of strings, example shown above.

## Patches

The patch that corresponds to the running game's CRC will take precedence over the default. Multiple patches using the same CRC cannot be defined and this will throw a validation error.

CRCs are case-insensitive, however uppercase is preferred stylistically!

Patches should be defined as multi-line string blocks, where each line would correspond with a line in a conventional `*.pnach` file

For more information on how to write a patch, see the following [forum post](#)