

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. Can be anything.
  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
    - EETimingHack
    - SkipMPEGHack
    - OPHFlagHack
    - DMABusyHack
    - VIFFIFOHack
    - VIF1StallHack
    - GIFFIFOHack
    - GoemonTlbHack
    - IbitHack
    - FullVU0SyncHack
    - VUSyncHack
    - VUOverflowHack
    - SoftwareRendererFMVHack
  # The value of the GS Fixes is assumed to be an integer
  gsHWFixes:
    mipmap: 1
    preloadFrameData: 1
  # The value of the speedhacks is assumed to be an integer,
  speedHacks:
    mvuFlag: 0
    InstantVU1: 0
    mtvu: 0
```

```

eeCycleRate: 2
memcardFilters:
  - "SERIAL-123"
  - "SERIAL-456"
# You can define multiple patches, but they are identified by the CRC.
patches:
  default: # Default CRC!
    content: |- # !required! This allows for multi-line strings in YAML, this type preserves whitespace
      comment=Sample Patch
      patch=1,EE,00000002,word,00000000
  crc123: # Specific CRC Patch!
    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 where 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`) with(`out`) (`eeDivRoundMode`) or **VU** (`vuRoundMode`) or specific VUs like **VU0** (`vu0RoundMode`) or **VU1** (`vu1RoundMode`)

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 VUs in micro mode
- `vu0ClampMode` refers to the VU0 in micro mode
- `vu1ClampMode` refers to the VU1 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**

GS Hardware Fixes

[] = GameDB values { } = GUI options () = Default values

GS Hardware Mipmap Fixes

- `mipmap` [0 or 1] {Off, On} Default: On (looks up GameDB)
- `trilinearFiltering` [0 or 1 or 2] {None, Trilinear, Trilinear Ultra} Default: None (0)

GS Hardware General Fixes

- beforeDraw {OI with suffix } {None unless specific game GSC} Default: Automatic (No value, looks up GameDB) with valid variable name (ex. OI_BurnoutGames)
- moveHandler {MV with suffix } {None unless specific game GSC} Default: Automatic (No value, looks up GameDB) with valid variable name (ex. MV_Ico)
- afterDraw {OO with suffix } {None unless specific game GSC} Default: Automatic (No value, looks up GameDB) with valid variable name
- conservativeFramebuffer [0 or 1] {Off or On} Default: On (1)
- texturePreloading [0 or 1 or 2] {None, Partial or Full Hash Cache} Default: None (0)
- deinterlace [Value between 0 to 9] {Automatic, Off, WeaveTFF, WeaveBFF, BobTFF, BobBFF, BlendTFF, BlendBFF, AdaptiveTFF, AdaptiveBFF} Default: Automatic (No value, looks up GameDB)

GS Hardware Renderer Fixes

- autoFlush [0 or 1 or 2] {Disabled, Enabled (Sprites Only), Enabled (All Primitives)} Default: Off (0)
- partialTargetInvalidation [0 or 1] {Off, On} Default: Off (0)
- PCRTCOffsets [0 or 1] {Off, On} Default: Off (0)
- PCRTCOverscan [0 or 1] {Off, On} Default: Off (0)
- disableDepthSupport [0 or 1] {Off, On} Default: Off (0)
- disablePartialInvalidation [0 or 1] {Off, On} Default: Off (0)
- cpuFramebufferConversion [0 or 1] {Off, On} Default: Off (0)
- preloadFrameData [0 or 1] {Off, On} Default: Off (0)
- textureInsideRT [0 or 1 or 2] {Disabled, Inside Targets, Merge Targets} Default: Off (0)
- PCRTCOverscan [0 or 1] {Off, On} Default: Off (0)
- PCRTCOverscan [0 or 1] {Off, On} Default: Off (0)
- cpuCLUTRender [0 or 1 or 2] {Disabled, Normal, Aggressive} Default: Disabled (0)
- cpuSpriteRenderBW [Value between 0 to 10] {Disabled, 1 (64), 2 (128), 3 (192), 4 (256), 5 (320), 6 (384), 7 (448), 8 (512), 9 (576), 10 (640)} Default: Off (0)

- cpuSpriteRenderLevel [0 or 1 or 2] {Sprites only, Sprites/Triangles, Blended Sprites/Triangles} Default: Off unless cpuSpriteRenderBW has value other than Off then it is ‘Sprites only’ (0)
- estimateTextureRegion [0 or 1] {Off, On} Default: Off (0)
- getSkipCount {GSC with suffix } {None unless specific game GSC} Default: Disabled (0) unless valid variable name (ex. GSC_PolyphonyDigitalGames, GSC_UrbanReign, ...)
- gpuPaletteConversion [0 or 1] {Off, On} Default: Off (0)
- gpuTargetCLUT [0 or 1 or 2] {Disabled, Enabled (Exact Match), Enabled (Check Inside Target)} Default: Disabled (0)
- minimumBlendingLevel [0 or 1 or 2 or 3 or 4 or 5] {Minimum, Basic, Medium, High, Full(Slow), Maximum (Very Slow)} Default: Automatic (No value, looks up GameDB)
- maximumBlendingLevel [0 or 1 or 2 or 3 or 4 or 5] {Minimum, Basic, Medium, High, Full(Slow), Maximum (Very Slow)} Default: Automatic (No value, looks up GameDB)
- recommendedBlendingLevel [0 or 1 or 2 or 3 or 4 or 5] {Minimum, Basic, Medium, High, Full(Slow), Maximum (Very Slow)} Default: Automatic (No value, looks up GameDB)
- readTCOnClose [0 or 1] {Off, On} Default: Off (0) // Tab 3 Hardware Fixes (4th checkbox on right row 2)

GS Hardware Upscaling Fixes

- alignSprite [0 or 1] {Off or On} Default: Off (0)
- mergeSprite [0 or 1] {Off or On} Default: Off (0)
- forceEvenSpritePosition [0 or 1] {Off or On} Default: Off (0)
- bilinearUpscale [0 or 1 or 2] {Automatic, Force Bilinear, Force Nearest} Default: Automatic
- skipDrawStart [Value between 0 to 10000] {0-10000} Default: Off (0)
- skipDrawEnd [Value between 0 to 10000] {0-10000} Default: Off (0)
- halfPixelOffset [0 or 1 or 2 or 3 or 4 or 5] {Off, Normal Vertex, Special (Texture), Special (Texture Aggressive), Align to Native, Align to Native with Texture Offsets} Default: Off (0)
- nativeScaling [0 or 1 or 2 or 3 or 4] {Off, Normal, Aggressive, Normal (Maintain Upscale) or Aggressive (Maintain Upscale) } Default: Normal (0)
- nativePaletteDraw [0 or 1] {Off, On} Default: Off (0)
- roundSprite [0 or 1 or 2] {Off, Half or Full} Default: Off (0)

Game Fixes

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

Game Fixes Options

- **FpuMulHack**
 - For Tales of Destiny: This fix addresses hanging issues.
- **SoftwareRendererFMVHack**
 - Used for complex FMV rendering in certain games.
- **SkipMPEGHack**
 - Prevents game hanging/freezing by skipping videos/FMVs.
- **GoemonTlbHack**
 - Preload TLB hack to prevent TLB miss on Goemon.
- **EETimingHack**
 - General-purpose timing hack affecting Digital Devil Saga, SSX, and others.
- **InstantDMAHack**
 - Resolves cache emulation problems, affecting games like Fire Pro Wrestling Z.
- **OPHFlagHack**
 - Affects games like Bleach Blade Battlers, Growlanser II and III, Wizardry.
- **GIFFIFOHack**
 - Corrects but slows down rendering, impacting games like FIFA Street 2.
- **DMABusyHack**
 - Affects games like Mana Khemia 1, Metal Saga, Pilot Down Behind Enemy Lines.
- **VIF1StallHack**
 - Resolves hang issues in games like SOCOM 2 HUD and Spy Hunter.
- **VIFFIFOHack**
 - Simulates VIF1 FIFO read ahead, affecting games like Test Drive Unlimited, Transformers.
- **FullVU0SyncHack**
 - Enforces tight VU0 sync on every COP2 instruction.
- **IbitHack**
 - Avoids constant recompilation in games like Scarface: The World is Yours, Crash Tag Team Racing.
- **VuAddSubHack**
 - For Tri-Ace Games: Star Ocean 3, Radiata Stories, Valkyrie Profile 2.
- **VUOverflowHack**
 - Checks for possible float overflows (Superman Returns).

- **VUSyncHack**
 - Ensures synchronization between VUs and EE to fix timing issues.
- **XGKickHack**
 - Uses accurate timing for VU XGKicks, affecting games like WRC, Erementar Gerad, Tennis Court Smash.
- **BlitInternalFPSHack**
 - Utilizes an alternative method to calculate internal FPS to avoid false readings in some games.

SpeedHacks

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

Options for SpeedHacks

- **mvuFlag**
- Accepted Values - 0 / 1
- Katamari Damacy has a peculiar speed bug when this speed hack is enabled (and it is by default)
- **mtvu**
- Accepted Values - 0 / 1
- T-bit games dislike MTVU, and some games are incompatible with MTVU.
- **instantVU1**
- Accepted Values - 0 / 1
- Games such as PaRappa the Rapper 2 need VU1 to sync, so you can force sync with this parameter.
- **eeCycleRate**
- Accepted Values - -3 / 3

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, 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. This can unlock certain content ranging from cosmetic features (gear/ visual costume) to mechanics (early unlock of weapons). These are not needed with file memcards as they don't hide the serials or outside date from the game itself nor its save memcard. **

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

Editor Tooling

We provide a JSON Schema for the GameDB's format. You can use this to validate the file, and assist in writing it properly.

VSCode Integration

If you use VSCode and you want it to properly lint the `GameIndex.yaml` file you should:

1. Download the YAML extension - <https://marketplace.visualstudio.com/items?itemName=redhat.vscode-yaml>
2. Add the following to your settings:

```
"yaml.schemas": {  
    "https://raw.githubusercontent.com/PCSX2/pcsx2/master/pcsx2/Docs/gamedb-schema.json": "**/  
},
```