PS2 port of Half-life has PAK files of two types: normal and compressed.

# 1 Normal PAKs

Structure of file is shown on fig. 1.
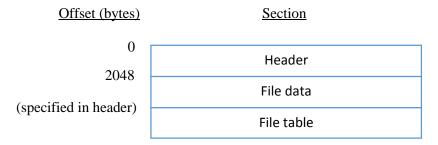


| Offset (bytes) | Section |
|---|---|

**Fig. 1**

As you can see structure of normal PS2 PAK is identical to PC version. The main difference is that all files inside PS2's PAKs must be aligned within CD file system sectors (2048 bytes). Empty space within sector is usually filled with 0x00 bytes.

PAKScape can open PS2's PAKs but if you edit them then game would crash.

## 1.1 Header

Header is identical to PC's PAKs:

```
struct sPS2NormalPAKHeader
{
        char Signature[4];          // "PACK" signature
        ulong TableOffset;          // File table location
        ulong TableSize;            // Size of file table (in bytes)
};
```

However, it should be inside its own 2048-byte sector.

## 1.2 File data

This section contains data of all files that preset in PAK. The main difference with PC's PAKs is that data of each file should be aligned within 2048-byte sectors.

## 1.3 File table

This section is identical to PC version:

```
struct sPS2PAKFileTableEntry
{
        char FileName[56];              // File name
        ulong FileOffset;               // File location
        ulong FileSize;                 // File size (in bytes)
}
```

Sometimes garbage and leftovers are present in FileName field after null terminator, don't be confused with it. Sometimes those leftovers can represent some interesting data from computers of Gearbox (see fig. 2).

```
models/agibs.dol.y for pak creation.........#Output pak na....p€..
models/crossbow_bolt.dol.#Output pak file byte alignmenth€..ħ...
models/grenade.dol.RENAME.....\Pak0\PakContents\models\agi0ĸ.......
```

**Fig. 2 – Leftovers from file table of VALVE.PAK (colored white)**

## 2 Compressed PAKs

Structure of file is shown on fig. 3.

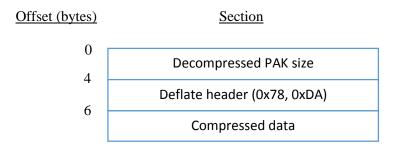Offset (bytes)                          Section



**Fig. 3**

Compressed PAK contains size of decompressed normal PAK and compressed normal PAK itself. However, this normal PAK differs from non-compressed ones by fact that all files inside are aligned within 16-byte sectors instead of 2048-byte.

## 3 PAK handling

There is no WAD files in PS2 version of Half-life. Instead, all map textures are stored inside *.BS2 map files and all decals are stored inside "DECALS.PAK" and "GDECALS.PAK" files.

Unlike the PC version of Half-life there is no support for "PAK1.PAK", "PAK2.PAK" and "PAK3.PAK" files.

All main game resources are stored inside "VALVE.PAK", "DECAY.PAK", "GLOBAL.PAK" ("GRESTORE.PAK") and "PAK0.PAK". Priority of PAKs from highest to lowest:

- "VALVE.PAK"/"DECAY.PAK" – overrides contents of both GLOBAL (GRESTORE) and PAK0
- "GLOBAL.PAK" ("GRESTORE.PAK") – overrides contents of PAK0
- "PAK0.PAK"

"VALVE.PAK", "DECAY.PAK", "GLOBAL.PAK" ("GRESTORE.PAK") are remaining resident in console's RAM during gameplay and acting as cache for frequently accessed files like weapon models, weapon sounds, GUI sprites, etc. This helps to avoid streaming stutters and improves loading times.

"VALVE.PAK" is cache for Half-life and Deathmatch (Head to head) files, while "DECAY.PAK" is cache for Decay files. One of those PAKs is loaded to console's RAM after you start/load game and remains resident through gameplay and mid-game menu. When you go to the main menu, this PAK is pulled out of RAM. Base address of those PAKs in RAM is 0x0B2CF40 (defines location of first byte).

"GLOBAL.PAK" (as you can tell by its name) contains data that is visible from all games: Half-life, Deathmatch and Decay. Base address of GLOBAL.PAK (GRESTORE.PAK) in console's RAM is 0x1F7DFD0 (defines location of last byte). This PAK is loaded to console's RAM before intro popup and remains resident through gameplay. However, each time you go to mid-game menu it is pulled out of RAM. When you are going back to game, "GRESTORE.PAK" is loaded to RAM instead of "GLOBAL.PAK". "GRESTORE.PAK" is a mirror of "GLOBAL.PAK", but with one main difference – all SPZ sprites are patched to be exactly like they are represented in console's RAM:

1) RAMFlag is set to 1
2) Unique FrameID is set for each sprite frame in PAK file (starts from 5 for some reason)
3) FrameOffset is pointing to frame location in console's RAM

Here is how you can calculate those fields:

1) FrameID: set up counter that counts frames from 5 and assign value of this counter to FrameID.
2) FrameOffset = (Base address of GLOBAL.PAK) – (Size of GLOBAL.PAK) + (Sprite file offset) + FrameOffset
   Example. Let's look at "bracket.spz" in GRESTORE.PAK. Sprite file is located at 0xE2B30 inside PAK. Original sprite frame offset is 0x10. PAK size is 0x1C8B90. Then new frame offset is:
   FrameOffset = 0x1F7DFD0 – 0x1C8B90 + 0xE2B30 + 0x10 = 0x1E97F80 (Little Endian: 0x807FE901).

I tried to find out maximum possible sizes of "VALVE.PAK" and "GLOBAL.PAK", here is the result:

| PAK | DATA SIZE | RESULT |
|---|---|---|
| VALVE.PAK | 4 Mb (Original) | OK |
| | 6 Mb | OK |
| | 8 Mb | Noticeable stuttering |
| | 10 Mb | Stuttering |
| | 12 Mb | Heavy stuttering |
| | 13 Mb | Stutters as hell but works |
| | 14 Mb | Straight game crash |
| GLOBAL.PAK | 2 Mb (Original) | OK |
| | 3 Mb | OK |
| | 4 Mb | OK |
| | 5 Mb | Straight game crash |

So the main rules for modifying those PAKs are:

1) Put in those PAKs frequently accessed files to avoid minor stuttering related to streaming and improve loading times.
2) Don't make those PAKs significantly bigger than their original size, otherwise heavy stuttering can occur due to reduced size of PAK0 streaming buffer or game can simply crash.

# 4 List of PAK files and brief description

| PAK name | PAK type | PAK location | PAK description |
|---|---|---|---|
| VALVE.PAK | Compressed | ..\VALVE\ | Contains frequently accessed files of Half-life and Deathmatch |
| PAUSEGUI.PAK | Normal | ..\VALVE\VALVE.PAK\ | Contains descriptors of Half-life and Deathmatch in-game menus |
| PAK0.PAK | Normal | ..\VALVE\ | Contains all main data of Half-life and Deathmatch (maps, models, sounds, sprites, etc.) |
| DECALS.PAK | Normal | ..\VALVE\ | Contains local decals of Half-life and Deathmatch |
| GDECALS.PAK | Compressed | ..\VALVE\ | Contains global decals |
| DECAY.PAK | Compressed | ..\DECAY\ | Contains frequently accessed files of Decay |
| PAUSEGUI.PAK | Normal | ..\DECAY\DECAY.PAK\ | Contains descriptors of Decay in-game menu |
| PAK0.PAK | Normal | ..\DECAY\ | Contains all main data of Decay (maps, models, sounds, sprites, etc.) |
| FRONTEND.PAK | Compressed | ..\GUIDATA\ | Contains images and compressed descriptor of main menu |
| GUISOUND.PAK | Compressed | ..\GUIDATA\ | Contains sounds of GUI (select, enter, etc.) |
| MIDDECAY.PAK | Compressed | ..\GUIDATA\ | Contains images and compressed descriptor of mid-game Decay menu |
| MIDTOH.PAK | Compressed | ..\GUIDATA\ | Contains images and compressed descriptor of mid-game Deathmatch menu |
| MIDHL.PAK | Compressed | ..\GUIDATA\ | Contains images and compressed descriptor of mid-game Half-life menu |
| CGMESAVE.PAK | Compressed | ..\ICONS\ | Contains standard PS2 icons for game saves |
| UGMESAVE.PAK | Normal | ..\ICONS\ | Contains standard PS2 icons for game saves |
| SYSSAVE.PAK | Compressed | ..\ICONS\ | Contains standard PS2 icons for configuration save |
| DICTS.PAK | Normal | ..\PAKS\ | Contains "*.hl1" (save ?) files |
| GLOBAL.PAK | Compressed | ..\PAKS\ | Contains global sounds, sprites, and *.txt sprite descriptors (hud.txt, w_9mmar.txt, etc) |
| GRESTORE.PAK | Compressed | ..\PAKS\ | Mirrors data of GLOBAL.PAK |
| SYSTEM.PAK | Compressed | ..\PAKS\ | Contains fonts and in-game "LOADING" popup image |
| DISCFAIL.PAK | Normal | ..\PAKS\SYSTEM.PAK\ | Contains descriptor of "Disc error" popup |