

=====
SID FILE FORMAT DESCRIPTION
=====

AUTHORS:

Michael Schwendt <sidplay@geocities.com> (PSID v1 and v2)
Simon White <sidplay2@yahoo.com> (PSID v2NG, RSID)
Dag Lem <dag@nimrod.no>(PSID v2NG)
LaLa <LaLa@C64.org> (This document)

INTRODUCTION

=====

This document describes the SID file format used for SID tunes in the HVSC (High Voltage SID Collection - <http://www.hvsc.c64.org>). It is based mostly on Michael Schwendt's excellent document found at:

http://www.geocities.com/SiliconValley/Lakes/5147/sidplay/doc_formats.html

and the PSID v2NG extensions described by Simon White and Dag Lem in:

<http://sidplay2.sourceforge.net>

SID files use the .sid file extension.

Since PSID v2 is simply an extension of PSID v1, PSID v2NG is an extension of PSID v2, and RSID is a restricted version of PSID v2NG, all of the formats are discussed together below. RSID in specific is discussed in detail under the 'magicID' field description.

The information presented here targets programmers, freaks, or other people with reasonable background. It is not suitable for newbies who have never before used a machine code monitor, a disassembler, or a hexadecimal editor.

The SID file header v1

=====

The detailed structure of the SID header looks like the following. Header offsets are in hexadecimal notation. Other integer values are decimal unless explicitly marked otherwise. Any stored integer values are in big-endian format:

+00 magicID: ``PSID'' or ``RSID''

This is a four byte long ASCII character string containing the value 0x50534944 or 0x52534944. 'RSID' (Real SID) denotes that the file strictly requires a true Commodore-64 environment to run properly. 'PSID' files will generally run trouble-free on older PlaySID and libsidplay1 based emulators, too.

Some words about the Real C64 SID file format (RSID):

The RSID format was designed to contain tunes that are not PlaySID compatible, but strictly require a real C64 environment to run. Tunes that are multi-speed and/or contain samples and/or use additional interrupt sources or do busy looping will cause older SID emulators to lock up or play very wrongly (if at all).

By using the name RSID for such rips all existing SID emulators will reject these tunes safely until they can be upgraded to cope with the additional requirements.

Due to the nature of these tunes, every effort must be made to make sure they are directly runnable on an actual C64 computer. As such the tunes will only be presented with the default C64 power-on environment and expected to configure and use all hardware appropriately.

RSID is based on PSIDv2NG with the following modifications:

magicID = RSID
version = only 2
loadAddress = 0 (reserved)
playAddress = 0 (reserved)
speed = 0 (reserved)
psidSpecific flag is called C64BASIC flag

The above fields MUST be checked and if any differ from the above then the tune MUST be rejected. The definitions above will force tunes to contain

proper hardware configuration code and install valid interrupt handlers.

The default C64 environment is as follows:

VIC - IRQ set to raster 0, but not enabled.
CIA 1 timer A - set to 60Hz with the counter running and IRQs active.
Other timers - disabled and loaded with \$FFFF.
Bank register - \$37

A side effect of the bank register is that init MUST NOT be located under a ROM/IO memory area (addresses \$0000-\$07E8, \$A000-\$BFFF and \$D000-\$FFFF). Since every effort needs to be made to run the tune on a real C64 the load address of the image MUST NOT be set lower than \$07E8.

+04 WORD version

Available version number can either be 0001 or 0002. Headers of version 2 provide additional fields. RSID and PSID v2NG files must have 0002 here.

+06 WORD dataOffset

This is the offset from the start of the file to the C64 binary data area. Because of the fixed size of the header, this is either 0x0076 for version 1 and 0x007C for version 2.

+08 WORD loadAddress

The C64 memory location where to put the C64 data. 0 means the data are in original C64 binary file format, i.e. the first two bytes of the data contain the little-endian load address (low byte, high byte). This must always be true for RSID files. Furthermore, the actual load address must NOT be less than \$07E8 in RSID files.

You must be absolutely sure what to enter here. There is no way to detect automatically whether the first two bytes in a C64 data file are meant to be a load address or some arbitrary bytes of code or data. Unless your C64 file is not a normal binary file and thus has no load address in front, you need not enter anything else than 0 here. The SID tune will not play if you specify a load address which is present in the C64 file already.

Normal C64 binary data files have a load address in their first two bytes, so they can be loaded to a pre-defined destination address by executing LOAD"FILE",8,1, for instance. If a load address is explicitly specified in the sidtune info file, some sidtune converters and utilities conjecture that the C64 data don't have a load address in their first two bytes. Hence, the explicit load address from the info file is moved in front of the C64 data to create a valid C64 binary file which can be easily loaded on a C64, too. If that C64 file were to be saved, it would contain two superfluous data bytes at offset 2 if an original load address had been in the first two bytes of the old file. This process of adding a duplicate load address can be repeated. The file loader strips off the first two bytes (the used load address) and puts the rest of the file contents (including the now obsolete load address at file offset 2) into memory. If the new load address is the same than the old one the two added bytes cause the whole data to be displaced by two bytes, which most likely results in malfunctioning code. Also, superfluous bytes in memory then can confuse disassemblers which start at the beginning of the file or memory buffer.

+0A WORD initAddress

The start address of the machine code subroutine that initializes a song, accepting the contents of the 8-bit 6510 Accumulator as the song number parameter. 0 means the address is equal to the effective load address.

In RSID files initAddress must never point to a ROM area (\$A000-\$BFFF or \$D000-\$FFFF) or be lower than \$07E8. Also, if the C64 BASIC flag is set, initAddress must be 0.

+0C WORD playAddress

The start address of the machine code subroutine that can be called frequently to produce a continuous sound. 0 means the initialization subroutine is expected to install an interrupt handler, which then calls the music player at some place. This must always be true for RSID files.

+0E WORD songs

The number of songs (or sound effects) that can be initialized by calling the init address. The minimum is 1. The maximum is 256.

+10 WORD startSong

The song number to be played by default. This value is optional. It often specifies the first song you would hear upon starting the program is has been taken from. It has a default of 1.

+12 LONGWORD speed

This is a 32 bit big endian number. Each bit in 'speed' specifies the speed for the corresponding tune number, i.e. bit 0 specifies the speed for tune 1. If there are more than 32 tunes, the speed specified for tune 32 is also used for all higher numbered tunes.

A 0 bit specifies vertical blank interrupt (50Hz PAL, 60Hz NTSC), and a 1 bit specifies CIA 1 timer interrupt (default 60Hz).

Surplus bits in 'speed' should be set to 0.

For RSID files 'speed' must always be set to 0.

Note that if 'play' = 0, the bits in 'speed' should still be set for backwards compatibility with older SID players. New SID players running in a C64 environment will ignore the speed bits in this case.

WARNING: This field does not work in PlaySID for Amiga like it was intended, therefore the above is a redefinition of the original 'speed' field in SID v2NG! See also the 'clock' (video standard) field described below for 'flags'.

+16 ``<title>''
+36 ``<author>''
+56 ``<released>'' (also known as ``<copyright>'')

These are 32 byte long zero terminated ISO 8859-1 ASCII character strings. Upon evaluating the header, a zero byte will always be put into the last byte of each string. So the maximum number of available free characters is 31.

+76 <data>

Version 1 of the SID header is complete at this point. The binary C64 data starts here.

The SID file header v2
=====

Version 2 of the header incorporates the v1 header fields and provides additional fields. Some of these are actually v2NG specific - those are noted below.

+76 WORD flags

This is a 16 bit big endian number containing the following bitfields:

- Bit 0 specifies format of the binary data (musPlayer):
 0 = built-in music player,
 1 = Compute!'s Sidplayer MUS data, music player must be merged.

If this bit is set, the appended binary data are in Compute!'s Sidplayer MUS format, and does not contain a built-in music player. An external player machine code must be merged to replay such a sidtune.

- Bit 1 specifies whether the tune is PlaySID specific, e.g. uses PlaySID samples (psidSpecific):
 0 = C64 compatible,
 1 = PlaySID specific (PSID v2NG)
 1 = C64 BASIC flag (RSID)

This is a v2NG and RSID specific field.

PlaySID samples were invented to facilitate playback of C64 volume register samples with the original Amiga PlaySID software. PlaySID samples made samples a reality on slow Amiga hardware with a player that was updated only once a frame.

Unfortunately, converting C64 volume samples to PlaySID samples means that they can no longer be played on a C64, and furthermore the conversion might potentially break the non-sample part of a tune if the timing between writes to the SID registers is at all altered. This follows from the ADSR bugs in the SID chip.

Today, the speed of common hardware and the sophistication of the SID players is such that there is little need for PlaySID samples. However, with all the

PlaySID sample PSIDs in existence there's a need to differentiate between SID files containing only original C64 code and PSID files containing PlaySID samples or having other PlaySID specific issues. As stated above, bit 1 in 'flags' is reserved for this purpose.

Since RSID files do not have the need for PlaySID samples, this flag is used for a different purpose: tunes that include a BASIC executable portion will be played (with the BASIC portion executed) if the C64 BASIC flag is set. At the same time, initAddress must be 0.

- Bits 2-3 specify the video standard (clock):
 - 00 = Unknown,
 - 01 = PAL,
 - 10 = NTSC,
 - 11 = PAL and NTSC.

This is a v2NG specific field.

As can be seen from the 'speed' field, it is not possible to specify NTSC C64 playback. This is unfortunate, since the different clock speeds means that a tune written for the NTSC C64 will be slightly detuned if played back on a PAL C64. Furthermore, NTSC C64 tunes driven by a vertical blank interrupt have to be converted to use the CIA 1 timer to fit into this scheme. This can cause severe problems, as the NTSC refresh rate is once every 17045 cycles, while the CIA 1 timer A is latched with 17095 cycles. Apart from the difference in timing itself, the SID ADSR bugs can actually break the tune.

The 'clock' (video standard) field was introduced to circumvent this problem.

- Bits 4-5 specify the SID version (sidModel):
 - 00 = Unknown,
 - 01 = MOS6581,
 - 10 = MOS8580,
 - 11 = MOS6581 and MOS8580.

This is a v2NG specific field.

The MOS6581 and the MOS8580 have three notable differences. First, combined waveforms are generally louder on a MOS8580, to the extent that some combinations that are clearly audible on a MOS8580 are completely silent on a MOS6581. Second, the internal DC levels in the MOS8580 are so small that software or hardware tricks must be used to play volume samples. Third, the MOS8580 analog filter has totally different characteristics from the MOS6581 analog filter.

To ensure that music specifically written for one of the two SID versions can be played back correctly, bits 4-5 in 'flags' are used as stated above.

- Bits 6-15 are reserved and should be set to 0.

+78 BYTE startPage (relocStartPage)

This is a v2NG specific field.

This is an 8 bit number. If 'startPage' is 0, the SID file is clean, i.e. it does not write outside its data range within the driver ranges. In this case the largest free memory range can be determined from the start address and the data length of the SID binary data. If 'startPage' is 0xFF, there is not even a single free page, and driver relocation is impossible. Otherwise, 'startPage' specifies the start page of the single largest free memory range within the driver ranges. For example, if 'startPage' is 0x1E, this free memory range starts at \$1E00.

+79 BYTE pageLength (relocPages)

This is a v2NG specific field.

This is an 8 bit number indicating the number of free pages after 'startPage'. If 'startPage' is not 0 or 0xFF, 'pageLength' is set to the number of free pages starting at 'startPage'. If 'startPage' is 0 or 0xFF, 'pageLength' must be set to 0.

The relocation range indicated by 'startPage' and 'pageLength' should never overlap or encompass the load range of the C64 data. For RSID files, the relocation range should also not overlap or encompass any of the ROM areas (\$A000-\$BFFF and \$D000-\$FFFF) or the reserved memory area (\$0000-\$03FF).

+7A WORD reserved

This is a 16 bit number and is reserved and should be set to 0.

+7C <data>

Version 2 of the SID header ends here. This offset is the start of the binary C64 data. See also 'loadAddress' for what the first 2 bytes of 'data' might indicate.