

Creating Chip Tunes with
SID-WIZARD

Mihaly Horvath & Mikael Norrgård

Creating Chip Tunes with
SID-WIZARD

FIRST EDITION

written by

Mikael Norrgård (WitchMaster)

and

Mihaly Horvath (Hermit)¹

This is version 1.0 of the document

11. November 2012

¹ The creator of SID-Wizard

1. INTRODUCTION	5
1.1 WHAT IS SID-WIZARD?	5
1.2 FEATURES OF SID-WIZARD	6
NEW FEATURES/IMPROVEMENTS IN SID-WIZARD VERSION 1.2	8
PLAYER COMPARISONS (LIGHT / MEDIUM / FULL / EXTRA)	9
1.3 PRINCIPLE OF SOUND SYNTHESIS.....	9
1.4 THE SID (6581/8580) CHIP	10
1.5 HEXADECIMAL NUMERAL SYSTEM.....	13
NIBBLE	13
1.6 A WORD ABOUT KEYBOARD SHORTCUTS.....	14
1.7 RUNNING SID-WIZARD ON A REAL COMMODORE 64	14
X1541 CABLE + SOFTWARE	14
SD2IEC (μ IEC / MMC2IEC)	14
1541 ULTIMATE.....	15
TURBO CHAMELEON 64	15
1.7 USING SID-WIZARD IN VICE (COMMODORE 64 EMULATOR).....	16
DOWNLOADS.....	16
SETTING UP VICE	16
MAKING EVERYTHING SIMPLE (USING .PRG FILE).....	17
MAKING EVERYTHING SIMPLE (USING .D64 IMAGES)	17
2. USER INTERFACE	18
2.1 STARTUP-MENU	18
2.2 USER INTERFACE OVERVIEW	19
INFO AREA	19
PATTERN EDITOR.....	20
ORDER LIST	20
INSTRUMENT EDITOR	20
CHORD TABLE	20
TEMPO-PROGRAM TABLE	20
SUBTUNE AND TEMPO	20
3. INSTRUMENT EDITOR.....	21
3.1 WAVEFORMS.....	21
WF-COLUMN	22
WAVEFORM / CONTROL FIRST NIBBLE.....	22
WAVEFORM / CONTROL SECOND NIBBLE	23
ARP-COLUMN.....	23
DT-COLUMN (DETUNE)	23
3.2 PULSE WIDTH	24
PULSE VALUES (TWO FIRST COLUMNS).....	25
3.3 ADSR ENVELOPE.....	25
3.4 VIBRATO	26
3.5 FILTER.....	27
3.6 OTHER INSTRUMENT PARAMETERS.....	29
1ST FRAME'S WAVEFORM-REGISTER	29

HARD-RESTART ADSR.....	29
HARD-RESTART TIMER / TYPE.....	30
DEFAULT ARPEGGIO TABLE SPEED	30
DEFAULT CHORD	30
OCTAVE-TRANPOSE AMOUNT	31
INSTRUMENT NAME.....	31
3.7 SAVING AND LOADING INSTRUMENTS	31
5. INSTRUMENT EXAMPLES	32
5.1 BAZZ.....	32
5.2 SNARE.....	33
5.3 ARPEGGIO.....	34
6. PATTERN EDITOR.....	35
6.1 NOTE-COLUMN EFFECTS	37
6.2 INSTRUMENT-COLUMN EFFECTS.....	38
6.3 EFFECT-COLUMN SMALL EFFECTS	39
6.4 EFFECT-COLUMN BIG EFFECTS	40
6.5 PATTERN EXAMPLE.....	41
7. ORDERLIST.....	42
7.1 ORDERLIST EFFECTS	43
8. TEMPOS AND MULTISPEED	44
8.1 TEMPO-PROGRAM TABLE.....	44
8.2 MULTISPEED	45
9. MAIN MENU AND FILE OPERATIONS	46
9.1 SIMPLE SETTINGS.....	46
9.2 LOAD/SAVE TUNE (WORKFILE) OR INSTRUMENT	48
USABLE KEYS IN THE FILE-DIALOG.....	49
THE SWM FILE FORMAT	49
10. SID-MAKER (EXPORT YOUR TUNES)	49
10.1 CONVERTING .S00 FILE TO .SID WITH SWM-CONVERT.....	51
10.2 EDITING METADATA IN YOUR SID-FILE.....	52
11. REFERENCE TABLES	52
11.2 PATTERN EDITOR RELATED KEYS	54
11.3 ORDERLIST RELATED KEYS	55
11.4 INSTRUMENT EDITOR RELATED KEYS.....	56
11.5 CHORD TABLE / TEMPO TABLE RELATED KEYS	56
11.6 DECIMAL TO HEXADECIMAL CONVERSION TABLE.....	56
11.7 EXACT NOTES FOR ARP-COLUMN	58
11.8 ATTACK / DECAY / RELEASE TIMINGS.....	58

1. INTRODUCTION

Welcome to this expanded manual for the excellent music tracker SID-Wizard² for Commodore 64. I will try to go into detail of most aspects of the tracker and I will also try to explain the functions of the wonderful SID chip as we go along. The reason why I expanded the work of Mihaly's manual is that there is some stuff left out that SID gurus know but beginners don't. I will probably expand this manual later, please don't hesitate to give comments about anything that you think is unclear or could be explained in a better way. My email address is *mikael.norrgard@gmail.com* and my CSDB user name should soon hopefully be *witchmaster*. Possible new editions of this e-book will be released on my webpage: <http://www.witchmastercreations.com>.

The next two sections (1.1 and 1.2) are entirely written by Mihaly Horvath (the creator of SID-Wizard), taken from the SID-Wizard Manual 1.2. Other sections are also a mix of my writings and that of Mihaly, some sections are more or less identical to the SID-Wizard Manual 1.2. I have been mailing questions to Mihaly about features I didn't really understand, so I give him thanks here for his patience and help.

If you are not used to SID trackers you shouldn't get stuck on the feature list of SID-Wizard, you will start to understand all of this in time.

1.1 WHAT IS SID-WIZARD?³

Hey folks, I'm Hermit! If you wonder what SID-Wizard is, I have a short answer: yet another tracker application for the famous Commodore 64 personal computer of the eighties. More specifically, a music editor for its sound chip called SID (Sound Interface Device). As you may know, there is still an evolving culture and technical background behind this machine - that's why demo musicians are still welcome to join the scene and compose wonderful SID tunes. SID-Wizard is an open-source project from the very 1st release, and everyone is welcome to help in further developments of my extensively commented source-code.

² SID-Wizard version 1.2 at the time of writing.

³ This section is taken from the manual of SID-Wizard 1.2

My aim from the beginning was to create a comprehensive native C64 tool for SID music creation, as there are so many editors around which are good at some aspects but on the other hand many of them lack very basic features. (For instance, saving/loading individual instruments, multi-speed support, detuning, jamming, keyboard-tracking, etc.)

However there are several trackers to mention which contain almost all important features, and which I got much inspiration from. My personal favourites - and possibly the most featureful, in order of appearance - are: Goatracker (cross-platform with emulated SID sound), SDI, X-SID, SID-Factory, JCH Editor, DMC, Hardtrack Composer. I made some comparison charts about these editors, and I figured out that I need to code a native C64 based editor, which should contain everything essential, and even some more inventive/innovative features. More or less I found the way to do so, and coupled the different features under common hood to achieve simplicity, ease of use despite the amount of new functions. Let's see in a nutshell in the next section what can be done with SID-Wizard....

1.2 FEATURES OF SID-WIZARD⁴

- 50 instruments, 100 of 250-byte long patterns and 16 (\$0..\$F) of subtunes in total, may vary later
- Optimized rastertime (around \$1A, below \$20) and small code size to let more room for musical data (editor code and graphics are ~12kbyte together, player-code is around -or a little more than- 2kbyte)
- Virtually no zeropage usage, so the music routine won't interfere with other parts of your program
- 3 x 25 rows pattern-editing perspective to keep as much as possible musical data in screen-focus.
- Goatracker-like pattern-editing concept and keyboard control to ease migration from Goatracker.
- Everything on one screen (except menu) to ease composition workflow. Extensive use of border.
- Polyphonic jam-mode with dovetailing (cycles through voice 1..3) - good for sounds with long release

⁴ This section is taken from the manual of SID-Wizard 1.2

- Realtime compression of pattern-data - zeroes don't appear in memory, therefore more memory is left for informative data (similar to XM-patterndata compression, but more effective).
- 'Jump to other subtune' command in orderlists/sequences. Tune can be written in subtune-chunks then be put together via sequence-copy/paste functions in orderlists.
- All instruments have independent tables and names- no need for table-pointers, instruments can be loaded/saved independently
- All instruments have own Hard-restart ADSR, Hard-restart timing, and 1st-waveform settings
- Per-row based detuning and keyboard-tracking in arpeggio, pulsewidth, filter instrument-tables
- Up to 8x (400Hz) multi-framespeed not affecting speed values and tempo (just like in Jeff's X-SID)
- Advanced partial copy/paste and auto-advance amount, page-up/page-down in patterns/orderlists
- Independent track-tempos / funktempos / tempo-programs and independent pattern-scrolling possible
- 4 vibrato types to simulate impression expressions of various instruments like violin/guitar/etc.
- 3 simultaneous pattern effects are possible at the same pattern row (note-FX,instrument-FX,etc.)
- Calculated vibrato and slide/portamento speed achieve evenly audible pitch-changes through octaves
- Oscilloscope and Pulsewidth/Cutoff-frequency displayer bars to visualize waveforms and sweeps
- Gate-off event triggered instrument-table pointers (instrument's release phase can be controlled)
- Basic editor settings are saved into workfile (framespeed, colour-theme etc.:`templates`)
- There is concept and separate table for CHORDS that can be called from arpeggio-table of instruments or from pattern-FX, therefore No need to create new instruments for different chords!
- Compression of empty places in note-column when saving tune (called 'packed rest' / `packed NOP`)
- Save/export tune in SID format, runnable PRG format (beside simple PRG and BIN formats)
- Export/import to/from XM/MID format with `SWMconvert` to ease cooperation with x86 tools

NEW FEATURES/IMPROVEMENTS IN SID-WIZARD VERSION 1.2

- NTSC machine support - detects machine-type and auto-sets graphics and frequency-table
- 1st frame's waveform-register now can be set to any values (except 0 in SWM1) for all instruments
- More configuration options in menu (they are saved to the workfile):
 - Selectable (predefined) colour-themes
 - Flickering rasterbars can now be hidden
 - Instrument-autotyping (unneeded repetitive instrument-selections optimized when tune saved)
 - GT/ DMC piano-keyboard layout, 2 row of notes (ASDF...), octave selection with `1..8` keys
 - follow-playback mode can be set as default - and can be toggled in realtime too (see keys' section)
- Rastertime of the tunes during playback is measured and peak/max rastertime is calculated
- Copy/Paste (even partial) now in Orderlist-sequences too, and Finding empty patterns (C= + E).
- Fast-forward playback (4x the speed of normal) now possible (with `left-arrow` key)
- Built-in cheat sheet (instant help) for the most important keys (the last menupoint)
- Values can be negated with `=` key in instrument-tables and chordtables (e.g. \$40 will be \$C0)
- Pressing Return/Enter on the instrument's default chord-number brings the chord in focus
- SID-Maker improvements: author-info moved to reused memory, faster relocation-address setting, `exe.prg` export can switch subtunes and displays playtime and rastertime-use (current & peak)
- some other fixes : table-cursorpositions remembered , edit/jam mode better distinguishable, muted track's filter-program is stopped, C+=Delete deletes in note-column if cursor is there, extended relocation range (\$0200...\$ffff), normal SID output for single-speed tunes
- Startup menu with selectable players: normal/ light/medium/extra (selectable in SID-Maker too). The purpose of light/medium-versions is to reduce player-code size and rastertime-consumption. The extra version has some more rastertime-consumption and memory-usage but better quality well suited for standalone music releases... All versions use the same SWM1 module format. but slightly different SID sound engine (i.e. player/driver-routine).

PLAYER COMPARISONS (LIGHT / MEDIUM / FULL / EXTRA)

Feature (in `source/settings.cfg`, alter & compile for particular needs.)	Light	Medium	Full	Extra
Calc.vibrato, Detune, Chord-table, Transpose, instr.Octave, WF-arp. speed		X	X	X
PW/filtetable-reset off, filter keyboard-track, 11bit filter, tempo-program		X	X	X
Vibrato-types, Hard-restart types, Frame1 \$09 waveform switch			X	X
Pulsewidth keyboard-tracking, Note-off table-pointers			X	X
Subtune-jump FX (now independent on tracks), Saving/Restoring zeropage			X	X
Filter/Pulsewidth/WF-program/slides never skipped, filt-ex.FX, Ghost-reg.				X
Tempo (0..2) full support, vibrato returns after portamento, delay note/track				X

1.3 PRINCIPLE OF SOUND SYNTHESIS⁵

There are basically two types of analog frequency synthesis: Additive and Subtractive. Additive synthesis tries to simulate the sound of instruments by adding a limited number of harmonics to the base frequency of a musical note. Subtractive method is based upon oscillators which can generate some kinds of waveforms (sinus, triangle, sawtooth, pulse/square, noise, etc.) with inherently rich spectral harmonic content and they can be mixed and filtered to be shaped to a desired timbre.

We can alter many other parameters of the sound, the most important is to control its volume-envelope (ADSR) in order to be able to model live instruments to a degree. Possibilities are endless compared to sample-based synthesis, only the constraints of hardware and software limit the number of variations a little bit.

I'm not going to say more about the sound synthesis at this point, many readings can be found in cyberspace.

⁵ This section is taken from the manual of SID-Wizard 1.2

1.4 THE SID (6581/8580) CHIP

"The 6581 Sound Interface Device (SID) is a single-chip, 3-voice electronic music synthesizer/sound effects generator compatible with the 65XX and similar microprocessor families. SID provides wide-range, high-resolution control of pitch (frequency), tone color (harmonic content) and dynamics (volume). Specialized control circuitry minimizes software overhead, facilitating use in arcade/home video games and low-cost musical instruments."

From the Commodore SID 6581 Datasheet

The MOS technology SID 6581 or 8580 sound chip in C64 is a subtractive sound-synthesis chip designed by Bob Yannes in 1982. It is one of the most important reasons for the C64's popularity. There are many differences between different revisions and types of SID chips produced over the years, maybe due to the shortened development time, and the specification of the chip was somewhat unfinished. The most significant change/improvement has appeared in the 8580 new SID model over the original 6581 SID, but many consider the 6581 better. 8580 has more balanced filter-cutoff frequency control curve and less-distorted resonance, better mixed-waveform support, and maybe more predictable ADSR envelope-handling. It's still a mystery in some aspects what the chip exactly does inside, however there were thorough reverse-engineering projects recently both in software and hardware emulation areas (e.g Resid dll, 1541 Ultimate VHDL code for SID). Even Bob Yannes himself doesn't keep track of the long lifetime of the SID chip and what people do with it.

As stated in the citation above, the SID has a three voice polyphony (three sounds can play at one time). These sounds can be generated by combining waveforms and applying effects like modifying the pulse width of the pulse waveform, using a filter (low-pass / high-pass / band-pass), applying ring modulation or hard sync and adding vibrato. Another common way of creating more complex sounds (for example drums) is to use a wave table, i.e. successively changing between different waveforms in a rapid manner. Arpeggios can also be created using a wave table.

The SID chip can generate three different waveforms (triangle, sawtooth and pulse) that can be combined or used on their own. There is also a fourth waveform, noise, which can't be combined with the other waveforms. The SID 6581 has a bug (or feature) that results in low volume when waveforms are combined, the SID 8580 can combine the waveforms more effectively.

The explanations below are taken from the SID 6581 datasheet. The functionality of Hard Sync and Ring Modulation are described according to voice 1. With both these effects it's the voice to the left that affects these effects (voice 1 => voice 2, voice 2 => voice 3 and voice 3 => voice 1).

WAVEFORM: TRIANGLE

The Triangle waveform is low in harmonics and has a mellow, flute-like quality.

WAVEFORM: SAWTOOTH

The sawtooth waveform is rich in even and odd harmonics and has a bright, brassy quality.

WAVEFORM: PULSE

The harmonic content of this waveform can be adjusted by the pulse width registers, producing tone qualities ranging from a bright, hollow square wave to a nasal, reedy pulse. Sweeping the pulse width in real-time produces a dynamic "phasing" effect which adds a sense of motion to the sound. Rapidly jumping between different pulse widths can produce interesting harmonic sequences.

WAVEFORM: NOISE

This output is a random signal which changes at the frequency of the oscillator. The sound quality can be varied from a low rumbling to hissing white noise via the oscillator frequency registers. Noise is useful in creating explosions, gunshots, jet engines, wind, surf and other un-pitched sounds, as well as snare drums and cymbals. Sweeping the oscillator frequency with noise selected produces a dramatic rushing effect.

HARD SYNC

This synchronizes the fundamental frequency of oscillator 1 with the fundamental frequency of oscillator 3, producing "Hard Sync" effects. Varying the frequency of oscillator 1 with respect to oscillator 3 produces a wide range of complex harmonic structures from voice 1 at the frequency of oscillator 3. In order for sync to occur oscillator 3 must be set to some frequency other than zero but preferably lower than the frequency of oscillator 1. No other parameters of voice 3 have any effect on sync.

RING MODULATION

Ring modulation replaces the triangle waveform output of oscillator 1 with a "ring modulated" combination of oscillators 1 and 3. Varying the frequency of oscillator 1 with respect to oscillator 3 produces a wide range of non-harmonic overtone structures for creating bell or gong sounds and for special effects. In order for ring modulation to be audible, the triangle waveform of oscillator 1 must be selected and oscillator 3 must be set to some frequency other than zero. No other parameters of voice 3 have any effect on ring modulation.

FILTER

The SID chip has only one filter that must be shared for all three voices. The filter types supported are low-pass, high-pass and band-pass. These types can also be combined. The cutoff frequency and resonance strength can be changed.

1.5 HEXADECIMAL NUMERAL SYSTEM

"In mathematics and computer science, hexadecimal (also base 16, or hex) is a positional numeral system with a radix, or base, of 16. It uses sixteen distinct symbols, most often the symbols 0–9 to represent values zero to nine, and A, B, C, D, E, F (or alternatively a–f) to represent values ten to fifteen."

<http://en.wikipedia.org/wiki/Hexadecimal>

You *should* read elsewhere about the hexadecimal numeral system to get a basic understanding of it. For the lazy ones I have included a conversion table in the end of this manual for converting between decimal and hexadecimal values. Both signed and unsigned values are covered in the table for one byte values. This gives the table a range from 0-255 for unsigned values and -128-127 for signed values. Most values in SID-Wizard don't go beyond this scope with the exception of setting the pulse width, which is a 12-bit number (\$000...\$FFF).

There *IS* a good feature in SID Wizard that helps a lot with signed values. The feature negates values for you automatically, so if you want the value -30 you write the positive value in hex format (\$1E) and press the key '=' to negate the value to -30 (\$E2).

HEX numbers are written with a dollar sign before the value, so for example \$A in hex equals 10 in the normal decimal system.

NIBBLE

We will use the word nibble in the coming chapters. A nibble is a four-bit aggregation, or half an octet. As a nibble contains 4 bits, there are sixteen possible values, so a nibble corresponds to a single hexadecimal digit. A full byte (octet) is represented by two hexadecimal digits; therefore, it is common to display a byte of information as two nibbles. Take the hex number \$4F, the first nibble is 4 and the second is F.

1.6 A WORD ABOUT KEYBOARD SHORTCUTS

The shortcuts keys mentioned in this article are for the C64 keyboard. If the key is not the same in the C64 emulator (Vice), then that keyboard shortcut will be shown in parentheses.

1.7 RUNNING SID-WIZARD ON A REAL COMMODORE 64

If you have a Commodore 64 put away in the cabinet, now is the time to take it back into use. Although you can run SID-Wizard in a C64 emulator and the emulation of the SID chip nowadays is ok, the sound quality (e.g. warmth and filter distortion) of the real thing is still unbeatable. There will also be practically no delay (latency) when jamming.

There are quite a few options of getting SID-Wizard transferred to your Commodore 64, I will list a few of them here. For a comprehensive list check this link out:

http://www.c64-wiki.com/index.php/Data_transfer_hardware.

X1541 CABLE + SOFTWARE

This is a series of cables that makes it possible to connect a C64 Floppy disk drive to PC and transfer files to and from floppy disks. You will need software like Star Commander or OpenCBM to use these cables. There are different versions of this cable (X1541, XA1541, XM1541, XMP1541). You will need a parallel port to use these cables. There is even a USB version of the cable (XU1541), I'm not sure how well that works though. Make sure that your version of the cable works with the software you want to use.

Cost: ~10-20€.

Requires assembling: Yes, if you don't buy a ready made cable.

SD2IEC (μIEC / MMC2IEC)

This is a hardware mass storage device using SD a memory card interfacing with the IEC bus. This means that SD2IEC is an emulation / replacement of a Commodore-1541 disk drive for a C64. You can transfer .D64 images to the SD card and use them as normal floppy disks. This is a really practical, you can then even work with your tunes on a proper

Commodore 64 at home and use your .D64 with your emulator (on the laptop on the go).

Cost: ~50€

Requires assembling: Maybe a little bit.

1541 ULTIMATE

The 1541 Ultimate is a storage solution for the Commodore 64 (and other models). There are two versions of this cartridge (I and II). The latter even supports USB pen drives.

Cost: ~130€

Requires assembling: No

TURBO CHAMELEON 64

Turbo Chameleon 64 is a user-friendly multipurpose cartridge for the Commodore 64. It has a MMC/SD card slot with MMC64 compatibility and 1541-emulation plus a lot of other features. The Turbo Chameleon 64 cartridge can even be operated as a stand-alone unit (since it's basically an FPGA computer like the C-One or the Minimig⁶ in a C64 cartridge case, which is really cool).

Cost: ~200€

Requires assembling: No

⁶ <http://en.wikipedia.org/wiki/Minimig>

1.7 USING SID-WIZARD IN VICE (COMMODORE 64 EMULATOR)

I will here explain how to use SID-Wizard in the emulator Vice, if you are not lucky enough to own a Commodore 64. There will be quite some sound latency when jamming when you use the emulator, compared to the real thing. These next steps are for Windows but I imagine this will work similarly in Linux or Mac OS X.

DOWNLOADS

1. Download and install Vice64:

<http://www.viceteam.org/#download>

2. Download SID-Wizard here:

<http://noname.c64.org/csdb/release/?id=112716>

or here:

<http://sourceforge.net/projects/sid-wizard/>

You can download the (SID-Wizard 1.2 disk.d64) or you can download the prg-files (SID-Wizard and SID-Maker) if you are going to use seamless folder integration⁷ in Vice, which is the recommended method and will be used here⁸.

SETTING UP VICE

1. Start Vice.
2. Activate Options / Double Size.
3. Make sure that Options / Warp Mode is inactivated and Sound playback is activated.
4. Go to Settings / SID settings... and choose the sid chip to emulate. ReSid gives you the most accurate emulation but it also requires more processing power. Which SID chip you choose to emulate is a matter of taste, as a guide, 8580 has more effective filters but a thinner sound than the 6581. For good sound quality in high pitches and thin sounds, you should also change sample method to *interpolating*. Remember that your music will sound different on different versions of the SID chip.

⁷ When loading a prg-file directly in Vice, the folder of the prg-file will be used for saving, which is very practical.

⁸ When using seamless folder intergration you need to use SWMConvert to convert .S00 files to .SID, this tool will work in Linux and Windows, so for other operations systems you should use D64-images and extract the sid-files from there.

5. This step is optional but will give the user interface a more authentic look, it will make the graphics and characters look better (the user interface is designed to utilize the smoothing effect of PAL-blur); Go to Options and activate PAL Emulation. :)
6. Click Settings / Save current settings.

MAKING EVERYTHING SIMPLE (USING .PRG FILE)

1. Create a folder for SID-Wizard and copy the .prg files there (tunes and instruments will also be saved to this folder).
2. Create a shortcut for Vice to the desktop (or wherever you want one).
3. Change the target of the shortcut (right-click and choose properties) to this:

"Path to x64.exe" -autostart "Path to SID-Wizard-1.2.prg"

This will start SID-Wizard automatically and seamless folder integration will be used, so your tunes and instruments will be saved to the same directory as SID-Wizard-1.2.prg.

4. Rename the shortcut to SID-Wizard.
5. Running Vice from the shortcut now should take you into SID-Wizard automatically. You can go to and from full screen mode with ALT + D.

MAKING EVERYTHING SIMPLE (USING .D64 IMAGES)

1. Go to Settings / Drive settings... / Drive 9 and choose drive type 1541.
2. Click Settings / Save current settings.
3. Go to File / Attach disk image / Drive 8 (or 9) and create a blank disk (for saving your music and instruments).
4. Create a shortcut for Vice to the desktop (or wherever you want one).
5. Change the target of the shortcut (right-click and choose properties) to this:

"Path to x64.exe" -autostart "Path to SID-Wizard 1.2 disk.d64:sid-wizard-1.2" -9 "Path to savedisk.d64" -attach9rw

This will start SID Wizard automatically and mount your save disk to drive 9. Notice that you can specify which prg to start in the disk image by writing a colon and the name of the prg after the disk image path (as in the case above, sid-wizard).

6. Rename the shortcut to SID Wizard.
7. Running Vice from the shortcut now should take you into SID Wizard automatically. You can go to and from full screen mode with ALT + D.

2. USER INTERFACE

2.1 STARTUP-MENU

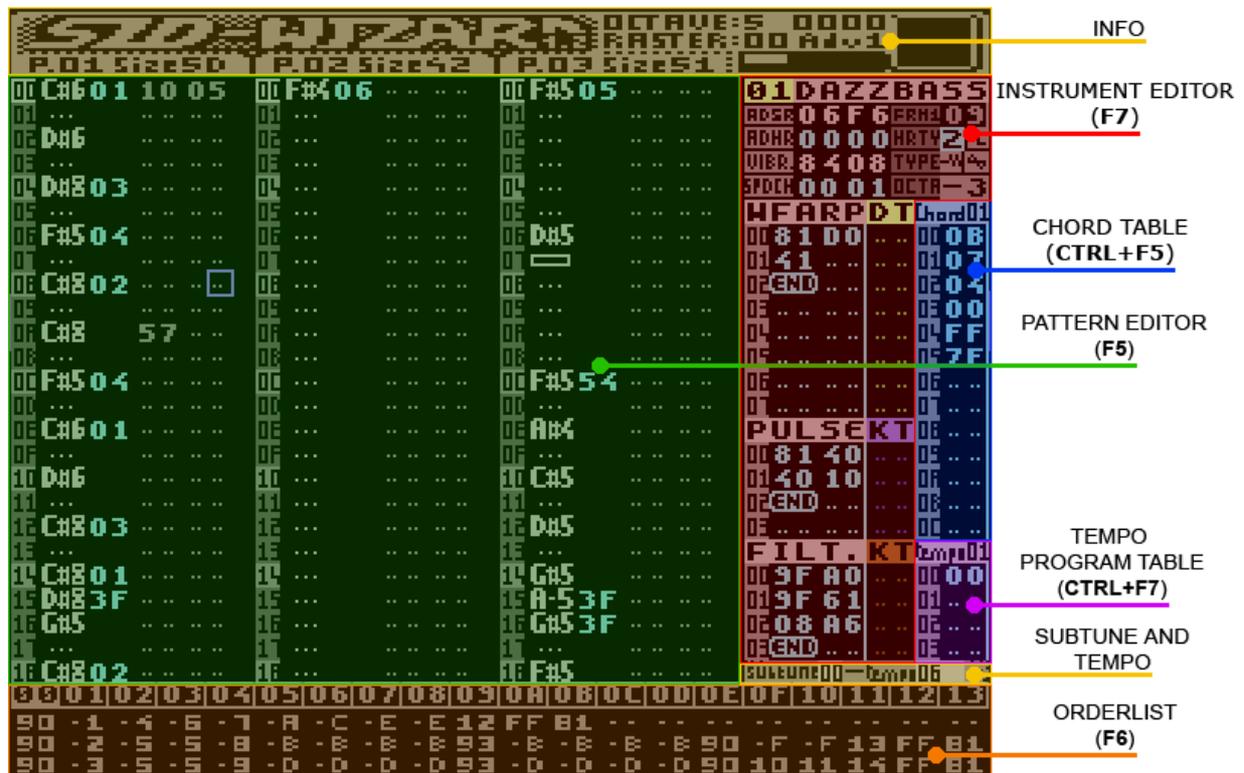
When you start SID-Wizard you will first see the startup-menu. Here you can select your preferred player. The purpose of light / medium-versions is to reduce player-code size and rastertime-consumption. The extra version has some more rastertime-consumption and memory-usage but better quality well suited for standalone music releases. All versions use the same SWM1 module format. but slightly different SID sound engine (i.e. player/driver-routine). You can see the features for each player when highlighting a menu point. The disabled features / tables for the chosen player version will appear as dark-grey in the editor after start-up.



Picture 1: SID-Wizard 1.2 startup-menu

2.2 USER INTERFACE OVERVIEW

The user interface is very intuitive and everything is visible on one screen, there are seven main areas. Picture 2 below shows these areas and the shortcut keys for getting there. You move around the user interface with the keyboard only and most parameters and values are given in the hexadecimal numeral format (read about this in the introduction). For a complete list of keyboard shortcuts in SID-Wizard, refer to the keyboard shortcut tables at the end of the manual.



Picture 2: Main screen overview

I will introduce these areas shortly now, the following chapters will explain these more thoroughly.

INFO AREA

The info area contains the logo, information about selected patterns, jamming octave, rastertime, playback time, auto-advance mode / amount, pulse width meter, filter cutoff meter and oscilloscope.

PATTERN EDITOR

SHORTCUT KEY: F5

This is where you write the actual music. You can edit three different patterns at one time, one for each of the three channels of the SID chip.

ORDER LIST

SHORTCUT KEY: F6

This is where you control the whole arrangement of your tune, i.e. the order of the patterns etc.

INSTRUMENT EDITOR

SHORTCUT KEY: F7

Here you create the instruments for your tunes. This area consists of four sections; the main area, waveforms-arpeggio table (WFARP), pulse table (PULSE) and filter table (FILT.)

CHORD TABLE

SHORTCUT KEY: C+=F5 (CTRL+F5)

In this table you can edit and store chords, you can choose which chord is default for an instrument, the chord used can be changed in the pattern editor. Storing the chords separately from the instruments gives you more flexibility (you don't have to create different instruments for different chords).

TEMPO-PROGRAM TABLE

SHORTCUT KEY: C+=F7 (CTRL+F7)

This table is only used if you need complex tempos, see the section about tempos for more information.

SUBTUNE AND TEMPO

This area shows the active sub tune and the default tune-tempo / funktempo.

3. INSTRUMENT EDITOR

The instrument editor consists of four different sections. The first section contains common parameters for the instrument, the second section is for editing waveforms and arpeggios, the third is the pulse width table and the fourth is the filter table. You can change instrument with the + and - keys. You should also remember that you can change instrument with SHIFT + '+' and SHIFT + '-' anywhere on the user interface, even in the chord-/ tempo-program tables. You can move between these sections with the CONTROL (TAB) key, to move backwards use SHIFT+CONTROL (SHIFT+TAB).

I will explain these areas in the order I think is most logical, starting with the waveform-arpeggio table, since this is the most fundamental part of creating instruments. Don't worry if you don't understand everything directly, in section 5 I will guide you through creating a couple of instruments, the information in this section will start making more sense then.

3.1 WAVEFORMS

The waveforms available are triangle, sawtooth, pulse and noise. These are explained in the introduction. You select waveforms in the waveform-arpeggio (WFARP) table. An instrument requires at least one row in this table, if you are creating a simple instrument then one row is usually enough (waveform doesn't change over time). When more rows are used the table advances to the next row every frame. You insert rows with the SHIFT+DEL (Insert) keys and remove rows with DEL/Pound (Delete/Backspace) keys.

WF	ARP	DT
00	81	00
01	41	..
02	END	..
03
04
05
06
07

Picture 3: Waveform-arpeggio table

Every row has three columns. The first column (WF) sets the waveform, the second (ARP) changes the pitch and the third is for detuning the waveform.

WF-COLUMN

\$00..\$0F Repeat only arpeggio/detune-columns for 1..16 frames (no Waveform-change).

\$10..\$FD Simple WAVEFORM/CONTROL register value setting.

\$FE Jump to a table-position (position in 2nd (ARP) column, if $\geq \$40$, it jumps to itself).

\$FF End of the table, table execution ends here. This value can't be typed, it's shown as END at the end of table.

WAVEFORM / CONTROL FIRST NIBBLE

The first nibble of this value sets the waveform (or combination of waveforms). To combine waveforms, you add the value for one waveform to the other, so to combine Sawtooth and Pulse, the value would be $\$2 + \$4 = \$6$. Remember that that noise waveform can't be combined with other waveforms. The values for the waveforms are:

\$1 TRIANGLE

\$2 SAWTOOTH

\$4 PULSE

For the pulse waveform to be audible you need to insert at least one row in the PULSE table to set the pulse width.

\$8 NOISE

WAVEFORM / CONTROL SECOND NIBBLE

The second nibble (control register) sets different options for the waveform. These values can again be combined by adding the ones you want together, so to put gate on (like you would normally do) and enable ring modulation, the result would be $\$1 + \$4 = \$5$.

- \$1** GATE ON
The gate needs to be on for the waveform to be heard.
- \$2** SYNC
To produce "Hard Sync" effects, explained in the introduction.
- \$4** RING MODULATION
This is explained in the introduction.
- \$8** TEST
This resets and locks the oscillator at zero until this value is cleared.

ARP-COLUMN

- \$00** Zero pitch-shifting (prime, '0' interval) - regain original note-pitch
- \$01..\$5F** Relative pitch-shift upwards in halftones (positive interval)
- \$7F** Jump to default or pattern-FX set Chord.
Depending on the chord, the execution of the waveform table may continue after the chord. With this command you can also override the default chord speed by setting the speed value in the (WF) column (values can be \$00 - \$0F).
- \$80** No process (useful when you don't want to touch pitch and detune, just waveform).
- \$81..\$DF** Set absolute pitch (frequencies are identical to C-1..A-8 notes, see table at the end of this manual).
- \$E0..\$FF** Relative pitch-shift downwards in negative direction (negative interval).

DT-COLUMN (DETUNE)

- \$00..\$FE** Set fine-detuning amount. The higher the value the sharper the pitch will be.
- \$FF** No process in detune-table (the previously set detune-value is retained).

3.2 PULSE WIDTH

You set the pulse width in the PULSE table. This will change the timbre of the pulse waveform only, so you need to set this waveform in the WFARP table. The pulse width doesn't need to be fixed but can be changed over time. This is called Pulse Width Modulation (PWM) and is very common in SID tunes, it makes the sound come to life. Like the WFARP table you insert rows with the SHIFT+DEL (Insert) keys and remove rows with DEL/Pound (Delete/Backspace) keys.

PULSEKT		
00	80 A0
01	40 10
02	END
03

Picture 4: PULSE table

Every row has three columns, the first two columns controls the pulse width and modulation, the third column sets keyboard tracking. Keyboard tracking makes the pulse width dependent on note-pitch. The sounds get more lively and varying by this setting. Keyboard-tracking strength is a signed value in exponential scale, \$00..\$7F will increase the pulse width for higher notes. Negative values \$FF..\$80 (-1..-127) will do the same but not as drastically (i.e. weaker or less steep effect).

When setting / modulating pulse widths, the pulse width bars in the info area of the user interface is of great help, the bar highest up is for voice 1, then voice 2 and voice 3. The last bar shows the cutoff frequency of the filter.



Picture 5: Pulse width / filter cutoff bars

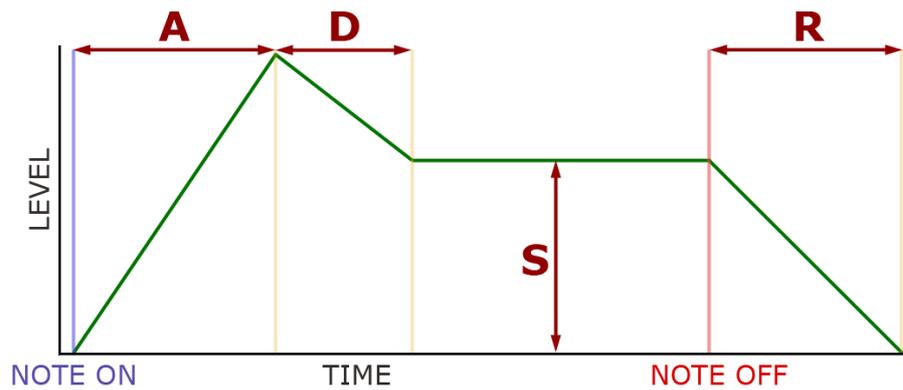
PULSE VALUES (TWO FIRST COLUMNS)

- \$8x..\$Fx xx** set pulse-waveform's pulse width hi-nibble and low-byte. A value of \$000 or \$FFF will produce a constant DC output (silence) a value of \$800 will produce a square wave. The value of the first nibble doesn't affect the pulse width, but it has to be \$8..\$F.
- \$00..\$7F xx** Adds / subtracts signed xx (0...\$7F / \$80...\$ff) to the pulse width 0-127 (\$00-7F) times, in other words, this will animate the pulse width. Use '=' key to negate a positive value or see the conversion table at the end of this manual for signed decimal values (-128 - 127).
- \$FE** Jump to a position in the table (position in 2nd column, can jump to itself)
- \$FF** End of the table, table execution ends here. This value can't be typed, it's shown as END at end of table.

3.3 ADSR ENVELOPE

The ADSR Envelope is common for almost all synthesizers. ADSR stands for Attack, Decay, Sustain and Release. With this envelope you can control how the amplitude of the instrument is changed over time. Three of these parameters are for setting a time value (A D and R) and one is for setting the amplitude while in sustain phase (S).

Take a look at picture 6 below. When a note is played the attack time (A) decides how long it takes for the volume reach max. After this the amplitude is affected by the decay time (D) which means the time it takes for the amplitude to reach the wanted sustain level (S). When a note off command is sent (key released) the release time (R) decides how long it will take for the amplitude to get to zero (silent).



Picture 6: ADSR Envelope

In SID-Wizard you control these parameters with four HEX values. The range of these values are all \$0..\$F.

ADSR 0 6 A A

Picture 7: ADSR values

The approximate timings can be seen in table 1 below. Attack time goes from 2ms (\$0) to 8s (\$F), decay and release times go from 6ms to 24s. See the table at the end of the manual for approximate attack, decay and release timings.

3.4 VIBRATO

SID tunes wouldn't be what they are without vibrato. :) In SID-Wizard there are four vibrato types to choose from, these are:

- TYPE-W|00** Delayed 'up-down'
- TYPE-W|01** Delayed 'downwards'
- TYPE-W|02** Delayed 'upwards'
- TYPE-W|03** Incremental (violin style)

You set the vibrato parameters with four hex values, see the picture below.



Picture 8: Vibrato parameters

The digit to the left is for setting amplitude, the second one is for the frequency (speed), the second half (two last turquoise digits) is the delay (the time it takes before the vibrato is activated) if a delayed vibrato type is chosen OR the increment speed (for the incremental vibrato type).

3.5 FILTER

The SID chip has, as explained in the introduction, only one filter that is shared between all three channels. Every channel gets filtered whose instrument has something in its filter-table's 1st row, when the note starts. The instrument which has \$00 there, is only filtered but doesn't control the filter, while if there's nonzero in the 1st row, the instrument controls the filter (band, resonance and cutoff-frequency). The latest instrument always takes over the control over the common filter-band / resonance / frequency (like with polyphonic jamming in JCH's editor). If there are 2 or 3 notes with filtered instruments starting at the same time, the instrument in the leftmost track controls the filter.

You edit the filter in the filter table (FILT.) The function of this table is quite similar to the WFARP table. Like the WFARP table and the filter table you insert rows with the SHIFT+DEL (Insert) keys and remove rows with DEL/Pound (Delete/Backspace) keys.



Picture 9: The filter table

\$8r..\$Fr xx The first nibble sets the filter band, the possible values are \$8 => filter deactivated, \$9 => low-pass, \$B => low-pass + band pass, \$C => high-pass, \$D => low-pass + high pass, \$E => band-pass + high-pass, \$F => all modes together.

The second nibble (r) sets the resonance, this can be any value from \$0 to \$F, \$F giving the strongest resonance of course. If band is set to \$F (all modes together) then r can't be set to \$F (this is due to the reserved end of table value \$FF).

The third and fourth nibbles (xx) sets the cutoff frequency. Valid values are \$00 - \$FF. The approximate cutoff frequency ranges between 30Hz and 10KHz.

\$00..\$7F xx Adds / subtracts signed xx (0...\$7F / \$80...\$ff) to the cutoff frequency 0-127 (\$00-7F) times, in other words, this will generate a filter sweep. Use '=' key to negate a positive value or see the conversion table in the end of this manual for signed decimal values (-128 - 127).

\$FE Jump to a position in the table (position in 2nd column, can jump to itself)

\$FF End of the table, table execution ends here. This value can't be typed, it's shown as END at end of table.

The third column is Keyboard-Tracking strength of cutoff-frequency. The cutoff frequency gets pitch-dependent with this setting. This effect works in the same way as in pulse width table, and produces more lively sounds which can open in timbre / resonate always near the frequency of the musical tone thus giving impression of live instruments. Keyboard-tracking strength is a signed value in exponential scale, \$00..\$7F will increase the cutoff frequency the higher a note is played. Negative values \$FF..\$80 (-1..-127) will do the same but not as drastically (i.e. weaker or less steep effect).

3.6 OTHER INSTRUMENT PARAMETERS



Picture 10: Instrument parameters

1ST FRAME'S WAVEFORM-REGISTER

(1) Press RETURN (ENTER) to activate / deactivate, the value can be changed when activated. This setting is activated by default. If this setting is activated and set to a value (waveform) that value will be the first waveform of the played sound in the very first frame (no matter if multispeed is set, it will be the first frame any way). The main function of this is the same as in Goattracker, to test / reset the oscillators with waveform \$09 (test-bit set) before the sound starts and get a short 'sexy' start of the sound (the 1st row of the WFARP-table sounds short and not like in old-school tunes).⁹

HARD-RESTART ADSR

(2) The ADSR value for hard restart (gets loaded to ADSR registers 1-2 frames before a new note is triggered). SID provides quite bad accuracy on attack after releasing a note, resetting the ADSR registers before a note is played helps with this problem. Common values are '0F00' and 'F800'.

⁹ These are the actual words of Hermit. :)

The ADSR bug of the SID chip is a mystery to many (after reading about it in the CSDB forums) This is a citation of a comment from the user *Frantic*¹⁰:

"Traditionally... To have access to hardrestart in the player one is using — or better yet, actually understanding the ADSR bug and how hardrestart works — is what separates the lowlife scums and lamers from the elite and the sages in the world of SID. Hence, using hardrestart is a scene sacrament that must be blown out of proportion if order is to be maintained. I vote for continuation of this ritualistic practice."

HARD-RESTART TIMER / TYPE

(3) The first box (numeric value) is the 'hard-restart timer' and sets how many frames before a note to reset ADSR-registers of SID. The second box is hard-restart type (normal hard-restart or staccato / aggregated mode). When enabled (toggle with RETURN / ENTER) the TEST-bit is also to be reset at hard-restart. (Adds 1-2 frames of gap between consequent notes.)

DEFAULT ARPEGGIO TABLE SPEED

(4) This is especially useful for multispeed to slow down WF-table execution if needed. If the value is >\$40 then multispeed will be activated for the pulse width table, if the value is >\$80 then multispeed will be activated for both the pulse width table and the filter table for the instrument. Remember that you can leave this value to \$00 and set the speed in the WFARP-table (see section 3.1 about ARP column).

DEFAULT CHORD

(5) The chord chosen here is linked to the instrument by default, but a pattern-FX can change the chord as well. Write the number of the chord (from the chord table) here.

¹⁰ <http://noname.c64.org/csdb/forums/?roomid=14&topicid=92196&firstpost=2>

OCTAVE-TRANSCOPE AMOUNT

(6) To incorporate bass and high pitched sounds easier - no need to jump between octaves back & forth when jamming. Input a number for the octave transpose and press RETURN (ENTER) with the cursor over the number (or over the box before the number) to switch between + and - (to transpose up or down X octaves).

INSTRUMENT NAME

You can change the name of the instrument by pressing the keys (SHIFT+N). Press RETURN (ENTER) to finish editing saving the name, or RUN/STOP (ESC) to quit editing without saving the name.

3.7 SAVING AND LOADING INSTRUMENTS

One really good feature in SID-Wizard is that you can save and load instruments. This makes it easy to re-use your instruments for other tunes. To save an instrument, have the instrument you want to save selected, then press F8 to get to the main menu. Now you just choose SAVE INSTRUMENT, give a name to the instrument and press RETURN (ENTER).

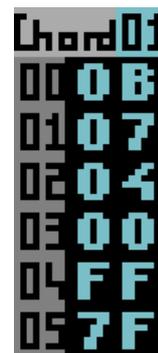
Loading is similar, remember that the selected instrument number in your tune will be replaced by the one you load.

Instruments and patterns stays in buffer between save and load sessions so if you want to bring one instrument from one tune to another, copy the instrument to the buffer with C+=C (CTRL+C) in the instrument editor and then load the other tune, now paste the instrument from the buffer with C+=V (CTRL+V). You need to be in the instrument editor when pasting.

There will be more about file operations and the main menu in section 9.

4. THE CHORD TABLE¹¹

To the right of instrument tables the thin table contains the selected chord's pitch-list. Chords are essentially arpeggios in SID-Wizard, but they don't depend on instruments so any one chord can be used by any of the instruments. The main point of using chords is to reduce the number / redundancy of creating more instruments of the same type just because we want them to sound as different chords on a channel.



Chord	Pitch List
00	0B
01	07
02	04
03	00
04	FF
05	7F

Press C+=F5 (CTRL+F5) anytime to get to chord table. The usage is very simple, you can select a chord with +/- keys here, or SHIFT+K/L keys anywhere. In the column you type the relative pitches (in hexadecimal format) counted in halftones in the order you want them to roll over time.

Insert/Delete keys can expand / shrink size / complexity of the chord. '7F' means looping the chord over, '7E' means returning to the waveform-arpeggio table the chord was called from and continuing waveform-arpeggio-table execution with its next row.

5. INSTRUMENT EXAMPLES

Now when we know the basics of creating instruments I can walk you through a couple of examples.

5.1 BAZZ

Let us first create a "standard" SID bass sound. Go to the WFARP table in the instrument editor, insert a row and write '41' in the first column. This will set the pulse waveform and set gate on.

Since we chose the pulse waveform we have to set the pulse width in the pulse width table. Go there, insert a row and write '82' in the first column, now we can try the instrument with the Z or Q key (this will give you a C-

¹¹ This section is taken from the manual of SID-Wizard 1.2

note). The octave is too high for a base sound, change the octave transpose amount to -3 and press Z again, better.

The sound is a bit boring, go to the PULSE table and add a row after the first one we created. Write '40 20' in the first two columns of the new row. The pulse width is now increased by \$20, \$40 times. Try the sound again. Take a look at the pulse width bar in the info area of the user interface while you try the instrument. Now you can insert the value '10' to both rows in the KT-column (keyboard tracking), when using keyboard tracking you normally set the same value for every row. Try the sound again, now you can see that the pulse width range changes depending on the note you play.

We can tweak the ADSR envelope a little, change the values to '089A' and try the sound, you probably notice that the release is longer.

It sound ok but let's add a little vibrato, set the vibrato parameters to '3720' and the vibrato type to delayed up-down (should be default).

Now we have a nice base sound, as a last thing we can add a low-pass filter. Go to the filter table and insert two rows. In the first row write '9B 45'. This will create a low-pass filter \$9 with \$B resonance at the cut-off frequency \$45. Try the instrument now, it sounds quite dark. Now, in the second row, write '40 05'. This will add \$05 to the cut-off frequency \$40 times (this will make the sound increasingly brighter). Now we have created a quite nice bass sound.

Go ahead and save this instrument now, we will use it later in the pattern example.

5.2 SNARE

A snare sound can be created by creating a wave table. Start with a blank instrument and go to the WFARP table, insert five rows. In the first row, write '81 CE', this will generate a noise wave (gate on) at the precise pitch CE (F-7 according to the table in the end of this manual).

In the next row write '41 AD' and in the following '41 AC', this will generate two frames of pulse waves at the pitches G#-4 and G-4. This will give a tone to the snare sound.

Now for the two last rows, write '80 C4' and '80 C5', this will generate two more frames of noise, notice that the gate is now set to off, so the sound will start fading out at the fourth row according to the R value in the ADSR envelope, even if you don't send a note off (or release the key while jamming).

Since we are using the pulse wave we need to set a pulse width, go to the PULSE table, insert a row and write '88' in the first column. Try the snare now, it stops quite abruptly, we need to edit the ADSR envelope.

Change the decay to 'D', the sustain level to 'E' and the release to '9'. The release parameter makes the biggest difference here, now the snare will fade out smoothly to the sound of beautiful noise. Now we have created a snare drum, since we only used exact pitch values in the WFARP table, all notes will sound the same.

5.3 ARPEGGIO

As a final instrument example, let's create an arpeggio. Start with an empty instrument again and insert a row in the WFARP table. Write '11 7F', the first column will generate a triangle wave and the value in the second column will start the default chord for the instrument.

Now, go to the chord table (C+=F6 or CTRL+F6) and insert six rows. Write (from the top) '00', '03', '07', '0C', '07', '03'. The seventh row should still read '7F', which means that the chord will loop. Check that the chord you edited has the same number as the default chord for the instrument.

Now try the instrument with the Z¹² key or the Q key. It runs quite fast. There are two ways to fix this. One is changing the chord speed to a higher value (try '02') in the instrument parameters. The second option is leaving chord speed at '00' and overriding that speed with a value (again \$02 as an example) in the first column to the left of '7F' we inserted earlier.

You can also change the attack to '8' and release to 'B' for a smooth attack and longer release.

6. PATTERN EDITOR¹³

The largest part of the screen is occupied by the pattern-editor window, where you can see and edit patterns corresponding to the 3 tracks / channels that the SID routine can play. See picture 11 below. Each track has 4 columns. The first column (1) is for the notes / pitch control and note-on / note-off (7) (plus several other) effects. The second column's (2) main function is to select instruments or 'legato', or to contain some small pattern effects. The third (3) and fourth (4) columns are explicitly designed for pattern-effects , but the fourth column is only needed when we use the 'BIG' effects, and in that case it contains the effect-value.



Picture 11: Pattern Editor

¹² You can preview the instrument with any of the note keys that isn't in the hex digit range (1..9 and A..F).

¹³ This section is mostly taken from the manual of SID-Wizard 1.2

To go to the pattern-editor window you can press F5 anytime. You can easily navigate between the three tracks using the cursor keys and 'CONTROL' keys (TAB key in Vice emulator), and the other obvious keys like Page-Up / Page-Down (Slash / SHIFT+Slash), Home, etc... If the cursor is located at an instrument number and you press RETURN (Enter), you get to the corresponding instrument's editor panel. The length / size of the patterns can freely be increased / decreased by pressing Insert / Delete keys in the last row of the patterns (showing the '-end-' signal). The pattern number and the size of the pattern can be seen in the info area ((5) and (6)).

If you change the 'auto-advance' amount with SHIFT+A (increase) or SHIFT+Z (decrease), the cursor jumps the specified amount of rows after you type a note in note-column. If you toggle the 'binding mode' by pressing C+=B, you can scroll each track / pattern independently or otherwise you can move / scroll them together (as long as they're long enough to be scrolled). If you want to mute / solo tracks, press Shift + 1 / 2 / 3 accordingly.

To insert notes you place the cursor in the first column at the row you want the note and then you press the corresponding key for the note you want to insert (see 6.1 below). If this doesn't work you need to press SPACE to activate edit-mode (SPACE toggles between edit and jamming mode, cursor will flash slowly in jamming mode). To insert a note-off you press RETURN (Enter) at the row you want note-off to happen. You can erase notes at the cursor position by pressing 1 or A. You should check out the keyboard layout at the end of this manual for complete summary of keys in the pattern editor.

6.1 NOTE-COLUMN EFFECTS

The note column (1) can hold some pitch / gating related effects, these are displayed graphically in the note-column and can be placed into the pattern by keyboard combinations described before. Almost all of them are simple switches (Sync ON/OFF, Ring-modulation ON/OFF, tone-portamento ON), except vibrato, which can have a one-nibble (0..F) amplitude value.

Note **2 3 5 6 7 9 0** (upper octave black keys)
 Q W E R T Y U I O P (upper octave white keys)

S D G H J L (lower octave black keys)
 Z X C V B N M , . (lower octave white keys)

Note OFF RETURN (Enter)
Auto-portamento SHIFT + P
Vibrato SHIFT + V
Ring Modulation ON SHIFT + R
Ring Modulation OFF C= + R (CTRL + R)
Sync ON SHIFT + S
Sync OFF C= + S (CTRL + S)

6.2 INSTRUMENT-COLUMN EFFECTS

The instrument column (2) can hold some Small effects where the 1st nibble is the effect-type, the 2nd nibble is the corresponding value (0..15 or \$0..\$F). These Small-FX numbers are almost identical to Small-FX and Big-FX numbers of effect-column, and as such, can be remembered easier.

- \$01..\$3E** Select an **instrument** for the actual note, which stays selected until another instrument is selected with this command. *This also resets the tables which restart was switched off with C=+P (CTRL+P) or C=+F (CTRL+F), in this case selecting the same instrument again makes sense and the repetitive instrument selections won't get optimized if 'auto.instr' is ON.*
- \$3F** **Tied note** (true legato, the instrument doesn't restart just note-pitch changes)
- \$40..\$4F** **Waveform** (reg.4) nibble adjusting - any subsequent WF-table waveform change overrides it.
- \$50..\$5F** **Sustain** nibble adjusting of ADSR (Small-FX)
- \$60..\$6F** **Release** (reg.6) nibble adjusting of ADSR (Small-FX)
- \$70..\$7F** Select **Chord** (overriding the default) for the instrument. (No need to create an instrument for every chord. Instead, create an instrument, call chord-table with \$7F from ARP-table, then select the chord with this pattern-FX.)

6.3 EFFECT-COLUMN SMALL EFFECTS

The Small effects' (3) 1st nibble is their type of operation, the 2nd nibble is the corresponding effect-value / amount. Small-FX numbers \$4..\$7 are identical in function to Big-FX numbers of the same range, (what's more, the hexa effect-numbers are related to SID-registers and effect-names) to support easier memorization. Unless you are using the 'extra' player version, only one Small-FX ADSR setting is allowed per note.

\$20..\$2F	Attack nibble adjusting of ADSR (Small-FX)
\$30..\$3F	Decay nibble adjusting of ADSR (Small-FX)
\$40..\$4F	Waveform (reg.4) nibble adjusting - any subsequent WF-table waveform change overrides it.
\$50..\$5F	Sustain nibble adjusting of ADSR (Small-FX)
\$60..\$6F	Release (SID reg.6) nibble adjusting of ADSR (Small-FX)
\$70..\$7F	Select Chord (overriding the default) for the instrument. (No need to create an instrument for every chord. Instead, create an instrument, call a chord with \$7f from ARP-table, then select chord.)
\$80..\$8F	Vibrato Amplitude adjustment - The frequency stays intact.
\$90..\$9F	Vibrato Frequency adjustment - Amplitude stays intact.
\$A0..\$AF	Adjust Main volume (low nibble of \$d418)
\$B0..\$BF	Filter Band (LOW/MID/HI/3OFF) (Filter band can be overwritten with filter-table execution)
\$C0..\$CF	Chord-speed adjustment (arpeggio-speed in case of explicit arpeggio)
\$D0..\$DF	Detune actual note with given amount
\$E0..\$EF	Enable/disable Test-bit/Ring-bit/Sync-bit/Gate-bit (Waveform-table can override)
\$F0..\$FF	Filter Resonance (strength) nibble setting (Subsequent filter-table command can override it.)

6.4 EFFECT-COLUMN BIG EFFECTS

The Big effects has their effect-type in effect-column (3) and their 1 byte (\$0..\$FF or -\$7f..+ \$80) value is in the last column of a pattern (4). (\$4...\$7 effects are identical to Small effect-types of the same range to aid human memorization.)

\$01	Pitch Slide UP (same FX-number as Goattracker's)
\$02	Pitch Slide DOWN (same FX-number as Goattracker's)
\$03	Tone-portamento (with given speed) (same FX-number as Goattracker's)
\$04	Waveform-Control (\$d404, etc.) register setting (or above \$f0: goes to WF-ARP table-position)
\$05	ATTACK/DECAY (SID-register 5) byte adjustment
\$06	SUSTAIN/RELEASE (SID-register 6) byte adjustment
\$07	Select Chord (overriding the default) for the instrument.
\$08	Vibrato amplitude and frequency adjustment (overrides instrument's default vibrato)
\$09	Select vibrato-type (possible values: \$00, \$10, \$20, \$30)
\$0A	Adjust Pulse width program table-index for current instrument (go to table-position)
\$0B	Branch to Filter-program table-index for the actual instrument
\$0C	Chord-speed adjustment (arpeggio-speed in case of explicit arpeggio)
\$0D	Detune actual note with given amount
\$0E	Simple pulse width setting for pulse-waveform (instr.PW-table changes can override it)
\$0F	Filter-cutoff frequency hi-byte shift (added to filter frequency). Now notes don't reset it!
\$10	Set Main (subtune's basic) single-tempo
\$11	Set Main (subtune's) funktempo . 1st (even rows) and 2nd (odd rows) tempo, in left/right nibbles.
\$12	Set Main tempo-program for whole subtune (any speeds can be given for each rows)
\$13	Set track's individual single-tempo

- \$14** Set the **track's funktempo** (even and odd rows' in left/right nibbles)
- \$15** Set the **track's individual tempo-program**
- \$16..\$1C** Kept for later expansions....
- \$1D** **Delay track** by \$00..\$ff (20ms) frames
- \$1E** **Delay** only the actual **note** by \$00..\$ff frames.
- \$1F** *Extra: 'Filt external source' bit on/off in SID ('off' by default, 'on' can reduce noise)

6.5 PATTERN EXAMPLE

In this section I will guide you through creating a simple pattern. You should have the keyboard shortcuts for the pattern editor handy. Remember the edit / jamming mode toggle with SPACE (if the cursor flashes slowly then jamming mode is activated and notes will not be written).

First you should load the 'Bazz' sound from section 5 if you saved it, otherwise create it.

Now, increase the pattern size to '40' (go to the end of the pattern in the first track and press SHIFT+DEL (INSERT) until the last line before the END line has the number '3F').

In the first row, insert a C-6 note (place cursor position over the first column and with a jamming octave setting of 5 press Q). Directly to the right of the C-6 value, write the number of the Bazz instrument.

Insert a G#5 (H) at line \$10, a F-5 (V) at line \$20 and finally a G-5 (B) at line \$30. Remember that you don't need to set the instrument number for every note, only when you want to change instrument. Press F3 now to play / loop the pattern (F4 will stop playing).

Let's try some portamento and pitch sliding. Go to line \$10 and insert a tone portamento big effect (\$03) with the speed '25'. Last two columns in row \$10 should now say '03 25'. Insert a pitch-slide down big effect (\$02)

on line \$37 with the value '20'. Last two columns of line \$37 should now read '02 20'. On line \$3C you can insert a note-off command (RETURN/Enter with cursor in first column). Play the pattern again.

When using slow portamento (the G#5 note on line 10) you may have to adjust the note to play a bit earlier so it doesn't sound as if the note starts late. So delete the contents on line \$10 with DEL (Backspace) and insert the G#5 note and the portamento effect on line \$0C instead.

7. ORDERLIST¹⁴

The very bottom of the screen in the lower border introduces the Orderlist. This is where you can control the whole arrangement of the tune by putting consequent pattern numbers after each other. In playback the currently played Orderlist positions of the song are blinking. It helps you to follow where you are in the timeline of your music.

To get to Orderlist, press F6 (SHIFT+F5) anytime. The navigation keys are obvious here, individual track-sequence lengths can be increased / decreased by pressing the Insert / Delete keys on the last 'FF' or 'FE' values. Pressing RETURN (Enter) over a pattern-number in Orderlist will lead you back to the pattern-editor window and selects the pattern for the corresponding track. Pressing SHIFT+RETURN (SHIFT+Enter)) will select all the 3 patterns corresponding to the current Orderlist cursor-position. Pressing C+=RETURN (CTRL+Enter)) anytime selects the patterns that are played at the moment (useful with different track-speeds or lengths).

*A new function in version 1.2: Pressing C+=E in the Orderlist will find the first empty (unused) pattern after the existing orderlist. If used over \$FE/\$FF end-signals, it automatically inserts the empty pattern's number to the end of the orderlist. The number is incremented upon a new C+=E keypress, and is refreshed anytime you go to a new pattern to edit it. This way you can expand your tune much faster than before.

¹⁴ This section is taken from the manual of SID-Wizard 1.2

The value 'FE' means the end of playback for the given track, 'FF' jumps back to a position in pattern-sequence given by the number following it. (As value '00' is reserved for 'no process', you can't use pattern '00', the lowest pattern-value is '01'.) Be aware that these are hexadecimal numbers, so for example, after '09' not '10' but '0A' comes.

*There are some effects that can be controlled from Orderlist like transposing the musical key or changing volume / tempo.

7.1 ORDERLIST EFFECTS

\$00..\$7F	The values kept for normal pattern-numbers in the orderlist
\$80..\$8F	Transpose key down (e.g.: \$8F = half-note down ; \$8E = whole note down...)
\$90	Switch off transposing (set original key)
\$91..\$9F	Transpose key up (e.g.: \$91 = half-note up ; \$92 = whole note up....)
\$A0..\$AF	Set main volume (\$0..\$f)
\$B0..\$FD	Set track-tempo on the corresponding track (\$00..\$5d)
\$FE..\$FF	These values are reserved to control sequence-flow . Playbacks stops at \$FE and loops when reaching \$FF position. If a position number below \$80 is written after \$FF, it loops to that given position in actual subtune. If the position-number is above \$80 in a track, that track jumps to the given (the number minus \$80) subtune's corresponding track/sequence! (useful for composing music in smaller chunks for e.g. demo-parts)

For a complete subtune-jump you have to type the command for all the tracks, as it's handled on the tracks independently by now. At subtune-jumping the current tempo will be retained unless a tempo-change effect is applied in the target subtune's orderlist/pattern...

(The default subtune-tempo is only effective for fresh tune-starts.)

**Note: An Orderlist-effect shouldn't be right before a \$FF loop-signal. If this happens, the loop command will be ignored to prevent freezing that could be caused by jumping rapidly between orderlist-FX and looping.*

8. TEMPOS AND MULTISPEED

The default tune tempo can be changed with 'SHIFT+.' and 'SHIFT+,'. You can toggle funktempo on and off with C=+T (CTRL+T) and the change the second part of the funktempo with 'SHIFT+: ' and 'SHIFT+; '.

Funktempo means that two tempos are alternated every pattern step so with funktempo activated the first tempo is for even rows in the pattern and the second tempo is for odd rows.

The tempo can also be changed with pattern effects and orderlist effects (see sections 6 and 7).

8.1 TEMPO-PROGRAM TABLE

The table in the bottom-right corner shows the selected tempo-program. For extreme pattern timing / swing / funktempo you can describe complex tempo-programs here. If the tempo-program is selected later from pattern-FX, all of its rows correspond to the timing of consequent rows of the pattern the tempo-program is applied to. Usually we don't need complex tempo-programs, just funktempo, but it might be useful for frequently appearing special rhythms...

Press C=+F7 (CTRL+F7) anytime to get to the Tempo-program table. To select a tempo-program, you can press SHIFT+T/Y anytime or +/- in the tempo-program table itself. Editing is similar to that of the chord-table, except there are no numbers allowed more than \$7F (hexa) value. The tempo-program automatically rolls over after the last value, there's no need for special jump-commands.

8.2 MULTISPEED

Multispeed gives finer / faster control of the SID by modifying its registers multiple times in a 50Hz PAL (or 60Hz NTSC) frame. With 8x multispeed the SID can be modified in 400Hz frequency, and the waveform / arpeggio tables can run really fast... However the 2x speed (100Hz) can be quite enough for some nice effects. (E.g. Rubicon's title music by Jeroen Tel is 2x, and has drums that are not possible in 1x single-speed at 50Hz).

Different programs handle multi-speed differently. In X-SID and SID-Wizard only the tables (waveform / pulse / filter) run at multispeed, the tempo- and vibrato / slide handling is only called in the single-speed part of the player only once in a frame. Opposed to GoatTracker and some others where the whole player is called. Both solutions have advantages and disadvantages. The only disadvantage of the solution in SID-Wizard is that you can't set tune-tempo in a finer scale, but on the other hand in multispeed the tempo / hardrestart / vibrato /slide settings can stay the same....

SID-Wizard supports up to 8x (400Hz) multi-framespeed. Using multispeed does not affect speed values and tempo (just like in Jeff's X-SID). You can change this setting from the main menu (see next section).

9. MAIN MENU AND FILE OPERATIONS

You get to the main menu by pressing F8 (or Shift+F7) as mentioned in the previous chapters, F8 will also close the menu. You can choose / set an option with the obvious cursor up/down keys and then pressing RETURN (Enter). The only exception is the 'DEF.PATT.LEN' menupoint to select default pattern-length, where you may use +/- keys to select default pattern-length for the tune.

```
*****
* - MAIN MENU - *
* ----- *
*   SAVE TUNE   *
*   LOAD TUNE   *
* SAVE INSTRUMENT *
* LOAD INSTRUMENT *
* SELECT DRIVE:08 *
* COLOR: DEFAULT *
* NOTE-KEYS: 2ROW *
* AUTO-INSTR: OFF *
* AUTOFOLLOW: OFF *
* HIDE RBARS: OFF *
* FRAMESPEED: 1X *
* DEF.PATT.LEN:20 *
* KILL TUNE&TEMPO *
* KILL INST&CHORD *
* INSTANT HELP! *
*****
```

Picture 12: Main Menu

Some keyboard functions (related to playback and instrument-selection) will still be functional while you can select an action from the menu. (Pattern-playback display won't be refreshed while menu is displayed in front of it... The latest used menupoint will be remembered even after file-operations or exiting from menu.)

9.1 SIMPLE SETTINGS

SELECT DRIVE

Select the active disk drive for file operations (if you are using Vice with seamless folder integration you don't need to care about this).

COLOR

Change the color theme of the user interface.

NOTE-KEYS

Change between 2-row (Goattracker/Fasttacker) and DMC/GMC piano-keys. 2-row mode is the default in SID-Wizard.

AUTO-INSTR

If this setting is ON the currently selected instrument will automatically be appended to the typed note in pattern-editor. If it remains ON at saving, the unneeded successive instrument-selections are cleared to reduce size of the saved tune. If it's OFF (default behaviour in C64 editors), the instrument-selection 'commands' should be typed by hand. In this case no optimization of successive instrument-selection commands is performed when saving the file.

AUTOFOLLOW

When set to ON, the cursor will automatically follow the playing position.

HIDE RBARS

Hides the raster bars on the sides of the user interface.

FRAMESPEED

Set multi-framespeed (see section 8.2).

DEF.PATT.LEN

Default pattern length.

KILL TUNE AND TEMPO

Clears the tune and sets tempo to default. Before clearing the tune, you might want to use +/- keys on 'default pattern-length' menupoint to give direction for SID-Wizard what pattern-length to use by default when initializing the tune. The default pattern-length will be saved as information in the .SWM workfile, so when you load the tune back later, the empty patterns will automatically be sized to that value (but size can be set afterwards manually as well).

KILL INST&CHORD

Clears all instruments and chords.

INSTANT HELP!

Well, what could this mean?

9.2 LOAD/SAVE TUNE (WORKFILE) OR INSTRUMENT

If you select 'Save' or 'Load' options, they'll invoke the PC-like file-dialog where you can type desired filename directly or select (and further edit) it from the disk directory. (When you want to load or save instrument, make sure the instrument (subject of load/save operation) was selected in the editor/menu beforehand with +/- keys.

When disk-directory is read, filenames are filtered and distributed in 3 columns, and you can observe diskname and free disk space (in kbytes) at the bottom of the screen. Note that only 12 character-long filenames are allowed. That is due to the automatically attached '.SWM' or '.SWI' extension ('M' stands for 'module', 'I' stands for instrument), which SID-Wizard uses to filter the filenames when displaying the disk-directory. The default disk-drive could be selected in advance from the Menu (device 8...15).

In the upper-right corner of the file-dialog you can see the editor's player-version number, if you want to load a newer/other module-version (possibly) coming in the future, you'll be warned, because the module's format-version should always match the editor's or exporter's¹⁵ version.

If you are using .D64 images on a PC you can not overwrite files. For seamless folder integration in Vice; P00 and S00 files will not be overwritten either but a file with the same name but extension (.P01, .P02) will be created, so you should avoid saving files with the same name as existing ones.

¹⁵ Editor: Sid-Wizard, Exporter: SID-Maker

USABLE KEYS IN THE FILE-DIALOG

F1	Re-read disk-directory, go to its beginning
F3/UP-arrow (Page-Down)	Page down in directory (if bigger than one-screen)
CONTROL (TAB)	Cycle between filename-typer / authorinfo-typer / filename-selector fields. (Author-info is only editable when saving tune.)
Cursor Up/Down/Left/Right	Obviously navigating cursor in typer / file-selector fields
RETURN (Enter)	Perform disk-operation (or leave author-info editor). You'll notified if problems occur (file exists/file not found).
F7/STOP (F7/F8/Esc)	Exit from file-dialog back to editor without performing any file operation.

THE SWM FILE FORMAT

The .SWM format is an optimized/compressed own format of SID-Wizard, not compatible with any other systems. But must stay compatible through further SID-Wizard versions. (It contains 'SWM1' filetype&version string to notify editor if its version mismatches.)

If 'auto-instrument-typing' mode is selected (Shift+I or menu), the unneeded repetitive instrument-selections are deleted/optimized before saving to reduce pattern-sizes. The .SWI format is a copy of one selected instrument (without the empty/unused spaces).

10. SID-MAKER (EXPORT YOUR TUNES)¹⁶

This is a separate executable application file on the SID-Wizard disk (or to be downloaded separately), because relocation could hardly be achieved to memory areas which SID-Wizard editor takes for code. I guess this won't be such a big problem considering that this step will possibly be the last step when a tune is ready to be released. The device (disk drive) will be the one SID-Maker was loaded from. SID-Maker can generate 4 kinds of file outputs at the moment: 3 native C64 formats ('.C64.PRG', '.BIN.SEQ' and executable '.EXE.PRG'), and '.SID.SEQ' (SID) format which enables cross-platform playback, if transferred with a cross-platform file-copier like Star Commander or Total Commander D64 plugin. (The .PRG

¹⁶ Text mostly from SID-Wizard Manual 1.2

and .SEQ are standard extensions appended by CBM-DOS. If you need a SID, you have to rename the '.sid.seq' extension of the exported tune to '.sid' - this might depend on your copier tool.) If you use SID-Maker in Vice with seamless folder integration, then you need to use SWMConvert to convert the .S00 file to .sid.

```
* SID-MAKER FOR HERMIT SID-WIZARD 1.2 *
-----
PLEASE SELECT OUTPUT FORMAT WITH CURSOR:
C64/PRG   BIN/SEQ   EXE/PRG   SID/SEQ
(NORMAL) (RAW DATA) (RUNNABLE) (PC-SID)

PLAYER-TYPE VIA CURSOR LEFT/RIGHT: EXTRA
MACHINE-TYPE WITH CURSOR UP/DOWN:  PAL

RELOCATION ADDRESS VIA +/- KEYS: $1000
```

Picture 13: SID-Maker

1. Run SID-Maker, then choose the tune you want to convert.
 2. Choose format. This will be SID/SEQ for SID-files obviously.
 3. Select player-type and machine-type (will be detected automatically).
The player type should match the one you have been using to create the tune.
 4. If you don't understand the relocation address, leave it at the default value (just press RETURN/Enter).
 5. Select the SID Model (will be detected automatically), this should (but doesn't have to) be equal to the version in your C64 or the version you have selected in Vice for the tune to sound similar to what you hear.
 6. Next, write a filename and press RETURN (Enter), your SID-file (or other file) is saved.
- If you export SID format and you used a ':' in the author-information, the SID-header will automatically be split in this fashion: The content before the ':' will be the author-name, the content after the ':' will be the tune-title.
 - Version of modules is checked in SID-Maker as well as in SID-Wizard, SWM module-version needs to match SID-Maker software-version.

- In case you have 'SAVE ERROR' (with CBM-DOS error code) you might not have a compatible Disk Drive like 1541, so the exporter cannot write SEQ files. If you want a compatible mode anyway, you can still perform a standard compatible 'SAVE' process by exporting in the (1st) C64 PRG format without modifying relocation address.

If you set relocation address for PRG or SID format, the exporter will switch to compatible SAVE mode automatically in case of error, but that will involve the exported tune's load-address to be forced to the default \$1000 (or \$0F82 in case of SID) - however the code itself gets relocated. If this is the case, you should load the relocated tune manually by forcing the desired load-address, when you include the PRG in a program. (In case of SID you might delete the 2 byte load-address with a hex-editor to make the SID compatible.)

10.1 CONVERTING .S00 FILE TO .SID WITH SWM-CONVERT

These instructions are for Microsoft Windows (but there is also a binary of SWM-Convert for Linux). You should copy SWMConvert.exe to the Windows\System32 so you can start it easily from the command prompt.

1. Open the Command Prompt and go to the folder where your .S00 file is. In Windows 7 you can do this easily by holding Shift while right-clicking on the folder name (or somewhere in the folder) in Windows Explorer and then choosing *Open command window here*.
2. Then you just write:
`swmconvert inputfilename.S00 outputfilename.sid`

The other parameter 'outputfilename.sid' is optional since a .S00 will automatically be converted to the sid-format.

You can do other things with SWM-Convert, like converting the SWM format of SID-Wizard to / from XM or MID formats. Read the original manual of SID-Wizard for more info about the possible conversions.

10.2 EDITING METADATA IN YOUR SID-FILE

To edit metadata (song name, author, date of release etc.) in your SID-file you can use SIDEdit that can be downloaded here:

http://www.transbyte.org/SID/SIDedit_download.html

11. REFERENCE TABLES

Starts on the next page ...

11.1 GENERAL PURPOSE AND GLOBALLY USABLE KEYS

Keys on C64	US int. Keys in VICE emulator	Category	Functions of the pressed keys
F5	F5	Navigation	Go to Pattern-editor window.
Shift+F5	F6	Navigation	Go to Orderlist .
F7	F7	Navigation	Go to Instrument-editor panel.
C= + F5	CTRL + F5 or F6	Navigation	Go to Chord-table .
C= + F7	CTRL+F7	Navigation	Go to Tempoprogram-table .
Shift+F7	F8	Navigation	Go to Main Menu .
CURSOR-down/right and Shift	Cursor up/down/left/right	Navigation	Move cursor step-by-step up/down/left/right where possible.
CONTROL or Shift+CONTROL	TAB or Shift+TAB	Navigation	Cycle through tracks or instrument-tables.
/ or Shift+/, UP-arrow* or RESTORE	/ or Shift+ / , Page-Down*/Page-Up	Navigation	Move cursor in 4 or 8 steps up/down /left/right depending on window (pattern/orderlist/etc.)
HOME	Home	Navigation	Move cursor to start-position on screen, then absolute start-position in table/pattern.
RETURN	Enter	Navigation	Car return to beginning of row, or go to pattern(s) or instrument under cursor.
F1 or C= + F1	F1 or CTRL+F1	Playback	Play the tune from start / follow-play tune from start.
Shift+F1 or C= + Shift + F1	F2 or CTRL+F2	Playback	Play the tune from marker position(s) / follow-play tune from marker position(s).
F3 or C= + F3	F3 or CTRL+F3	Playback	Play the selected patterns / follow-play the selected patterns.
Shift + F3 or STOP	F4 or ESC	Playback	Toggle pause/continue the playback of tune/patterns.
Shift + SPACE	Shift + SPACE	Playback	Play the selected patterns from cursor-position .
Left-Arrow	` (above TAB)	Playback	Fast-forward play (4x the speed of normal)
Shift + Left-Arrow C= + Left-Arrow	Shift + ` (above TAB) Control + `	Playback, Setting	Toggle follow-play mode / normal playback. Set follow-play mode as default on/off.
Shift+ 1 / 2 / 3	Shift + 1 / 2 / 3	Play./Setting	Toggle Mute / Solo on channels 1..3
Shift + A/Z	Shift + A/Z	Setting	Increase/Decrease auto-advance (stepping) amount after typing notes.
Shift + I	Shift + I	Setting	Toggle instrument auto-typing with notes.
Shift + D	Shift + D	Setting	Toggle monophonic+legato or polyphonic+ dovetailing jamming mode.
Shift + F/G	Shift + F/G	Setting	Decrease/increase framespeed (1x singlespeed...8x multispeed).
Shift + H/J	Shift+H/J	Setting	Decrease/increase step-highlighting aid frequency in pattern-editor window.
C= + B	CTRL + B	Setting	Toggle pattern/ track-binding on/off . Patterns can be scrolled together/independently.
C= + T	CTRL + T	Setting	Toggle global normal tempo / funktempo mode for subtune.
Shift + colon/dot , Shift + : or ;	Shift + colon/dot , Shift + ; or '	Setting	Increase/decrease global normal tempo / 2nd funktempo for subtune.
C= + colon / dot	CTRL + colon / dot	Selection	Select subtune (Orderlist). Patterns are shared between subtunes.
C= + 1...8	CTRL + 1...8	Selection	Select octave 1...8 to edit or jam musical notes. (In 'DMC' mode no need for C=)
Shift+PLUS or Shift+MINUS	Shift+PLUS or Shift+MINUS	Selection	Select instrument for jamming / editing.
Shift + T/Y	Shift + T/Y	Selection	Select (decrease/increase number of) tempo-program .
Shift + K/L	Shift + K/L	Selection	Select (decrease/increase number of) Chord to edit.
SPACE	SPACE	Editing	Toggle Keyboard note-jamming or note-editing in pattern-editor window.
Shift + N	Shift + N	Editing	Rename selected instrument. Max. 8 characters, Esc/Stop aborts the renaming.
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease pattern/table size in end-positions.

*In Linux the VICE build seems to have up-arrow key associated to Page-Down, while the default would be 'Del'-key (which is good, because therefore in Linux the Page-Down (up-arrow) is in correct place, below Page-Up, which is associated to Restore C64 key.) In Linux the 'Del'-key works as Backspace which is better selection IMO.

*Note that in follow-play modes the pattern-editor keys are inactive to prevent accidental editing while playing the tune.

11.2 PATTERN EDITOR RELATED KEYS

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift+CONTROL	TAB or Shift+TAB	Navigation	Cycle through tracks forward/backward.
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	Go to next row, place Note-on/off , or select instrument/chord under cursor.
Z X C V B N M , , . (A...L in DMC mode)	Z X C V B N M , , . (A...L in DMC mode)	Edit/Jam	Lower octave white piano-keys (C major diatonic whole-tones). Legato jamming possible.
Q,W,E,R,T,Y,U,I,O,P (not in DMC mode)	Q,W,E,R,T,Y,U,I,O,P (not in DMC mode)	Edit/Jam	Upper octave white piano-keys (C major diatonic scale whole-tones).
S,D, G,H,J, L , 2,3, 5,6,7, 9,0 (non-DMC)	S,D, G,H,J, L , 2,3, 5,6,7, 9,0 (non-DMC)	Edit/Jam	Lower and upper octave black piano keys (halftones). In DMC-mode it's in QWERTY row!
C= + 1...8 (1...8 in DMC-mode)	CTRL + 1...8 (1...8 in DMC-mode)	Edit/Jam	Select octave 1...8 where editing or jamming musical notes happens.
A, 1, C+=Del (Q in 'DMC' mode)	A, 1, Cotrol+Del (Q in 'DMC' mode)	Editing	Empty-note (Delete note in note-column pattern-position without moving the rest.)
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	In instrument/effect columns type value , in note-column set note or vibrato-amplitude .
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease pattern-size in pattern-end position.
C= + DEL	CTRL + Delete/Backspace	Editing	Delete note or instrument+effect columns in actual pattern-row, depending on cursor-position.
C= + Shift + DEL	CTRL + Shift + Del./Backspace	Editing	Delete the entire pattern-row (note and instrument and effect) in the actual track.
Shift + Q/W	Shift + Q/W	Editing	Transpose notes up/down by half-notes in actual pattern after cursor-position.
C= + Q/W	CTRL + Q/W	Editing	Transpose notes up/down by octaves in actual pattern after cursor-position.
Shift + R or C= + R	Shift + R or CTRL + R	Editing	Place ring-modulation effect on/off into note column at actual cursor position.
Shift + P	Shift + P	Editing	Place auto-portamento effect into note column at actual cursor position.
Shift + S or C= + S	Shift + S or CTRL + S	Editing	Place sync-bit on/off effect into note column at actual cursor position.
Shift + V	Shift + V	Editing	Place vibrato -effect into note column at actual cursor position. Amplitude editable.
C= + X	CTRL + X	Editing	Cut/delete pattern content from cursor position to end of pattern and copy to pattern-buffer/clipboard.
C= + C	CTRL + C	Editing	Copy pattern content from cursor position to end of pattern to pattern-buffer/clipboard.
Shift + C	Shift + C	Editing	Limit the range of data copied (or cut) into buffer to cursor position. Define end of buffer.
C= + V	CTRL + V	Editing	Paste pattern-buffer content from cursor position till end of pattern.

11.3 ORDERLIST RELATED KEYS

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	Selects the pattern(s) under the cursor position in Orderlist and jumps to actual track in Pattern-editor.
C= + RETURN	CTRL + Enter	Navigation	Selects the patterns that are currently played and jumps to actual track in the Pattern-editor.
Shift+SPACE	Shift + Space	Playback	Sets Orderlist playstart-marker (for F2-playing) to cursor-position for all tracks. Position number gets inverted.
C= + SPACE	CTRL + SPACE	Playback	Sets Orderlist playstart-markers to individual played position for all tracks. Position number of 1st track gets inverted.
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	To type hexa pattern-numbers and effect/jump numbers into Orderlist sequences of tracks.
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease Orderlist sequence-size if cursor is in sequence-end/loop position.
C= + C	CTRL + C	Editing	Copy orderlist-sequence from cursor to buffer.
Shift + C	Shift + C	Editing	Set the end of copied data in buffer. (limit buffer)
C= + V	CTRL + V	Editing	Paste buffer to cursor-pos ., appends existing data
C= + E	CTRL + E	Editing	Type and the first Empty (unused) pattern number . Increased if pressed more times, works on endsignal.

11.4 INSTRUMENT EDITOR RELATED KEYS

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift+CONTROL	TAB or Shift+TAB	Navigation	Cycle through main instrument-setting panel and instrument-tables.
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	Car return' to beginning of next row in tables (or toggle/cycle some main instrument-parameters.)
+ / -	+ / -	Selection	Select instrument. The same in pattern/orderlist/instrument windows, while Shift is needed in Chord/Tempo tables.
RETURN	Enter	Selection	Toggle/cycle some of the instrument's main parameters like HR-type, Vibrato-type, octave-shift sign.. Or goes to default-chord if it's number is under the cursor.
1...9, A...F hexa keys and F..Z	1...9, A...F hexa keys	Editing	To type hexa values into instrument-data fields, or type the name of the instrument (some signals allowed too.)
=	=	Editing	Negate value (e.g. \$40 will be \$C0 = -\$40)
Shift + N	Shift + N	Editing	Rename selected instrument . Max. 8 characters, Esc/Stop aborts the renaming.
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position. Increases/decreases table-size.
Shift + Space	Shift+Space	Editing	Set/clear gate-off index to current table-row under the cursor
C= + C	CTRL + C	Editing	Copy the entire selected instrument to instrument-buffer.
C= + V	CTRL + V	Editing	Paste instrument -clipboard content to the selected instrument. The entire instrument will be overwritten.
C= + P or C= + F	CTRL + P / F	Editing	Toggle pulsewidth-/filter-table reset on note-start. (Selecting instrument resets it.)

11.5 CHORD TABLE / TEMPO TABLE RELATED KEYS

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift+CONTROL	TAB or Shift+TAB	Navigation	Go to pattern editor or go to instrument-editor.
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	Car return' to beginning of next row in tables (or toggle/cycle some main instrument-parameters.)
+ / -	+ / -	Selection	Select chord / tempo-program. Shift+K/L or Shift+T/Y is needed in other windows.
Shift+PLUS or Shift+MINUS	Shift+PLUS or Shift+MINUS	Selection	Select instrument for jamming / editing.
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	To type hexa values and chord loop/return numbers into chordtable/tempoprogram-table.
=	=	Editing	Negate value (e.g. \$04 will be \$fb -4 2s' compl.)
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position. Increases/decreases table-size.

11.6 DECIMAL TO HEXADECIMAL CONVERSION TABLE

Next page ...

DECIMAL TO HEXADECIMAL CONVERSION TABLE

Mikael Norrgård 2012
(<http://www.witchmastercreations.com>)

UNSIG.	SIG.	HEX
0	0	00
1	1	01
2	2	02
3	3	03
4	4	04
5	5	05
6	6	06
7	7	07
8	8	08
9	9	09
10	10	0A
11	11	0B
12	12	0C
13	13	0D
14	14	0E
15	15	0F
16	16	10
17	17	11
18	18	12
19	19	13
20	20	14
21	21	15
22	22	16
23	23	17
24	24	18
25	25	19
26	26	1A
27	27	1B
28	28	1C
29	29	1D
30	30	1E
31	31	1F
32	32	20
33	33	21
34	34	22
35	35	23
36	36	24
37	37	25
38	38	26
39	39	27
40	40	28
41	41	29
42	42	2A
43	43	2B
44	44	2C
45	45	2D
46	46	2E
47	47	2F
48	48	30
49	49	31
50	50	32
51	51	33
52	52	34
53	53	35
54	54	36
55	55	37
56	56	38
57	57	39
58	58	3A
59	59	3B
60	60	3C
61	61	3D
62	62	3E
63	63	3F

UNSIG.	SIG.	HEX
64	64	40
65	65	41
66	66	42
67	67	43
68	68	44
69	69	45
70	70	46
71	71	47
72	72	48
73	73	49
74	74	4A
75	75	4B
76	76	4C
77	77	4D
78	78	4E
79	79	4F
80	80	50
81	81	51
82	82	52
83	83	53
84	84	54
85	85	55
86	86	56
87	87	57
88	88	58
89	89	59
90	90	5A
91	91	5B
92	92	5C
93	93	5D
94	94	5E
95	95	5F
96	96	60
97	97	61
98	98	62
99	99	63
100	100	64
101	101	65
102	102	66
103	103	67
104	104	68
105	105	69
106	106	6A
107	107	6B
108	108	6C
109	109	6D
110	110	6E
111	111	6F
112	112	70
113	113	71
114	114	72
115	115	73
116	116	74
117	117	75
118	118	76
119	119	77
120	120	78
121	121	79
122	122	7A
123	123	7B
124	124	7C
125	125	7D
126	126	7E
127	127	7F

UNSIG.	SIG.	HEX
128	-128	80
129	-127	81
130	-126	82
131	-125	83
132	-124	84
133	-123	85
134	-122	86
135	-121	87
136	-120	88
137	-119	89
138	-118	8A
139	-117	8B
140	-116	8C
141	-115	8D
142	-114	8E
143	-113	8F
144	-112	90
145	-111	91
146	-110	92
147	-109	93
148	-108	94
149	-107	95
150	-106	96
151	-105	97
152	-104	98
153	-103	99
154	-102	9A
155	-101	9B
156	-100	9C
157	-99	9D
158	-98	9E
159	-97	9F
160	-96	A0
161	-95	A1
162	-94	A2
163	-93	A3
164	-92	A4
165	-91	A5
166	-90	A6
167	-89	A7
168	-88	A8
169	-87	A9
170	-86	AA
171	-85	AB
172	-84	AC
173	-83	AD
174	-82	AE
175	-81	AF
176	-80	B0
177	-79	B1
178	-78	B2
179	-77	B3
180	-76	B4
181	-75	B5
182	-74	B6
183	-73	B7
184	-72	B8
185	-71	B9
186	-70	BA
187	-69	BB
188	-68	BC
189	-67	BD
190	-66	BE
191	-65	BF

UNSIG.	SIG.	HEX
192	-64	C0
193	-63	C1
194	-62	C2
195	-61	C3
196	-60	C4
197	-59	C5
198	-58	C6
199	-57	C7
200	-56	C8
201	-55	C9
202	-54	CA
203	-53	CB
204	-52	CC
205	-51	CD
206	-50	CE
207	-49	CF
208	-48	D0
209	-47	D1
210	-46	D2
211	-45	D3
212	-44	D4
213	-43	D5
214	-42	D6
215	-41	D7
216	-40	D8
217	-39	D9
218	-38	DA
219	-37	DB
220	-36	DC
221	-35	DD
222	-34	DE
223	-33	DF
224	-32	E0
225	-31	E1
226	-30	E2
227	-29	E3
228	-28	E4
229	-27	E5
230	-26	E6
231	-25	E7
232	-24	E8
233	-23	E9
234	-22	EA
235	-21	EB
236	-20	EC
237	-19	ED
238	-18	EE
239	-17	EF
240	-16	F0
241	-15	F1
242	-14	F2
243	-13	F3
244	-12	F4
245	-11	F5
246	-10	F6
247	-9	F7
248	-8	F8
249	-7	F9
250	-6	FA
251	-5	FB
252	-4	FC
253	-3	FD
254	-2	FE
255	-1	FF

11.7 EXACT NOTES FOR ARP-COLUMN

Value	Note	Value	Note	Value	Note	Value	Note
81	C-1	99	C-3	B1	C-5	C9	C-7
82	C#	9A	C#	B2	C#	CA	C#
83	D	9B	D	B3	D	CB	D
84	Eb	9C	Eb	B4	Eb	CC	Eb
85	E	9D	E	B5	E	CD	E
86	F	9E	F	B6	F	CE	F
87	F#	9F	F#	B7	F#	CF	F#
88	G	A0	G	B8	G	D0	G
89	G#	A1	G#	B9	G#	D1	G#
8A	A	A2	A	BA	A	D2	A
8B	Bb	A3	Bb	BB	Bb	D3	Bb
8C	B	A4	B	BC	B	D4	B
8D	C-2	A5	C-4	BD	C-6	D5	C-8
8E	C#	A6	C#	BE	C#	D6	C#
8F	D	A7	D	BF	D	D7	D
90	Eb	A8	Eb	C0	Eb	D8	Eb
91	E	A9	E	C1	E	D9	E
92	F	AA	F	C2	F	DA	F
93	F#	AB	F#	C3	F#	DB	F#
94	G	AC	G	C4	G	DC	G
95	G#	AD	G#	C5	G#	DD	G#
96	A	AE	A	C6	A	DE	A
97	Bb	AF	Bb	C7	Bb	DF	Bb
98	B	B0	B	C8	B		

11.8 ATTACK / DECAY / RELEASE TIMINGS

HEX	Attack Rate	Decay / Release Rate
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	250 ms	750 ms
A	500 ms	1.5 s
B	800 ms	2.4 s
C	1 s	3 s
D	3 s	9 s
E	5 s	15 s
F	8 s	24 s