

Pocket Tools 2.00

User Manual

08.06.2015

1.	Introduction	2
1.1.	Why do you need the Pocket Tools or why not?	2
1.2.	What do I need so I can use the Pocket Tools?	2
1.2.1	Hardware	2
1.2.2	Software	2
2.	Basic Description	2
2.1.	Wav2bin 2.00	2
2.2.	Bas2img 6.00	2
2.3.	Bin2wav 2.00	3
3.	Recording	3
3.1.	How to record the wave file from a Pocket Computer?	3
3.2.	Recording guide for Audacity	3
3.2.1	What should the recorded wave file look like?	4
3.2.2	What should the recorded wave file not look like?	5
3.3.	How to prevent processing errors with Wav2bin from a wave file?	6
3.4.	How to transfer programs to a Pocket Computer using the tape interface?	7
4.	Feature list	7
4.1.	Not implemented	7
5.	Handled file types	8
6.	Command line options	10
6.1.	Bin2wav	10
6.2.	Wav2bin	11
6.3.	Bas2img	13
7.	Error codes, returned to OS	13
8.	Old Parameter format	14
9.	Supported SHARP PCs, cassette files and commands	15
10.	Use of the Tools, chain and examples	17
11.	Default load addresses of binary code	20
12.	How To	21
12.1.	Best method to transfer BASIC source files to PC-E500 series	21
12.2.	What type of conversion of special chars should be used with Wav2bin type BAS	21
12.3.	How should you type in the Basic source code?	21
12.4.	Convert variable data tap format to Transfile PC, edit and return it (PC-1210 to PC-1475)	22
13.	Known Limitations	22
14.	Thanks to	23

1. Introduction

The Pocket tools are a set of programs with which you can convert between audio records of SHARP Pocket computers, and binary files and text files of a personal computer.

You can edit these files on your PC with any editor and convert them back to audio files.

1.1. Why do you need the Pocket Tools or why not?

If you have an adapter cable for each of your SHARP pocket computer that you can connect to a port of your PC, you have the right driver for this interface installed on your operating system (and on the pocket computer, if necessary), and if the data transmission with the (cable-specific) interface settings and your software for the required file types with that operating system works, then you do not need the pocket tools.

If you are not sure whether everything still works now, or want the data transmission using your SHARP pocket computer still to work for many years, then we recommend that you learn about Pocket Tools.

1.2. What do I need so I can use the Pocket Tools?

1.2.1 Hardware

You need any sound adaptor or sound card with a microphone input and headphone output.

The adaptor of your personal computer should support microphone pre-amplification, and the cables to the audio terminal should not run next to lines with high current. For very insensitive microphone inputs an additional preamp is needed.

For the pocket computer you need an audio interface, usually a cassette interface, that suits your pocket computer. In general interfaces work also without batteries or low batteries.

PC-1500: CE-150 or CE-162E

PC-1600: CE-1600P (for mode 1 see PC-1500)

PC-121x: CE-121 or CE-122

All other pocket computers: CE-124 (except G850 series), CE-126P or another suitable 11-pin commercial or self-made cassette interface

To backup a BASIC program from PC-1245 up to PC-1475 you need only a microphone and a silent room, but this is tricky and not recommended.

If you use an emulator for a SHARP pocket computer, then you need no interface.

1.2.2 Software

You need a recording software or an audio editor with recording function. Audacity is recommended. No driver is needed.

2. Basic Description

2.1. Wav2bin 2.00

With Wav2bin you can convert the content of a wav file containing the sound of a SHARP pocket computer program to a binary file, to be used on your Personal Computer or a source text file. The resulting files are editable.

2.2. Bas2img 6.00

With Bas2img you can convert a source text file containing SHARP BASIC language to

1. a binary image file (IMG) with intermediate code,
2. a binary image (TXT) without intermediate code or
3. an ASCII file (ASC) with pocket-specific format.

2.3. Bin2wav 2.00

With Bin2wav you can convert the content of a binary image file of your Personal Computer utilizing the internal format of a SHARP pocket computer to a wav audio file, which you can play back with any media player.

Initially, set both the system master volume and the media player volume to 75%.

Then modify both values simultaneously to find the limits of the range within which the pocket computer can do a CLOAD of the program from the sound card correctly.

For both volume controllers, select a value slightly above the mean of this range, and note down each combination of settings for this PC and audio hardware.

3. Recording

3.1. How to record the wave file from a Pocket Computer?

Record PC-1500 with a minimum sample rate of 11025 Hz, others with 22050 Hz.

For sound from modified Pocket Computers with a hardware speedup switched on: use 48000 Hz or more (and later use the wav2bin -cspeed parameter +-5% accurate).

You can record high levels and clear signals with 8-bit, otherwise: Record 16-bit!

Record monophonic. If the software does not support this, then record stereo.

In general, the microphone input (pink) should be used. Pre-amplification should be switched on for the microphone in the sound card, especially for PC-1500, PC-1600, and PC-121x.

Where there is a choice between Mic and Line In, the LINE INput (blue) should only be used for direct recordings from cassette recorder.

Unplug and replug the microphone plug (usually red) in the sound card to select the correct port. Disable all mixers and filters.

Record at a higher level, but not above 95%.

First start CSAVE or the cassette recorder, wait up to one second for a stable audible sound, then record it.

Sound from PC-1600 and newer models starts after a silent time. The preceding silent period can be omitted, however the intermediate periods and some time after the end of transfer should be recorded. The data blocks are split by silent periods. Don't stop recording too fast! If the silence continues more than 8 sec, then it is safe to stop.

In earlier versions of Wav2bin you had to convert to 8-bit, mono and low frequency.

With this version all this is no longer required. Normalizing is recommended but not mandatory.

3.2. Recording guide for Audacity

The following guide of "Edgar Pühringer" and "Norbert Roll" for "Audacity" is adapted to the current tools version, new supported SHARP formats and software version now.

1. System settings

On a Mac, you may have to set the sound input to "Line In" using the utility "LineIn". Normally set the recording device to the microphone input.

On Windows right click the loudspeaker symbol in the task bar.

Switch microphone pre-amplification on in the preferences of your microphone-device (never to the highest level, if you can adjust it).

2. Start Audacity.

Set Edit->Preferences as follows:

Devices (I/O) -> Channels = Mono (1)

Quality-> Standard Sample Frequency = 22050 Hz

(PC-1500: = 11025 Hz or better)

(Hardware Speedup used : = 48000 Hz or better)

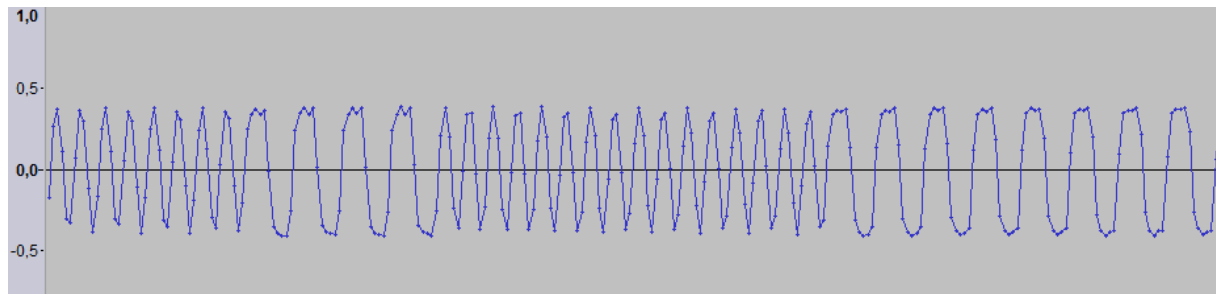
Quality-> Standard Sample Format = 16-bit (or better)

3. Set the Input Volume from 85% (older series) to 95% (PC-1500 and newest series) for most sound hardware first. If the resulting recording will be overdriven (between +/-100%), then the level should best be reduced to +/-70%.
4. Start the SHARP pocket (C)SAVE / PRINT# and wait for it to produce sound.
E2- and G-series may not produce any sound. For the other:
Wait up to 1 second and start a recording in Audacity (avoid preceding silence).
5. After the Sharp has stopped producing sound, stop the recording in Audacity.
However about 10 milliseconds (minimum) should remain after the last sound.
PC-16/E/G-series include one or more silent blocks between and after the data:
PC-1600 up to 9 sec and PC-E/G-series up to 6 sec silence
If in doubt, wait 10 seconds after the last sound was sent.
Normally it should not be trimmed and never resampled!
It is best if you normalize the recording: Effect->Normalize (remove DC, -1.0 dB).
6. File->Export the sound recording as WAV 16-bit PCM.

Wav2bin applies its own preamp with a filtering rule and then converts the PCM format to 8-bit internally.

3.2.1 What should the recorded wave file look like?

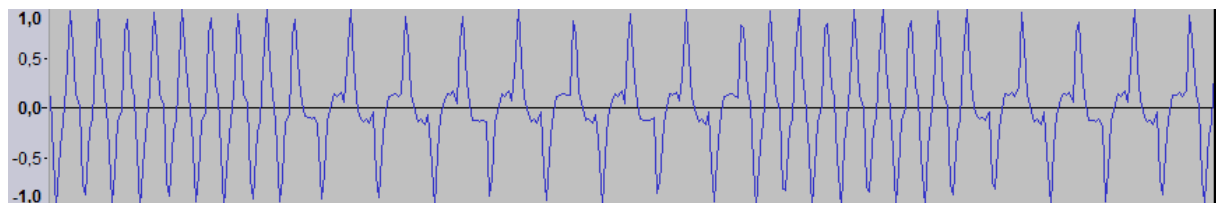
Part of a wave file (PCs from PC-1210 to PC-1475)



The number of wave cycles must be a multiple of 8 for the 4.0 kHz signal (for Bit 1, sync bits or stop bit(s)) and a multiple of 4 for the 2.0 kHz signal (Bit 0 or the start bit of a quater).

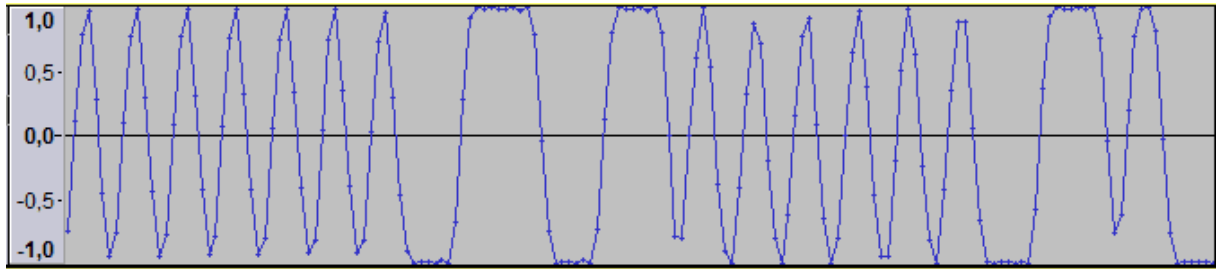
For more information, please see Heise c't 5/88, page 116 "Draht zum großen Bruder" and "Ergänzungen und Berichtigungen" in c't 5/88, page 116 or see the article by Norbert Unterberg "SHARP <--> PC" chapter 5.2.

Part of a normalized wave file of the PC-1500, CE-150/162E directly connected



The number of wave cycles must be a multiple of 8 for the 2.5 kHz signal (for Bit 1, sync bits or stop bits) a multiple of 4 for the 1.25 kHz signal (Bit 0 or the start bit of a quater). For more information, please see the Technical Reference Manual of PC-1600 ch 3.11.2, page 122-124.

Part of a wave file of the PC-1600 with CE-1600P or PC-E- and -G-series



There are only allowed waves of 3.0 kHz signal (for Bit 0, sync bits) and waves of the 1.2 kHz signal (Bit 1 or the start bit of a byte, less sync bits). For more information, please see the Technical Reference Manual (TRM) of PC-1600 chapter 3.11.1, page 117-121 (IOCS) and the TRM of PC-E500 chapter 3 no.04 cas: page 64-66.

Avoid excessive activities of other software at your computer while recording!

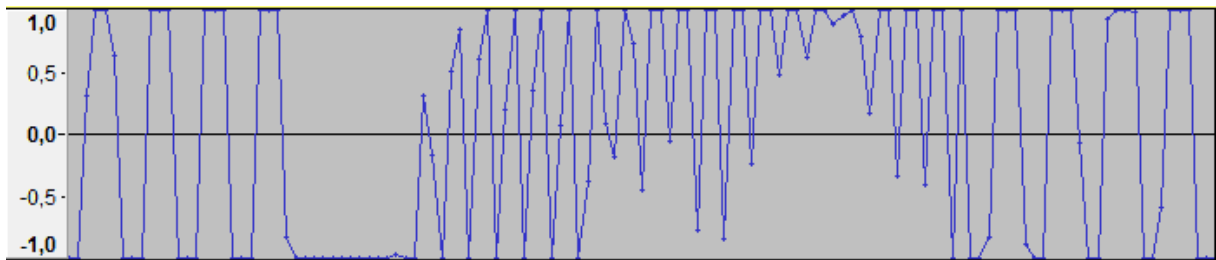
Your sound system must always provide a clear signal with no dropouts, clicks or hum during the entire recording.

If you cannot resolve this, then you have to use another sound adapter or computer for the recordings. Otherwise Wav2bin can not fully convert and reports the errors of your wave file.

3.2.2 What should the recorded wave file not look like?

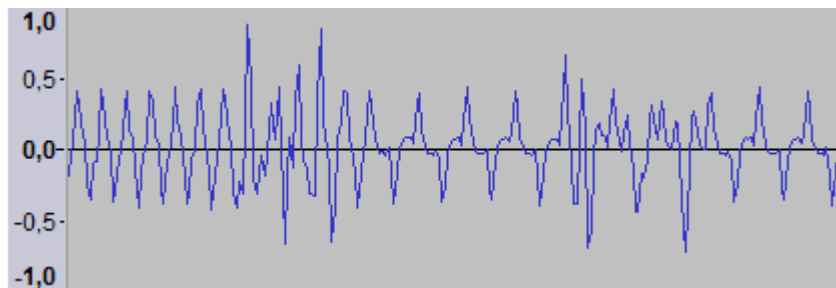
Shown here are some typical errors. Many of these errors cannot be resolved by wav2bin.

Dropouts (PC-1245-1475)



Cause: Tape dropouts and less DC compensation of the interface with the tape recorder output

Disturbances, PC-1500 (PC-1211)

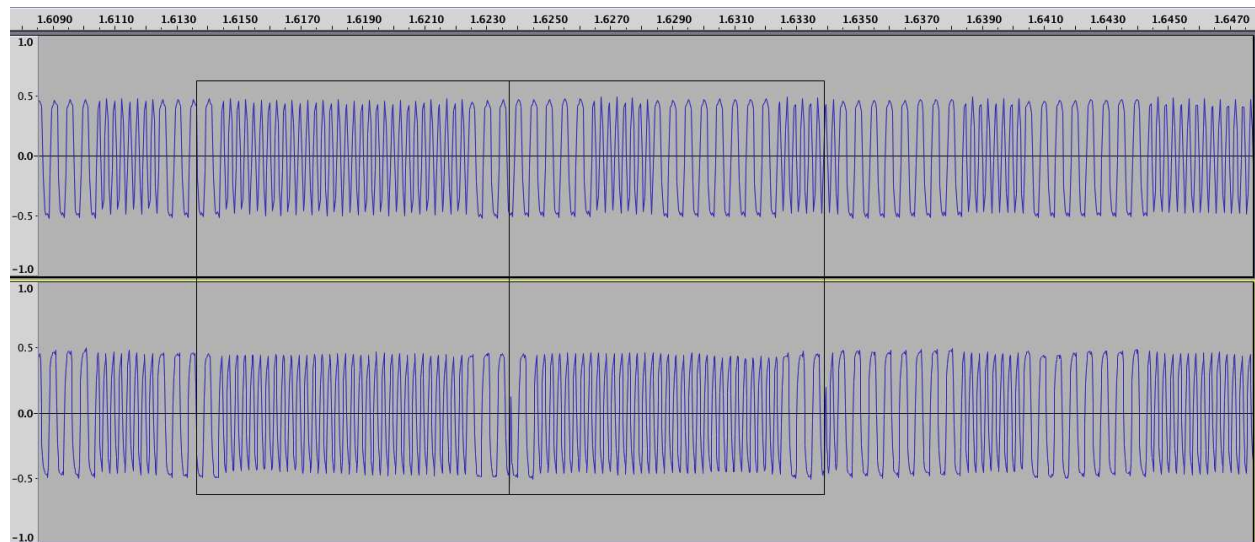


Cause: Induction from other cable to the audio cable (internal power supply cable)

Some of the following problems are caused by energy-saving modes of the computer. The screensaver should not be active during long audio recordings.

Dropped waves can be detected by spikes in the curve or incorrect wavenumbers.

Repeated samples – a copy of a part of the waveform overwrites a later part (PC-1403)



Upper trace: correct recording

Lower trace: disturbed recording

Cause: Sound recording software, operating system and sound hardware

3.3. How to prevent processing errors with Wav2bin from a wave file?

If the level from mic input is only less than 5% during recording, then you must use a good mic preamps. Behringer MIC100, connected with ProCab CAB714 to SHARP CE, has been successfully tested.

In case of errors you should use batteries and avoid the use of an old power supply (mains hum) for CE-126P or other interface. Try to unplug any power supply first, and then connect the CE. Avoid a long audio cable. In case of side effects, connect only the microphone plug, not the headphone. Signal monitoring, cables and low levels may generate uncorrectable bit errors.

When digitizing from cassette, use a cassette deck with a flutter < 3%.

Preceding speech or other sound can preclude the detection of base frequency (no sync found). The same problem arises if the preceding silent period of PC-16/E/G-series is overlaid with a disturbing frequency. In this case you have to cut it, but normally don't cut the lead in! If this is necessary, then cut directly on a sample point to avoid resampling.

Before you normalize very low sound, cut louder noise following the transfer 10 ms or more after the signal. Normalization and wave editing was tested with SoX and Audacity.

Dont convert the sample rate! If you have to do that for other reasons, then use a recording with a sample rate of 48 kHz or higher.

Try to use the device specific filter rules:

1. For sound via system from emulator also set `--device=EMU` and `cspeed`. For sound recorded via microphone from air use a silent room, avoid near (about 5 cm) reflections (for example a display) and set `--device=AIR`.
2. For borderline recordings direct from cassette recorder `--device=CAS` may be helpful.

If you get constant errors from a tape, use debug levels to find the wave time of the error and check it with an audio editor or convert the wav file to a tap file.

If Wav2bin still can't find any synchronisation, then open the wav file in the audio editor, select Analyze->Frequency analysis (FFT), Spectrum and navigate to the highest peak.

This is the base frequency, typically for

PC-1500	2500 Hz,
PC-1600/E/G series	3000 Hz
All others PCs	4000 Hz

If the frequency deviates by more than 5% of it, then something is wrong and the corresponding factor must be passed to Wav2bin with the parameter "cspeed" = (measured/ typically base frequency).

3.4. How to transfer programs to a Pocket Computer using the tape interface?

If the wav file was zipped, unzip it (for example, use 7-zip).

Do not connect the original small black plug (Remote) to the sound card!

1. Connect the EAR input of your SHARP cassette interface with the grey plug (usually) preferably to the HEADPHONE output of your computer, or alternatively to the LINE OUTput (green) of the sound card. Do not connect the original small black plug (Remote) to the sound card!
2. Enter the appropriate (C)LOAD / INPUT# command on your pocket computer and wait 2 seconds (for some of PC-E/G series, until remote relays is switched on).
3. Play back the WAV file with a media player on the PC.
You can start your tests with a volume of 75% (Master and Player), and adjust both, if necessary. PC-E500S and PC-G850 series may need a higher volume.

4. Feature list

1. All SHARP pocket computers with a cassette interface are supported:
from PC-1210 up to PC-G850VS and PC-1100 series
2. Reads and writes all standard cassette file types of all SHARP pocket computer generations:
BASIC source file or Image, ReSeRve key data, Binary data or machine code, all multiple variable data, ASCII Data, ASCII Sources, Text Mode
3. Reads and writes for PC-1245 (PC-1211) up to PC-1475 all binary files for "Transfile PC plus " (SHC) and the BASIC sources (SHA)
4. Reads and writes merged basic source (line number 99999) and images
5. Supports inline comments and offline comments (offline ' not transfered) and different line number formats without and with ':'
6. Supports shortcuts of many commands 'C.'
7. Conversion of special chars between SHARP generations and operation systems conversion to upper case letters (if necessary),
Processing of undocumented signs and tokens within a generation
8. Limited conversion of numeric variable data and text between SHARP generations
9. Supports all usual wave audio files without any conversion or finishing, support for different wave forms
10. Supports wave- and tap-files from emulators and wave files from pocket computers with hardware speedup
11. Automatic correction of many disturbances in wave audio files and named filters for recording from cassette recorder and microphone, all checksums are checked, easy and reliable transmission
Note: CE-1600P, E-Series and G-Series is a clear signal quality with working DC-compensation required for the data contained
12. All Operating Systems with ANSI C Compiler are supported
13. Provides return codes for inclusion in scripts
14. Parameters in standard format and old format supported
15. Continuous debug traces and more special options exist, see the help screens

4.1. Not implemented

Bas2img does not support E500 series intermediate code completely. Read "HowTo" for doing a full transfer from ASCII source code to E500 series.

5. Handled file types

- WAV file** RIFF Wave File, mono or up to four channels (first is used)
sample rate 5 kHz (PC-1500) /8 kHz (PC-1211 to PC-1475) up to 96 kHz
- A) from the audio interface of a SHARP pocket computer
 - B) from a cassette tape --device=CAS
 - C) from the emulated cassette tape device of "Pockemul"
 - D) from the system sound of an emulator --device=EMU (PC-1500, 1600 and newer)
 - E) via microphone from the buzzer --device=AIR (PC-1245-1475)
- The wave file should be recorded using an audio editor with a sample frequency about 22050 Hz and 16-bit. PC-1500 can be recorded with 11025 Hz. If more editing is required, use 48000 Hz.
- The wave files made by Bin2wav are aligned and therefore the frequency can be lower.
- A wav file could be normalized and minor errors corrected with an audio editor. Normalizing is recommended but nothing else.
- If you want to convert the wav file to 8-bit (unnecessary), then you have to normalize it first! Conversions can be done with SoX. Audacity is the recommended editor.
- The wave file must be a public readable type with less chunks.
- tap file** Raw binary format of quaters (PC-1500) or bytes (all other PCs), represents the content of the wave file
- This format was introduced by Olivier De Smet for his emulators running on Android devices for SHARP pocket computers.
- The ID, the name, all headers, end marks and all checksums are included in this files type.
- You should not change the file with a hex editor, because you also have to correct the checksums. But you can try to correct errors from a corrupt wav file after the wav file was converted to a tap file: wav2bin -type=tap
- All single bits, synchronisations and spaces are ignored!
- It is possible to create a tap file from a wav file but impossible to make a wave file from a tap file.
- shc file** Binary byte format of the Software "Transfile PC plus" or an OEM version, Copyright Yellow Computing or licensed partners
- All formats of PC-1245-1475 are supported with -type=shc .
- PC-1211 is supported, but included in the format of PC-1251.
- The ID, the name, all headers and the end marks are included in this files type, but no checksums. A shc file can contain images, binary data or variable data.
- You could edit it with a hex editor, but note that most headers and some data bytes consist of swapped quaters.
- img file** Binary byte format of Wav2bin -type=img, Bin2wav and Bas2img that represents the internal format of a pocket computer series, the intermediate code of a BASIC program, usually generated with CSAVE.
- All formats of all SHARP pocket computers are supported, but using Bas2img the computers of PC-E series have to switch to the TEXT Modus and back to BASIC after the transfer.
- No ID, no name, no file headers, and no checksums are included in this file type.
- In older versions sometime an end mark was included, but now it is removed and should removed from all older images.

- img file**
(type txt)
- Binary byte format of Wav2bin, Bin2wav and Bas2img that represents the internal format of a pocket computer, if it is switched to text mode, usually generated with CSAVE.
- There does not exist any intermediate code inline of the BASIC program, but line numbers and the envelope of every line are binary coded. The content of the line is released in the ASCII format.
- This format is supported with PC-1360, 1475, all older pocket computers with a serial interface and the PC-E500 series, but it is not a recommended format.
- No ID, no name, no file headers and no checksums are included in this files type.
- rsv file**
(or in img)
- Binary byte format of Wav2bin –type=rsv and Bin2wav, that represents the internal format of ReSerVe key data of pocket computers, if it is switched to RSV mode, usually generated with CSAVE.
- No ID, no name, no file headers, no file end and no checksums are included in this files type.
- This format is interchangeable between PC-1500 and PC-1600 or between similar pocket computers of series PC-1350 - PC-1475.
- dat file**
- Binary byte format of Wav2bin and Bin2wav –type=dat, that represents the special internal format of data variables of the pocket computers, usually generated with PRINT #.
- No ID, no name, no file headers, no file end and no checksums are included in this file type.
- The block headers of all data variable blocks and the end mark of any standard variable data block are included in this file type. Multiple data blocks are supported.
- All formats of all SHARP pocket computers PC-1211- PC-1600 are supported. Data of PC-1211 are saved in the standard variable format of PC-1251.
- The formats of all the different PC generations are limited interchangeable, if possible. Sometime you have to specify the type of conversion with an –I option.
- To edit the special data, we recommend not to use a hex editor, but to use the shc format and the third-party software (Transfile PC plus).
- bin file**
- Binary byte format of Wav2bin and Bin2wav –type=bin, that represents any internal format of any data of a pocket computer, usually binary code, generated with CSAVE M.
- No ID, no name, no file headers, no file end and no checksums are included in this files type. Very limited interchangeable!
- If you use this format for machine code, then you have to note the start address of the code (and the entry address also, if this is supported by the pocket computer) or use shc instead.
- For ihx files use Hex2bin to convert into a bin file first.
- For PC-E500 note the following regarding the switch –dINV.
- asc file**
- ASCII data file format of Wav2bin and Bin2wav, that represents the data format of the pocket computer, if the commands OPEN “CAS:data” with PRINT #n was used
- This format is supported for PC-E200/PC-G Series and also for PC-1600/E500-series. For the PC-E500-series with level inverting interfaces the signal must be mirrored by Bin2wav with –dINV, same for some self made interfaces.
- Also some of PC-G series can write but not read this format from CAS: correctly.
- No ID, no name, no file headers, no fill bytes, no end mark, and no checksums are included in this file type. The format is similar to bas, but there are differences in the header.
- For use with INPUT #n this is not recommended for G850 series.

- bas file** ASCII source file format of Bas2img, Bin2wav and Wav2bin, that represents the source format of the pocket computer, if it is switched to the Text Editor Cmt menu or the command
- SAVE "CAS:filename" (A with PC-1600) was used.
- This format is supported with Bin2wav for PC-E200/PC-G Series and also for the PC-1600/E500-series. For PC-E500-series some interfaces require --device=INV and other not understand it.
- The user manual of PC-1600 points to a similar problem at page 6-24 (4) .
- No ID, no name, no file headers, no fill bytes, no end mark and no checksums are included in this file type.
- This format is editable with a text editor, however we recommend to always use bas2img before bin2wav (also for PC-1600–G850, because of the differences between the newer PC series).
- sha file** Text format of the Software "Transfile PC plus" or an OEM version, Copyright Yellow Computing or licensed partners, only the basic source format ".BAS" is supported by Bas2img and Wav2bin --type=sha directly, other formats have to be saved as binary shc file for use with Bin2wav.
- The type, the name and all headers are included in this files' first text line of a block, but no checksums.
- The format is very similar to bas files.
- Special chars are encoded for DOS with CP437 only.

6. Command line options

It is recommended to use pre-configured scripts with environment variables or an overlaying software. Otherwise you have to open a **console window** and run the tools in order to pass all parameters.

6.1. Bin2wav

Usage: bin2wav [Options] SrcFile(.typ) [DstFile(.wav/tap)]

SrcFile	Binary image file (usually created by BAS2IMG or WAV2BIN)																
DstFile	WAVE file (default: SrcFile.wav) or tap file																
-t, --type=TYPE	Source file type: <table> <tr><td>img</td><td>BASIC-program binary image (default)</td></tr> <tr><td>bin</td><td>Binary assembly program or data</td></tr> <tr><td>dat</td><td>Data variable blocks (binary data)</td></tr> <tr><td>rsv</td><td>ReSeRve data (binary image)</td></tr> <tr><td>txt</td><td>Text modus</td></tr> <tr><td>shc</td><td>Transfile PC</td></tr> <tr><td>asc</td><td>ASCII Data</td></tr> <tr><td>bas</td><td>ASCII Source</td></tr> </table>	img	BASIC-program binary image (default)	bin	Binary assembly program or data	dat	Data variable blocks (binary data)	rsv	ReSeRve data (binary image)	txt	Text modus	shc	Transfile PC	asc	ASCII Data	bas	ASCII Source
img	BASIC-program binary image (default)																
bin	Binary assembly program or data																
dat	Data variable blocks (binary data)																
rsv	ReSeRve data (binary image)																
txt	Text modus																
shc	Transfile PC																
asc	ASCII Data																
bas	ASCII Source																
-p, --pc=NUMBER	Model number of Sharp pocket computer, currently available: 1211, 1245, 1251, 1261, 1280, 1350, 1360, 1401, 1402, 1403, 1421, 1450, 1460, 1475, 1600, E500, E220, G850 and more (default: 1500)																
-c, --cspeed=VALUE	Ratio of CPU frequency to original (use it with a modified Pocket Computer with speedup switched on, 0.94 to 2.7)																
-a, --addr=VALUE	Start address, needed for BIN type (default: see this manual)																

Usage: bin2wav [Options] SrcFile(.typ) [DstFile(.wav/tap)]

	Entry address 0 to 65535 or 0xFFFF, E500:0xFFFFFFFF (default: no autostart)
-s, --sync=VALUE	1. use Synchronisation duration, expressed in seconds, 0.5 to 9 (default: 0.5 or minimum for the PC and waveform) 2. use of this parameter: for spaces (silence) of PC-1600 and newer
-nNAME, --name=	Sharp file name (7 characters max, 16 for the PC-1500, E:8) (default: DstFile without extension, nor path)
-q, --quiet	Quiet mode (minimal display output)
--tap	Destination file: Emulator tap byte format (not wave file)
--version	Display version information
--help	Display this information
--help=I	show option screen
-d, --device=TYPE	INV interface with inverting level converter (mirror)
-I, --level=VALUE	Option bits and Print debug traces (2-times usable) a hexadecimal integer (0x____) or sum of it: <u>Waveform and frequency</u> (default sample rate is 4* base frequency):
1	Force triangle waveform for base frequency (old compact format)
2	Force wave with 48 kHz (PC-1500: 44.1) near rectangle waveform
3	Force wave with sample rate of 16 kHz for emulator and so on
	<u>Convert Data variables between series:</u>
0x04	Convert PC-1500/1600 numeric data to other PC standard variable, otherwise to numeric array
0x08	Data for PC-1500/1600 of length 8 are numeric data from other PC
0x10	Convert Strings between ASCII code and Old Basic Code
0x1000	Use tape format of PC-1475 (slow) for E5-series CLOAD@ of old images
0x4000	Write no file header, have to merge data blocks manually
0x8000	Data variable block is from Wav2Bin 1.5 or version before
0x800	Write also, if checksum bug will be activated (not readable)
0x400	Write long synchronisation like the original measure
0x200	Write long synchronisation like Tech Ref Man.
0x80	Print some global infos more
0x40	Print all bytes and (Sum_calculated) - see also Wav2bin
0x20	Position and byte list, for data only

For more options - see the source code

6.2. Wav2bin

Usage: wav2bin [Options] SrcFile(.wav/tap) [DstFile(.typ)]

SrcFile	WAVE file (PCM, normalized, recommended is mono 8-bit with a sample rate from 11025 (PC-1500) to 22050 (other Pockets) or original wave files made by Bin2Wav
DstFile	Destination file (BASIC-program text or Binary image file)

Usage: wav2bin [Options] SrcFile(.wav/tap) [DstFile(.typ)]

Options

-t, --type=TYPE	Destination file type
bas	BASIC-program text file (default), sha shc Transfile
img	Binary image file (BASIC program, machine code, data)
rsv	ReSeRVe mode data (image), tap Emulator raw format
-p, --pc=NUMBER	SHARP pocket computer, (for BASIC-program text, is required for 1421, 1600 Mode 1, optionally for other)
-c, --cspeed=VALUE	Ratio of CPU frequency to original (use it with a modified Pocket Computer, if speedup was switched on, 0.94 to 2.7 or from emulator via system sound 0.2 to 5)
-u, --utf8=VALUE	Convert special characters (only for BASIC-program text)
no	to ASCII transcription with brackets [reversible]
yes	to UTF-8 (default)
2dos	to DOS-US
2asc	to ASCII transcription for later serial transfer
-w, --width=VALUE	Minimum width of line number (only for BASIC-program text)
3 to 10	(default: 5) for text format
0	indentation off for pure serial transfer
-q, --quiet	Quiet mode (No display output)
--tap	Source file: Emulator tap format, no wave
--version	Display version information
--help	Display help, more screens with =r:recoding FAQ, =l:level(debug)
-l, --level=VALUE	Special and debug options, 2-times a hexadecimal integer (0x____) or sum of it
0x800	Ignore false checksums and continue: Result must be corrected manually.
0x1000	Depress base frequency tuning and use fixed base frequency only
0x2000	Use flat method of Wav2Bin 1.5 to detect H/L-transitions, no amplitudes
0x4000	No amplitudes based gain, count amplitudes only
0x8000	No analysis of wave file, no pre-amplification
0x80	Print lines of text written (BASIC only)
0x40	Print all Bytes and (Checksum_readed=Sum_calculated) checksums in brackets - see also Bin2wav
0x400	ByteList + WaveTime(after the Byte)
More options exist for low level debugging - see the source code	
More write options:	
--type=tap	convert wav to tap file, no checksum processing
--type=raw	for debugging and =rawdats (quaters swapped)
Device specific filters:	
--device=AIR	: PC-1245-1475 buzzer -> Mic (tricky, avoid near reflections, quiet)
--device=CAS	: For wave made directly from recorder, better DC compensation
--device=EMU	: Wave from emulator via system sound, set --cspeed and --pc

6.3. Bas2img

Usage: bas2img [Options] SrcFile [DstFile]

SrcFile : BASIC program text file

DstFile : Binary image file (default: SrcFile.img or .asc)

Options:

- p, --pc=NUMBER : Sharp pocket computer, currently supported
 1150, 1211, 1245, 1248, 1251, 1261, 1280, 1350, 1360, 1401
 1402, 1403, 1421, 1425, 1430, 1445, 1450, 1460, 1475, 1500
 1600, E220, G850, and more (default: 1500),
 Only at E500 series must be used the commands 'TEXT' and then 'BASIC' after an img transfer
- t, --type=TYPE : destination file type (default: img)
 img BASIC program binary image with intermediate code
 txt TEXT mode image,
 asc ASCII file (for device CAS: or Text Editor Menu Cmt)
- q, --quiet : Quiet mode (minimal display output)
 --help : Display this information
 --version : Display version information
- l, --level=SUM : 1 Don't compile fixed line numbers (inline)
 : 2 Append missing apostrophes at end of line
 : 4 Don't replace shortcuts(.) with commands
 : 8 Don't convert to upper case
 : 0x10 Deactivate preprocessor with special chars conversion
 : 0x80 (0x20) Print lines in, 0x40 Print values out
 : 0x800 Depress some line errors, result may not editable

Bas2img should also used, to convert from general BAS-files to pocket-specific ASC-files.

All offline comments, lines beginning with ' and empty lines are removed.

7. Error codes, returned to OS

ERR 1	Arguments missing, syntax error or nothing to do
ERR 2	Misplaced bit order in the wave file, nibbles in bytes or other
ERR 3	Arguments problem, for example: pocket not implemented
ERR 4	Error with line numbers
ERR 5	File I-O error or file used by another software
ERR 6	Image line to long or other buffer overflow
ERR 7	False or unknown format of wave or SHARP file header
ERR 8	Transmission Error, Checksum read differs from calculated result, or result would not be readable
ERR 9	No synchronisation found in wav file or unexpected lost
ERR 10 (and higher)	Multiple errors found: Error number = last error + 10

8. Old Parameter format

Use it only with old software, that needs this format!

WAV2BIN *SrcFile DstFile [T:type] [G:graph] [D:level]*

<i>SrcFile</i>	WAV file, which must be compliant to the following rules :
<i>DstFile</i>	Output file, the format will depend of the source file : - ASCII file, if the source WAV contains a Basic program, - Binary file, if the source WAV contains Assembly Program or Datas,
<i>type</i>	Destination file type (option for the BASIC sources files) :
<i>graph</i>	Convert special characters (option used with T:BAS) :
<i>level</i>	Print debug traces

BIN2WAV *SrcFile DstFile T:type PC:num [A:addr] [S:sync] [N:name]*

<i>SrcFile</i>	Binary file
<i>DstFile</i>	WAV format output file.
<i>type</i>	Source file type :
<i>num</i>	Destination PC number :
<i>addr</i>	Start addresse for the binary files (option for T:BIN)
<i>sync</i>	Synchro duration, expressed in seconds
<i>name</i>	Sharp file name

BAS2IMG *SrcFile [DstFile] [PC:type] [/Q] [/?]*

<i>SrcFile</i>	Basic source file in ASCII
<i>DstFile</i>	Binary Image destination file
<i>type</i>	Basic file type :
<i>/Q</i>	Quiet mode (disable display)
<i>/?</i>	Display the help

This format was reactivated for compatibility reasons only and does not support new parameters.

Sample Scripts

You can find sample scripts for the Windows command line on our server. Drag and drop is supported. An external tool for the file selection is included, see the GetFName script.

All the settings you need to apply in accordance with your environment in the SHARPSET file first time! Start with the scripts Bas2wav and Wav2bas.

9. Supported SHARP PCs, cassette files and commands

PC-1211, 1210, 1212

0x80	PC-1211 Basic image <u>or</u> RSV image CSAVE / CLOAD
0x8F	PC-1211 data PRINT # / INPUT # Data stored with header of Old data, A(27) and following removed from end, if equal zero (use CLEAR after any changes of the program) No binary available, because of 4-bit CPUs
Shc	Supported, use Pctyp: PC-1251, avoid unknown token

PC-1251-1255, 1245, 1246-1248, 11x0

0x20	Old format, Basic image <u>or</u> RSV image CSAVE / CLOAD
0x21	Old format, Basic image with password (only Wav2bin)
0x24	Old data, multiple blocks, PRINT # / INPUT #
0x26	Old binary, assembly program or datas CSAVE M / CLOAD M no binary for 4-bit CPU available (PC-1246-1248)
Shc	Supported

PC-1260-1262, 1280, 1350, 1360, 1401-1475

0x70	New format, Basic image <u>or</u> RSV image CSAVE / CLOAD, also tokens of 2 byte length are supported CLOAD@ (PC-E200 series, use -p1460)
0x71	New format, Basic image with password (only Wav2bin)
0x72	Extended format, Basic image <u>or</u> TEXT modus image <u>or</u> RSV img. CSAVE / CLOAD (PC-1280, 1360, 1470U, 1475) CLOAD@ (PC-E500 series, use -l 0x1000 / 2bin: -p475)
0x73	Extended format, Basic image with password (only Wav2bin)
0x74	New/Ext. data, also <u>multiple blocks</u> , PRINT # / INPUT #
0x76	New/Ext. binary, assembly program or datas CSAVE M / CLOAD M no binary for 4-bit CPU is available (PC-1430/31)
Shc	Supported

PC-1500, PC-1600 Mode 1

pc=1600M1)

(--

Id	Sub-Id	
0x A 0		PC-1500 binary, assembly program or datas, CSAVE M / CLOAD M
0x A 1		PC-1500, Basic image CSAVE / CLOAD,

		for PC-1600 tokens use -pc=1600
0x A 2		PC-1500/1600 format RSV image CSAVE / CLOAD from RSV mode
0x A 3		Not supported by this version
0x A 4		PC-1500/1600 data, multiple blocks, PRINT # / INPUT #
Shc		Not supported, but new/old data are converted to PC-1500/1600 data with Bin2wav

PC-1600 with CE-1600P (--pc=1600P)
Default CAS: I/O only, No Double write!

Id	Sub-Id	
0x 0	0	ASCII Data, OPEN CAS:, INPUT#n / PRINT#n <u>or</u> Source file image, splitted in blocks, SAVE/LOAD CAS: (without ,A :only Wav2bin)
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-1600, Basic image CSAVE / CLOAD,
0x 2	2	PC-1600/1500 format RSV image CSAVE / CLOAD from RSV mode
0x 4	0	ASCII source file, SAVE/LOAD CAS: (with ,A) ASCII comment lines, SAVE* CAS:
0x 8	4	PC-1500/1600 data, multiple blocks, PRINT # / INPUT #

PC-E500- E650, 1480-1490, U6000

Default CAS: I/O parameter only!

Id	Sub-Id	
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-E500, Basic intermediate code image <u>or</u> TEXT modus image, CSAVE / CLOAD,
0x 4	4	ASCII data, OPEN CAS: , INPUT#n / PRINT#n <u>or</u> ASCII source file, SAVE/LOAD CAS:

PC-E200, E220,G801-G850

Default CAS: I/O parameter only!

Id	Sub-Id	
0x 1	0	PC-16/E/G binary, assembly program or datas, CSAVE M / CLOAD M
0x 2	1	PC-G800, Basic image CSAVE / CLOAD (G850V: BSAVE/ BLOAD)
0x 4	1	ASCII data, OPEN CAS: , INPUT#1/PRINT#1 <u>or</u> ASCII source file, Text Editor Cmt menu

10. Use of the Tools, chain and examples

Convert a

Wav file into a BASIC source text (CSAVE)	wav2bin <i>WavFile</i> < <i>BasFile</i> > recommended: --pc= <i>Number</i> optional: --type=BAS --utf8=yes(no...)
BASIC source text into Wav file (CLOAD)	1. bas2img <i>SrcFile</i> < <i>ImgFile</i> > --pc= <i>Number</i> 2. bin2wav <i>ImgFile</i> < <i>WavFile</i> >--pc= <i>Number</i> optional: --name= <i>SharpName</i>
Wav file into a BASIC image file (with intermediate code normally)	wav2bin <i>WavFile</i> < <i>ImgFile</i> > --type=IMG recommended: -pc= <i>Number</i>
BASIC image file into a wav file (CLOAD)	bin2wav <i>ImgFile</i> < <i>WavFile</i> > --pc= <i>Number</i> optional: --type=IMG --name= <i>SharpName</i>
Wav file into a RSV image file (with intermediate code normally)	wav2bin <i>WavFile</i> < <i>RsvFile</i> > mandatory for PC121x-1475: --type=RSV (img) recommended: -pc= <i>Number</i> <u>for PC-1475/1280 see Techn. Report P-055</u>
RSV image file into a wav file (CLOAD)	bin2wav <i>RsvFile</i> < <i>WavFile</i> > --pc= <i>Number</i> required for PC1500/1600: --type=RSV optional: --name= <i>SharpName</i>
Wav file into a DAT image file (PRINT#)	wav2bin <i>WavFile</i> < <i>DatFile</i> > --type=DAT(img) recommended: -pc= <i>Number</i>

DAT image file into a wav file (INPUT#)	<pre>bin2wav <i>DatFile</i> <<i>WavFile</i>>--pc=<i>Number</i> --type=DAT optional: --name=<i>DataName</i> recommended for PC-1600: -s1 -s2</pre>
Wav file into a BIN image file (CSAVE M)	<pre>wav2bin <i>WavFile</i> <<i>BinFile</i>> --type=BIN (img) recommended: --pc=<i>Number</i> <u>Note the start (and entry) address!</u> Write it to filename.ADR and filename.CAL with scripts.</pre>
BIN image file into a wav file (CLOAD M)	<pre>bin2wav <i>BinFile</i> <<i>WavFile</i>>--pc=<i>Number</i> --type=BIN recommended:--addr=<i>StartAddress</i> optional 15/1600/E5 2nd. : --addr=<i>EntryAddress</i> optional: --name=<i>SharpName</i> recommended (PC-E500): --device=INV/std</pre>
Wav file into a SHC image file (CSAVE (M) / PRINT#)	<pre>wav2bin <i>WavFile</i> <<i>ShcFile</i>> --type=SHC recommended: --pc=<i>Number</i></pre>
SHC image file into a wav file (CLOAD (M) / INPUT#)	<pre>bin2wav <i>ShcFile</i> <<i>WavFile</i>>--pc=<i>Number</i> --type=SHC optional: --name=<i>SharpName</i> --addr=<i>StartAddress</i></pre>
Wav file into a SHA source text (CSAVE)	<pre>wav2bin <i>WavFile</i> <<i>ShaBasFile</i>> --type=SHA recommended: --pc=<i>Number</i></pre>
Wav file into a BASIC image PC-1600 - SAVE (CAS:	<pre>wav2bin <i>WavFile</i> <<i>ImgFile</i>> --type= IMG recommended: --pc=<i>Number</i></pre>

Wav file into a BASIC source text Text Editor Cmt-SAVE (CAS:,A) for PC-G/E/1600	wav2bin <i>WavFile</i> < <i>AscFile</i> > --type=ASC recommended: --pc= <i>Number</i>
BASIC source into Wav file Text Editor Cmt - LOAD (CAS:) for PC- PC-G/E/1600	1. bas2img <i>SrcFile</i> < <i>AscFile</i> > --type=ASC/bas --pc= <i>Number</i> 2. bin2wav <i>AscFile</i> < <i>WavFile</i> >--type=BAS --pc= <i>Number</i> optional: --name= <i>SharpName</i> recommended (PC-E500 series): --device=INV/std
Wav file into a BASIC source text from Text Mode CSAVE	wav2bin <i>WavFile</i> < <i>BasFile</i> > see <i>WavFile</i> with intermediate code into <i>BasFile</i>
BASIC source text into Wav file Text Mode – CLOAD for 1280/1350-60/1450-75/(PC-E)	1. bas2img <i>SrcFile</i> < <i>TimgFile</i> >--type=TXT --pc= <i>Number</i> 2. bin2wav <i>TimgFile</i> < <i>WavFile</i> >--type=IMG(txt) --pc= <i>Number</i> optional: --name= <i>SharpName</i>
Wav file into ASCII data (OPEN CAS:, PRINT #n)	wav2bin <i>WavFile</i> < <i>AscDataFile</i> > --type=ASC recommended: --pc= <i>Number</i>
ASCII Data text into Wav file (OPEN CAS:, INPUT #n) <u>Problems described above.</u>	bin2wav <i>AscDataFile</i> < <i>WavFile</i> > --pc= <i>Number</i> --type=ASC optional: --name= <i>DataName</i> recommended (PC-E500 series): --device=INV/std (only if I/O error after 2. block)

Wav file into a binary raw tap image file (no check of plausibility or correctness)	wav2bin <i>WavFile</i> < <i>tapFile</i> > --type=TAP recommended: --pc= <i>Number</i>
tap file into Dstfile	wav2bin <i>tapFile</i> < <i>DstFile</i> > --tap all other parameter: see Wav into DstFile
Image File into tap file (Bin2tap)	bin2wav <i>File</i> < <i>tapFile</i> > --tap all other parameter: see File into Wav
More variants exist.	

11. Default load addresses of binary code

Read a system manual of your Pocket Computer, before you use this:

PC number	Default load address (hex)	Before entering "CLOAD M" you have allocate memory with:
1245-1255	B830	Pointer Begin Basic: C6E1/2, NEW
1260	5880	Pointer Begin Basic: 66E1/2, NEW
1261, 1262	4080	Pointer Begin Basic: 66E1/2, NEW
1401, 1421	3800	Pointer Begin Basic: 46E1/2, NEW
1402	2000	Pointer Begin Basic: 46E1/2, NEW
1450	2030	Pointer Begin Basic: 5F01/2, NEW
1350	2030	Pointer Begin Basic: 6F01/2, NEW
1403, 1425, 1460	8030	Pointer Begin Basic: FF01/2, NEW
1360	8030	Pointer Begin Basic: FFD7/8, NEW
1475, 1280	8030	Pointer Begin Basic: FFF0/1, NEW
1500	40C5	NEW Address_after_code or set 7865/6
1501, 1500A	7C01	NEW Address_after_code or set 7865/6
1600	C0C5	NEW "Sn",length+C5 or set F865/6 or CALL &02DD,Len (barcode rsv, bgnptr F034/5)
E200, E220, G series	0100	MON, *USER end_address
E500 series	BE000	Set BFD1A-C to BE000 (before BF21B) with: POKE &BFE03,&1A,&FD,&B,0,&1C,0:CALL &FFFD8 Caution! Images of binary disk files may contain a header of 16 bytes length before the executable machine code

Please backup all your programs and data before you modify the pointers, use NEW or CLOAD M ! Read in the manual how to reset your system in case something unexpected happens.

12. How To

12.1. Best method to transfer BASIC source files to PC-E500 series

We recommend to also use the simple CSAVE command for PC-E500 series.

If you generate a image file from a source file with Bas2img, then the numeric data in the lines is not converted to BCD and the binary distances for jumps are not generated. This is a limitation of this version of Bas2img for the format of PC-E500 series only. That's why the pocket computer has to be switched to TEXT menu and back to BASIC after the transfer.

Nevertheless, you should use this format because it is faster than the others and has a stable transmission performance.

The older format with Bin2wav -l 0x1000 for CLOAD@ would be very slow (full returnable from wav file only with wav2bin --pc=E475), the CSAVE format of the TEXT modus is a "one way" for E500 series and the ASCII wave format must sometimes be level inverted (depending from the interface), is slower and was optimized for an audio interface, that switches a cassette recorder off and on before every data block. Use ASCII format for short sources and to MERGE sources.

For archiving purposes, you should always keep an IMAge that was created with a Wave file from an executable program - not the BASic file only at the E500-series.

12.2. What type of conversion of special chars should be used with Wav2bin type BAS

When using a modern editor that can work with UTF8 characters, then use -utf8=yes (default)
This way you can handle many graphic chars directly.

When transferring data between different systems, transferring programs between different pocket computer generations, or using a older editor, use -utf8=no .

All special chars are converted to placeholders in square brackets.

Bas2img will transfer it back depending on the target model.

If you want to transfer the source text via serial transfer later, or use old basic text for a newer pocket generation, use -utf8=2asc .

PI and SQR are resolved, and some stars are added between old variable names.

If you want to use the source text with a DOS-based software, then use the command line option -utf8=2dos. This will convert many of the special chars to codepage 437.

12.3. How should you type in the Basic source code?

1. Try to find the source code on the Internet or on a CD. If the quality is sufficient, you can try to scan the text and apply an OCR software for text recognition.
2. Otherwise, you must enter it with a text editor.
Use an editor that supports ASCII and utf-8 characters.
It is best to use an editor with syntax highlighting. There are text editors that allow the use of text modules. These text blocks can be assigned to common BASIC statements.
3. The line numbers do not need to be aligned. Although they can be separated by spaces and a column of the subsequent program text, they do not require any separation.
4. Ignore spaces outside of strings! You must not enter this spaces.
5. Do not worry about upper and lower case, except within strings (and PC-1421: i, n).
For Pocket Computer that do not support lowercase letters, strings are automatically converted into uppercase.
6. Use shortcuts, for example, 'P.' or 'PR.' for 'PRINT'.
A list of shortcuts can be found for example in the manual of the PC-1475 (or E500).
7. Enter characters as utf-8 characters or with the following transcriptions:
Π [PI], √ [SQR], ¥ [Y], □ [INS], ■ [FUL], old exponent € [E], other hex code ##: [##]
8. Convert the source code into a TAP or Wav file and convert it back into a BAS file to resolve the shortcuts and format the text with spaces.

Enter comments outside lines with leading ' and blank lines at the very end one, because they are lost in the conversion.

12.4. Convert variable data tap format to Transfile PC, edit and return it (PC-1210 to PC-1475)

1. PRINT # variable data inside the commercial pocket emulator to a tap file
2. Export saved tap file to an public accessible directory with the emulator menu
3. Copy the tap file to a personal computer (USB cable or a Webdav server)
4. Open a command line window on your Personal Computer or use a script
5. Run Wav2bin with the options `-tap` and `-type=shc`
This makes a shc file from the tap file.
6. Start Transfile PC plus 5.55 (runs inside DOSbox)
[ESC] [ESC] [F10]
 >Pocket
 >PC type

 >Binary file load
[ESC] [ESC]
Edit the ASCII file: first line includes the filename and dimensions
[F10]
 >Pocket
 >Binary file save
7. Run Bin2wav with the options `-pc=.... -type=shc -tap`
This makes a new tap file from the edited shc file
8. Move and import the tap file to the emulator
The OS file name must be exact identical to the SHARP-file-name.tap !
9. Use INPUT # to read the new variable data inside the emulator

Note: Everytime you have changed the PC type inside "Transfile PC", you have to open/load the source file (SHA) again.

13. Known Limitations

1. Many third-party extensions of the PC-1500 and some unknown tokens are not supported.
Loadable keyword files are planned for a later release.
2. Wave files from BASIC images that contain binary code (poked in), can not always be converted.
3. Special chars in file names are not supported.
4. Third-party fast tape formats are not supported.

14. Thanks to

the original authors

Jean-Francois LAROCHE, Joachim BOHS

the authors who have developed the software in the meantime

Norbert ROLL, Marcus VON CUBE, Manfred NOSSWITZ, Olivier DE SMET

for the debugging and other help

Christophe LE GLATIN, Emmanuel BEAUREPAIRE, Simon LEHMAYR, Christoph LAU,
Manfred BECKER, Edgar PÜHRINGER, Michael (irrlichtproject.de),
Harald RICHTER, Eric MILLESCAMPS

for the support with hardware and information

Raymond BAKKER, Ralf DERSCHIED, Rik HARTL, Rainer KRATZER,
Klaus DITZE, Reto AMBÜHLER, Andrea WEIGERT, Jörg WALENTA,

Horst SCHAEFFER for his batch tool "Wfile"

and all the others who have contributed to it, that this project could be realized

Torsten MÜCKER, developer of this new version, June 2015

This is free software. There is NO warranty, not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE, We are not reliable for any consequences arising from the usage of it.

Use it at your own risk.

All Trademarks, company or product names belong to their respective owners!