WinSCHNIPP™

... is SCHNIPP for Microsoft Windows 3.1.
The name "SCHNIPP" is due to the fact that we (see below what is meant by "we") do our work in analyzing recorded spectra primarily - though not only - by setting cuts within the spectra. The German word for cut is "Schnitt" and "Schnipp" is just something like a slang expression for that.

Please keep in mind:	this documentation is just a one hour's work with the main objective to explain the file format used by WinSCHNIPP™, so that you can use this program with data of your own, which you have to format the right way. So don't expect too much of it and please refer to the help file for information about keyboard commands, menu commands, mouse movement etc.

Disclaimer

Usual disclaimer applies:

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND DOCUMENTATION ARE DISTRIBUTED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE, STABILITY, ACCURACY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED.
BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

WHILE I HAVE MADE EVERY EFFORT TO TEST THE PRODUCT IN A WIDE VARIETY OF OPERATING ENVIRONMENTS, GOOD PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT.

THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM.

ANY LIABILITY OF THE AUTHOR WILL BE LIMITED EXCLUSIVELY TO REFUND OF COMMITTED CONFIDENCE.

Introduction

WinSCHNIPP™ is a program for displaying, handling, analyzing time-of-flight (TOF) spectra within Microsoft Windows 3.1.
These spectra are actually recorded by a group of physicists (where I am a member of) at the "I. Physikalisches Institut" of the "Universität zu Köln".
We operate an apparatus for the investigation of ion-molecule collisions in a crossed-beam system. It consists of a 150 kV ion accelerator providing the projectiles (H^{+}, H_2^{+}, H_3^{+} ions), a molecular beam system creating the gas target, and a time-of flight mass spectrometer to analyze the charged reaction products. The fragments are identified by their flight time µ §, which is measured in the following way: The time of production of the fragments is fixed by pulsing the projectile beam by means of fast high voltage switches. Simultaneously with each projectile pulse a clock (multihit TDC, LeCroy) is started. The charged fragments produced are extracted by the electric field of an ion optical system and conducted to a field free drift tube at whose exit they get detected. The amplified detector signal stops the running clock and the measured time is sorted into a computer recorded spectrum. If more than one fragment originates from a projectile pulse their time-of-flights don't get sorted into a spectrum but are saved as a pair or triple etc. of TOF's in a listmode data memory area. A typical listmode file therefore consists of a spectrum of 65536 channels containing all the fragments that were detected exclusively during one projectile pulse (counts per channel as longint) and several memory areas with pairs or triples etc. of TOF's of type word (unsigned integer), recorded in the order they get detected.
Though we use this program with data that corresponds to TOF's, you surely can use it with any data, where you get values - single or in correlation - that can be formatted in an appropriate way for reading by WinSCHNIPP™. Please refer to the File Format section in this document for further information.
If you have further interest in our work then feel free to contact us (see my address at the end of this manual) or get some of the articles that we've published in the last years:

References:

Production of negatively charged fragments in collisions of swift positive hydrogen ions with molecules
Tybislawski, M., Bends, M., Berger, R.J., Hettlich, M., Lork, R., Neuwirth, W. :Z. Phys. D**?** (1993) (to be published)

An ion optical system for the detection of charged fragments with high acceptance in a time-of-flight mass spectrometer
Lork, R., Bends, M., Berger, R.J., Gassen, D., Schäfer, K., Tybislawski, M., Neuwirth, W.: Nucl. Instrum. Methods B**71**, 330 (1992)

An apparatus for ion-molecule collision experiments via time-of-flight spectroscopy with ion energies of 10 to 150 keV
Baek, W.Y., Gassen, D., Förster, K., Schäfer, K., Neuwirth, W.: Nucl. Instrum. Methods B**58**, 266 (1991)

Analysis of initial energies of fragments produced by 65-keV proton-molecule collisions using a time-of-flight mass spectrometer

Schäfer, K., Baek, W.Y., Gassen, D., Förster, K., Neuwirth, W.: Z. Phys. D**21**, 137 (1991)

Packing List

After unpacking you should find the following files:

bwcc.dll	Windows custom controls from Borland
kelly4ws.bmp	bitmap of Christina Applegate alias Kelly Bundy for displaying as splash screen at start-up of program
n2o003.lft	sample TOF spectrum[1] of the molecule N_2O [2]
readme.txt	quick information about WinSCHNIPP™
winmem32.dll	DLL from Microsoft for handling 32 bit memory (flat memory model)
ws_manul.doc	this file
wschnipp.bmp	about box bitmap
wschnipp.exe	WinSCHNIPP™ executable
wschnipp.hlp	help file for WinSCHNIPP™
wschnipp.ini	sample ini-file (usually placed in your Windows directory)

Installation

Just unpack the Zip-file - what you've already done when you read this - and put all that stuff in a directory called e.g. WSCHNIPP. This holds true except for the two DLLs which have to be placed in the Windows SYSTEM directory. The preferred place for the spectrum is a directory called LMDATEN, though you can put it anywhere but in the root directory (this is due to a known bug with the associated data function). Then add WinSCHNIPP to one of your program manager groups.
If you execute WinSCHNIPP™ without any command line parameters (see help file for details) then you should place the ini file in your Windows directory, otherwise you will be prompted that no ini file has been found. BTW, you can create such a file by selecting Save Options in the Settings dialog box.

System Requirements

First of all WinSCHNIPP™ needs a lot of memory for handling and manipulating spectra. So you should get yourself a fast 386 with 8 MB as a minimum configuration. The exact requirements are as follows:

386 or better

1recorded by Mark Bends
2Hints for calibrating this spectrum: remember that the peak with the biggest mass in the spectrum is N_2O^+, the molecular ion, with mass N+N+O = 14+14+16 = 44. Always visible is also H^+ originating from fragmentation of the water molecule (when something's present at 10^{-7} mbar, it's water) with a mass of 1 right in front of the spectrum

DOS 3.1 or newer (that is the DOS version you need to run Windows)
Windows 3.1 in enhanced mode (won't work with Version 3.0)
winmem32.dll in your Windows system directory

File Format[1]

The most important part of this documentation.

For reasons of flexibility we choose to use a file format that a priori doesn't restrict the type of data being saved in it. Therefore we use a descriptor structure that can hold as different types of data as spectra, listmode data, ASCII texts, record data etc. in one file.

The file starts of with some of these descriptors that are each responsible for one area of data. A descriptor consists of a data record of size 12 bytes that is organized in 3 subunits each of size 32 bit. The first 32 bit value describes the type of data used, the second one the offset in bytes of this area of data with regard to the start of the file and the third one the size of this area in bytes.
The very first descriptor plays a special role in that it doesn't refer to an area of data. This descriptor contains some data that allow for the spectrum program the identification and verification of the data-format. Moreover it gives the total number of following descriptors. The scheme below illustrates the context:

µ §

Definition of descriptors

Start descriptor:

byte 0	**:**	**contains a 'V' in ASCII code**
byte 1-3	**:**	**contain a three-digit ASCII version number**
byte 4-7	**:**	**size of file as longint**
byte 8	**:**	**number of following descriptors**
byte 9-11	**:**	**same as byte 1-3 (repeat of version number)**

V	**1**	**0**	**0**

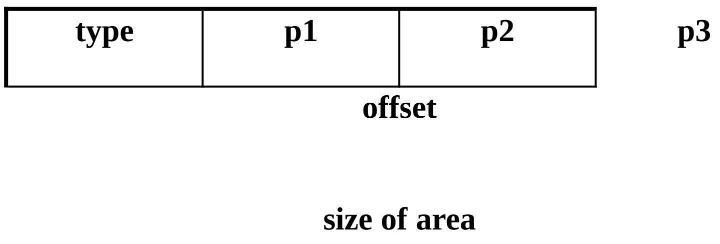
size of file

number of following descriptors	**1**	**0**	**0**

[1]The file format of WinSCHNIPP™ has been developed mainly by Markus Hettlich (see the credits section in the help file for further information)

Generic form of a following descriptor:

byte 0	**:**	**type of the affiliated area of data**
byte 1-3	**:**	**sub parameters of the type of this area of data**
byte 4-7	**:**	**offset of the area of data relative to the start of file**
byte 8-11	**:**	**size of this area of data**



The following tables show the types and sub parameters that have been associated so far (note: this is subject to change, whenever we enhance our apparatus):

type	**description**
0	pointer structure
1	structured data
2	spectrum data
3	listmode data
4	ASCII data

p1	**description**
0	uncompressed data
1	compressed data

Spectrum data

This descriptor points to an area that contains spectrum data where the number of the channel is determined by the order in the area of data.

byte 0	**:**	**2**
byte 1	**:**	**cf. global definition**
byte 2	**:**	**size of a channel in bytes**
byte 3	**:**	**no significance**
byte 4-7	**:**	**offset of the area of data relative to the start of file**
byte 8-11	**:**	**size of this area of data**

2	**p1**	**p2**

offset

size of area

Definition of p2:

p2	**description**
0	1 byte
1	2 bytes
2	3 bytes
3	4 bytes

Listmode data

This descriptor points to an area containing listmode data where the order in the area of data stems from the chronological recording of the measured values.

byte 0	**:**	**3**
byte 1	**:**	**cf. global definition**
byte 2	**:**	**size of a channel in bytes**
byte 3	**:**	**number of fragments**
byte 4-7	**:**	**offset of the area of data relative to the start of file**
byte 8-11	**:**	**size of this area of data**

3	**p1**	**p2**	**p3**

offset

size of area

Definition of p2:

p2	**description**
0	1 byte
1	2 bytes
2	3 bytes
3	4 bytes

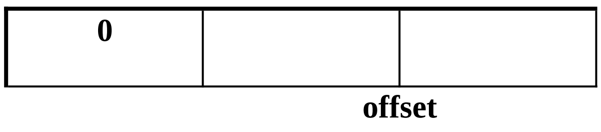
Definition of p3:

p3	**description**
2	2er fragments (pairs)
3	3er fragments (triples)
...	...
8	8er fragments (octuples)

Pointer structure

This descriptor points to another descriptor. This structure is needed in case of an extension of a file.

byte 0	**:**	**0**
byte 1-3	**:**	**no significance**
byte 4-7	**:**	**offset in bytes of the destination descriptor relative to the start of file**
byte 8-11	**:**	**no significance**



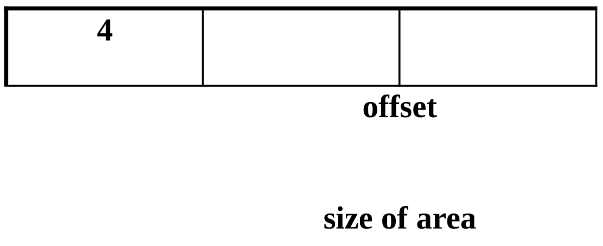
Structured data

This is currently not defined in detail, because we had no use for it so far.

ASCII data

This descriptor points to an area of unstructured text data in ASCII form.

byte 0	**:**	**4**
byte 1-3	**:**	**no significance**
byte 4-7	**:**	**offset in bytes of the destination descriptor relative to the start of file**
byte 8-11	**:**	**size of this area of data**



Compressed spectrum data

The procedure for saving raw spectrum data in a compressed format[1] is as follows:

To be able to save negative values, the first step is that the smallest value in the spectrum is determined and then subtracted from all channels.
To have the possibility to add it on load of the spectrum it is saved ahead of the compressed spectrum data as longint. After subtraction of this value the smallest occurring value within the spectrum equals 0.

Subsequently it is determined channel by channel, whether the contents of the channel equals 0 (0 byte integer), or whether it can be saved as 1, 2 or 3 byte integer value (i.e. as byte, word, "byte+word" - though the internal representation of the channels is longint, longint isn't used here, as 3 byte integer cover a range from -8388608 to +8388607, fairly enough for our needs). Of course you have to know for later reconstruction of the data, how the following bytes have to be interpreted. That is why an information is saved ahead of every sequence of bytes that have to be interpreted in the same way. This information contains the way how they have to be interpreted and how many of them follow. Two cases are distinguished:

1the compression algorithm was mainly developed by Karl Förster and used in his program Display in the late 80's, when the former members of the group were recording spectra using one TAC

A maximum number of 62 values of the same type follow:
one byte of information is saved
byte = i + 4 * number of values (i = 0, 1, 2, 3 byte)
on load i and the number of values are computed as follows:

i	=	byte mod 4
number of values	=	byte div 4

A maximum number of 62 to 256 values of the same type follow:
two bytes of information are saved
byte1 = i (i = 0, 1, 2, 3 byte)
byte2 = number of values -1

If, e.g., the value 0 successively follows 7 times, the byte of information that gets saved is 28, directly followed by the next byte of information. On load the value 0 is automatically inserted 7 times. If, e.g., 400 values of the same type successively follow, then first the information for the first 256 bytes is saved, then these 256 values, then the information for the latter 144 values inclusive their values.

Bug Reports / Contacting the Author / Shareware Notice

If you find this program helpful and are going to use it regularly then please feel free to support my efforts in developing this software by sending me $20 in U.S. funds. I can't promise you to do anything for this money, but the important thing is that it keeps your conscience clear.

Greatly appreciated would also be any bug reports that you can give to me (see address below) - that is if you can find bugs in this program (you sure will if you could ask any member of our group).

Any comments, suggestions or bugs should be mailed to my address (preferably by EMail):

Philipp Schmitz	**Voice:**	**+49 221 424441 (Home)**
Uni-Center 1705		**+49 221 4703495 (Office)**
50939 Köln	**EMail:**	**philipps@sun.ph-cip.uni-koeln.de**

I'm looking forward to hearing from you. Up the Irons!

Thanks to:

Andrea, Claudia, Matthias, Marius, Mark, Marcus, Markus, Martin, Rainer, Rolf-Jürgen, Thomas, Ulli, Volker, Wolfgang for supporting me developing this software.

Revision History

this is going to be grow soon if you have useful suggestions and if I have the time to convert it (currently working hard to get my physics diploma).

0.5.2 first version of WinSCHNIPP to be put on FTP