**********
**********

# Exporting VnmrJ / VNMR Data

**********
**********

Last update: 2005-04-18

---

NOTE: This collection of articles from Agilent MR News focuses on the aspects of exporting VNMR data for PROCESSING with other software, or in a different environment, as well as aspects of importing VNMR data into third-party processing software. Exporting graphics and spectral PLOTS for desktop publishing (DTP) is discussed in a separate document, "faq/plot_export".

---

List of Titles in the Sequence of Appearance in the Section Below:
==========



==========

1993-04-16:

## FELIX FOR WINDOWS:

A new VXR5000.DLL is now available from Biosym which correctly handles rfl/rfp parameters and thus imports correctly referenced VXR-5000 spectra. Contact Biosym at 619-458-9990 if this improvement is important to you. As you

may infer from this item, we at Varian have begun working closely with Biosym to ensure the most accurate handling of Varian data by Biosym products (in this case Felix for Windows).

[ Agilent MR News 1993-04-16 ]

1997-01-15:

EXPORTING DATA IN JCAMP-DX FORMAT:

JCAMP-DX is a data format that was created (originally for IR spectroscopy) to allow for a platform-, software- and media-independent data exchange: JCAMP-DX is pure ASCII and allows for data transfers even on printed paper (supposing you have a scanner). The final definition for JCAMP-DX for NMR was published in 1993 (A.N. Davies & P. Lampen, Appl. Spectroscopy 47(8), 1093 (1993). There are several third party software packages (such as Specinfo, MaxEnt and others) that rely on the JCAMP-DX format for importing spectra (or FIDs).

Varian initially supplied a simple macro for limited JCAMP-DX parameter exporting - but now the on-line user library contains a utility for full JCAMP-DX exporting for 1D FIDs and spectra from VNMR: to obtain a copy, fetch "bin/parhandler" from the on-line user library (from the private Web or FTP site, or using the automatic user library e-mail responder).

"parhandler" now contains three new macros: "svfj" saves a 1D FID (or a trace from an arrayed experiment) as JCAMP-DX file, together with all VNMR acquisition parameters. The macro "svsj" saves a 1D spectrum (or a trace from an arrayed experiment, real/phased part only) as JCAMP-DX file, including all VNMR acquisition and processing parameters. In order to keep the JCAMP-DX file compact, "svsj" converts the phasefile trace to 16-bit integers first. "svlsj" does the same as "svsj", but uses a bigger numeric range in the converted file for spectra with a large dynamic range.

So far, only single 1D traces can be exported into JCAMP-DX format, and there is currently no plan to add a JCAMP-DX data import facility, as VNMR already has the ability to convert Bruker data, and data from further manufacturers can be converted using utilities from the user library.

[ Agilent MR News 1997-01-15 ]

1997-03-18:

EXPORTING DATA FROM VNMR INTO THIRD PARTY SOFTWARE:

Several third-party software manufacturers such as Molecular Simulations (FELIX-PC, which is also available from Varian) have written special data import programs for VNMR data. There are also "general purpose" third party conversion programs, such as V-Helper (by Steve Silber, Texas A&M University, for converting XL/VXR-4000/GEMINI data for NMR processing software such as PCNMR+, FELIX-PC 1.0, Felix 2.0, FTNMR, LabCalc/SpectraCalc/GRAMS386, NUTS).

The VNMR User Library also contains contributions for exporting VNMR data:

- bin/parhandler contains software for exporting 1D FIDs and spectra in JCAMP-DX format. Software such as SpecInfo (Chemical Concepts) relies upon this format for importing NMR (and other) data.
- For the CSEARCH and SearchMaster C13 database software packages (Biorad), a macro "csll" (maclib/csll) was written by Steve Patt.
- For importing C13 line listings (with integration) into the SpecInfo C13 database software (Chemical Concepts), the macro "dlli" can be used. This macro has been available through packages/extend only - we have now put it into a separate contribution (maclib/dlli).

In previous software releases imaging users have had access to a special "FDF" format for exporting data. With VNMR 5.3, FDF has been expanded in capability for any type of VNMR FID (1D and nD), using the command "svfdf" (see the "VNMR Command and Parameter Reference" manual). We plan on further expanding the use of FDF files for data exporting in future VNMR releases.

Besides that, the formats of VNMR parameter files, 1D FIDs and spectra, nD FIDs, and phased 2D spectra are fully documented in chapter 5 of the VNMR User Programming Manual (note the extra information on phased 2D spectra that was presented in VMR News on 1994-06-07!). If you are looking for sample C programs that access VNMR 2D FIDs and spectra, you find several of them in the VNMR user library, inside packages/adv2d.

[ Agilent MR News 1997-03-18 ]

1997-10-09:

FREE NMR PROCESSING SOFTWARE FOR THE PC:

Volker Brecht from the University of Freiburg/Germany found a nice NMR processing package for PCs that is available for free on the Internet, see

http://qobrue.usc.es/jsgroup/js-eng.html

(check "Goodies"). The Program, "MestRe-C" (Magnetic Resonance Companion), was created in the group of Prof. F. Javier Sardina at the University of Santiago de Compostela, Spain. MestRe-C 1.3 imports VNMR and VXR/GEMINI data (so far 1D only) and performs 8 different weighting functions, interactive apodization, FT, phasing, baseline correction, integration and peak picking. The program seems to be comfortable and fast, and exporting spectra from MestRe-C into Word documents is easily possible. MestRe-C runs under Windows 95 or Windows NT and requires at least a 486 CPU (Pentium highly recommended) and 8 MB of RAM.

[ Agilent MR News 1997-10-09 ]

1998-03-26:

USING VNMR WITHOUT GRAPHICS?

It sounds contradictory that in the time of the graphical user interface somebody wants to run VNMR without graphics. However, there are two situations where different rules apply:

- You are connected to a remote systems via very slow connection (such as a dial-up connection): in this situation, starting VNMR may easily take 5 to 10 minutes, and any graphical interaction becomes very slow.
- You may be sitting inside a firewall. Firewalls often allow TELNET, remote logins, FTP, and Internet browsing (HTTP connections) to systems outside the firewall - but typically, remote X Windows operation is not possible because the return connection is not allowed to cross the firewall.

It is little known that in both these situations VNMR offers a useful option, non-graphical operation! For one, after a simple login, you can always type

```
showstat
```

to see what would normally be displayed in the Acqstat window. For this you often don't even need to log in remotely:

```
showstat <host_name>
```

may also work directly, depending on the networking setup.

But VNMR offers much more: when remotely logging in (via telnet or rlogin), you can specify "d" (dumb) as the terminal type, followed by "vn" (without "&") to start VNMR. Alternatively you can specify the usual (proper) terminal type and then call

```
setenv graphics none
vn
```

Note that in both cases we start VNMR in foreground! The second option is supposed to keep the terminal definition such that editing with "vi" still works. With this call, VNMR will operate in non-graphical mode:

- the user gets a shell-like prompt (">"), the VNMR command line, on which ALL VNMR commands (built-in commands, macros, including commands that start or stop acquisitions, or plotting commands) can be called;
- any output that would normally be shown on the top line ("FT", "PH", etc.), error messages and other output on line3, as well as any output to the alphanumeric ("old dg") window is shown like ASCII output from shell commands, followed by a new prompt;
- graphics and menus (including the function keys) are suppressed.

Sure, you can't see any spectra - but in many cases you can get to the information you need - and compared to remote X displays, this is well over 100 times faster! It may some times even be useful to export the current FID or spectrum into an ASCII file (for FIDs use the VNMR "writefid" command, for spectra use either "maclib/writespec" or "maclib/writetrace" from the VNMR user library), or to plot to a file (PostScript or HP/GL), followed by a transfer of these exported data: this way, you can at least look at the NMR data off-line.

[ Agilent MR News 1998-03-26 ]

1998-09-11:

MORE FREEWARE FROM ACD SOFTWARE:

Among other (commercial) new software for the analytical (NMR) laboratory,

ACD software has just announced freeware for viewing and printing NMR spectra. NMR Viewer allows the import and display of JCAMP-DX, Varian FIDs and spectra (phasefiles), Bruker files, NUTs files, Galactic SPC format and ACS/ESP files. You can reference, integrate, and annotate the data, draw structures, and export the data in a variety of formats. For more information visit ACD software at

http://www.acdlabs.com/

(Note that FIDs cannot be processed within NMR Viewer, but just viewed.)

[ Agilent MR News 1998-09-11 ]

1998-09-18:

USING LimNET ON SUN ULTRA COMPUTERS:

By default, the LimNET software in VNMR 6.1A is looking for Ethernet devices named le0 (le1) or ie0 (ie1). When trying to start the LimNET daemon (using the command "limnetd&") on PCI-based Ultra workstations (Ultra 5, 10, 30, or 60) or Ultra 1 computers with Creator graphics (Ultra 1/140E, 1/170E, 1/200E), LimNET aborts with an error message

failed to open /dev/le: no such device or address

The reason is that these computers use "hme" (hme0, hme1) as the device name for their fast (10/100BaseT) Ethernet ports. To run the LimNET daemon on these computers, type

```
cd /etc
./limnetd /dev/hme0&
```

The same argument must be specified with the "eaddr" command for extracting the Ethernet hardware address:

```
/vnmr/bin/eaddr /dev/hme0
```

To make LimNET automatically start with the correct port address at bootup time, change line 10 in /etc/rc3.d/S21rc.limnet to

```
/etc/limnetd /dev/hme0 &
```

Remember that in order to be able to export data from the GEMINI or VXR-4000 system to the Sun you need a directory with universal write permission, e.g.:

```
chmod 777 /export/home/vnmr1/incoming
```

[ Agilent MR News 1998-09-18 ]

1999-06-04:

SwaN-MR - FREE AND YET POWERFUL NMR PROCESSING SOFTWARE FOR THE MAC:

The editor has tested a very nice processing for the Mac, SwaN-MR, written by Giuseppe Balacco at Menarini in Florence, Italy. SwaN-MR has been around for a number of years and has already found many users who like it. SwaN-MR can import GEMINI/XL/VXR, VNMR, Spinsight, JEOL, GE/Nicolet, Bruker data, ASCII format (and Nuts or Felix data); in principle it can process data up to 4D, with a strong emphasis on 1D/2D NMR. SwaN-MR is available via

http://qobrue.usc.es/jsgroup/Swan/index.html

and it's FREE! Even though it's free, support is available from the author at

gbalacco@menarini-ricerche.it

The installation of the package is simple (trivial compared to a typical PC software installation), and you get exactly three files: SwaN-MR itself, the SwaN-MR on-screen manual (also very user-friendly!), plus a very nice little interactive utility that calculates 3J(H,H) coupling constants as a function of the dihedral angle, with the possibility to change substituents and to adjust the constants in the Karplus formula.

The editor tested 1D and 2D processing (absolute value and phase-sensitive) and found the package to be quite easy and intuitive to use after a short learning period. Note that the author wrote SwaN-MR as a research tool for himself, not primarily for the casual user. SwaN-MR comprises (among other features)

- interactive FID display;
- weighting (interactive and via direct parameter entry): exponential, Gaussian, shifted straight and squared sinebell, Hanning and other;
- FT and linear prediction for 1D and 2D (quite fast, actually);
- interactive 1D and 2D spectrum displays (color intensity maps and contour displays with selectable colors);
- interactive display and phasing both for 1D AND 2D spectra (well, you want a fast Mac for interactive 2D phasing!);
- autophase, baseline correction;
- spectral annotation;

- plotting, copying to the clipboard, plotting to (PostScript) files, taking PICT snapshots, exporting to HP/GL format;
- you can open several datasets at the same time, in different windows;
- spin simulation for up to 8 closely coupled, non-equivalent spin species, plus one species with X-approximation (this one can even be spin >1/2); you can interactively change the spin system parameters and watch how it affects the result!

SwaN-MR can also handle arrayed 1D spectra and stacked displays - but it currently isn't very good at handling DEPT data and T1 analyses. Other limitations: VNMR 3D data and imaging data sets can currently not be directly converted to the SwaN-MR format

The system requirements: any PowerPC-based Mac should do, for 2D processing you want to have enough RAM to keep the 2D data set in memory (unlike VNMR and other workstation packages, SwaN-MR doesn't use submatrices). The editor used SwaN-MR on a PowerMac 8500/180 with 48 MBytes of RAM. A 1k x 1k 2D FT (2k x 2k in VNMR terms) took about 1 + 4 seconds (f2 and f1, respectively), even 2k x 1k (4k x 2k in VNMR terms) could be handled without problems on that system.

[ Agilent MR News 1999-06-04 ]

1999-06-11:

SOME HINTS ON USING SwaN-MR:

Last week (Agilent MR News 1999-06-04) we reviewed SwaN-MR, a free NMR processing package for the Mac. As a VNMR user, you need to first get used to some of the SwaN-MR features, for instance:

- to open a VNMR FID, you need to first open the VNMR *.fid folder, then import (convert) the "fid" file. SwaN-MR then by default creates a SwaN-MR file "fid.swan" in the same directory. You can also define a different suffix, or select to take over the name of the parent directory. The latter option makes sense, as then the header bar in SwaN-MR window will show something more meaningful than just "fid.swan".
- the proper protocol for f1 processing of phase-sensitive (States) VNMR 2D data is "hyper-invert".
- to re-process a 2D spectrum, select "Repeat" (Command-R), then you can do f2 and f1 weighting ("apodization") and transforms in steps or all in one step, using "deep FT" (using the previously defined weighting functions and FT parameters).
- by default, SwaN-MR will show a ppm scale in f2, but points in f1. To change the default select the preferred scale units, colors and anything else you like, and choose "Save as Default" from "View". It will ask you for a name for a file. Answer "Cancel".
- select contour display only in the final stages of 2D processing. If you happen to have "Contour" selected and you switch back to the (2D) FID, SwaN-MR will take a LONG time to display the FID - you can abort this with <Cmd-.>. By default, SwaN-MR uses the "Blind flight" option for the 2D FID and interferogram stages.

[ Agilent MR News 1999-06-11 ]

1999-11-25:

VNMR DATA CAN NOW BE IMPORTED INTO "CORE":

In Agilent MR News 1999-03-06 Peter Stilbs (Institute of Physical Chemistry, Royal Institute of Technology, Stockholm) announced the availability of his CORE (COmponent-REsolved [spectroscopy]) software via his Web home page at

http://omega.physchem.kth.se/~peter/

"CORE was primarily designed to analyze FT-PGSE data, but it can also be used for kinetic studies and various spin relaxation experiments. CORE processing is based on a simple, global, least-squares approach, using a two-level minimization scheme. No assumptions whatsoever are made with regard to component bandshapes - the iteration is made on the (global) kinetic rate constants of the data set only. The basic method is described in P.Stilbs, K.Paulsen & P.C. Griffiths, J. Phys. Chem. 100, 8180 (1996) and P.Stilbs & K.Paulsen, Rev. Sci. Instrum., 67, 4380 (1996)."

Thanks to the efforts of Leo Timmermans (AGFA, Mortsel, Belgium), CORE can now also import native VNMR data (previously you had to convert Varian data into ASCII files). The import software (including source code, scripts and user instructions) can also be downloaded from Peter Stilbs' home page.

CORE is now available for the Intel/DOS, Alpha NT, and Alpha OpenVMS

platforms (graphics output requires Matlab).

[ Agilent MR News 1999-11-25 ]

2000-11-04:

EXPORTING VNMR FIDS INTO ASCII FILES:

Third party and user-written processing software often has no import function for binary VNMR data. Note that the formats for 1D and nD FIDS and phased spectra ("~/vnmrsys/expn/datdir/phasefile"), as well as for VNMR parameter files ("~/vnmrsys/expn/curpar", "~/vnmrsys/expn/procpar") is completely and accurately described in the VNMR User Programming Manual, Chapter 5, "Parameters and Data".

On the other hand, external software often accepts ASCII files for importing NMR data, even though this is not a very efficient way to store NMR data. VNMR has a utility to export 1D FIDS or individual FID traces from nD data sets into ASCII files:

```
writefid(file_name<,element_number>)
```

This produces a file with one complex pair per line, e.g.:

```
-42677  -66895
-127  -86824
56747  -5979
...     ...
```

Note that if the FID was acquired with "dsp='i'", "writefid" produces output with fractional numbers (with in-line DSP, the FID is stored in 32-bit floating point format).

VNMR can also IMPORT individual 1D FIDs from such files, using

```
makefid(file_name<,element_number<,format>>)
```

If you already have a (template) FID file in your current experiment, this will simply substitute the FID (or the specified nD trace) by the imported data. If the format (precision) or the number of data points in the imported data does not correspond to the FID in your current experiment, you should remove the FID using

```
rm(curexp+'/acqfil/fid')
```

and use the "format" option ('dp=n' or '16-bit' / 'dp=y' or '32-bit') instead. If the ASCII file contains fractional numbers (from "dsp='i'", see above), either use the "template method" with a FID acquired with "dsp='i'", or specify "'float'" as format option.

Note that "writefid" output contains the "naked" FID only - there are no parameters attached! The user library contains a contribution

```
bin/parhandler
```

with a command "svfj" that permits exporting 1D FIDs into JCAMP-DX format, see also Agilent MR News 1997-01-15. JCAMP-DX is a format for data exchange which includes both parameters (in the file header), as well as data (in the body of the file, using a complex "compressed ASCII" format). Note that there is no VNMR import facility for JCAMP-DX data at this point in time.

[ Agilent MR News 2000-11-04 ]

2000-11-11:

EXPORTING VNMR SPECTRA INTO ASCII FILES:

Last week (Agilent MR News 2000-11-04) we described how 1D FIDs can be exported into ASCII files. For exporting (1D, phased) spectra into ASCII files you can use several user library contributions:

- "maclib/writespec" contains a macro for writing an ASCII file (Y only) of the DISPLAYED portion of the current spectrum or 2D trace (f1 or f2). The syntax is

  ```
  ds writespec('filename')
  ```

  for nD spectra use "ds(n)".
- "maclib/writetrace" contains two macros: "writetrace" is much faster than "writespec" (particularly for large 1D data sets), but it always writes the ENTIRE 1D spectrum or 2D trace (f1 or f2). The syntax is

  ```
  writetrace
  ```

  or (for 2D or arrayed 1D spectra)

  ```
  writetrace(trace_#)
  ```

  The resulting ASCII file is stored inside the current experiment; for 1D spectra it is named "trace.#" (where # is 1 or the trace number), for 2D spectra the filename is "f1trace.#" or "f2trace.#", depending on your current "trace" selection.

- The second macro in "maclib/writetrace" is "writexy", which has the same syntax as "writetrace", but add a second output file with X,Y pairs (referenced frequency in Hz, intensity in mm). The output files are named "xytrace.#" (1D, arrayed 1D), or "f1xytrace.#" / "f2xytrace.#".

In all these cases, the output contains JUST the spectral data, NO parameters. The contribution "bin/parhandler" contains two macros "svsj" and "svlsj" that save a (complete) 1D spectrum or individual 2D traces in a JCAMP-DX file (for JCAMP-DX see Agilent MR News 2000-11-04 and 1997-01-15), complete with all relevant parameters and the spectrum in "compressed ASCII" format (the data part is only readable by software that understands the JCAMP-DX format). "svsj" scales the Y-axis to 16-bit (integer) digital precision, "svlsj" is for spectra with a larger dynamic range - it uses 24-bit integer precision. As with JCAMP-DX FID data, there is no VNMR import facility for spectra in JCAMP-DX format (nor for spectra in other ASCII formats).

[ Agilent MR News 2000-11-11 ]

SAVING A SUBSET OF TRACES FROM A 2D FID AS 1D OR ARRAYED DATA SET:

On rare occasions, users want to save a single FID trace from an nD or array data set as an individual data set. This can be done using the "add" command:

```
df(n) clradd add jexp5
```

at this point you need to manipulate the parameters in the "processed" tree in exp5 such that they match a single 1D data set, e.g.:

```
df $ct=ct ni=0 phase=0 setvalue('ct',$ct)
groupcopy('current','processed','acquisition')
```

The "df" command copies the "processed" tree over the "current" one (the "add" command copies the entire parameter set from the (first) source FID into exp5 (the "addsub buffer"), so you will now see the parameters for a 2D data set. Now we first "remember" the value of "ct" by storing it in a temporary variable "$ct" because any change in acquisition parameters will immediately reset "ct" to zero (you can create and use arbitrary new "local" parameters the same way as in VNMR macros). Then we reset the parameters defining the nD experiment or array. For the latter case you would reset the arrayed parameter to the value for the selected (nth) array element, e.g.:

```
df(n) clradd add jexp5
df $ct=ct d2=d2[n] array='' setvalue('ct',$ct)
groupcopy('current','processed','acquisition')
```

After changing the 2D or array parameters you can restore "ct" using the "setvalue" command ("ct" is not directly enterable), then we copy the current tree back over the processed one (which remains associated with the FID), then we can save that 1D FID using "svf".

If what you want to save is not a single 1D FID, but maybe a subset of the array or nD experiment, you can continue adding traces using "add('new')" :

```
df(i) clradd add
df(j) add('new') df(k) add('new') ... jexp5
```

and then of course adjust the parameters for the new array. For nD experiments we only need to adjust the nD incrementation parameters such as "ni". For arrays we need to first reset the array and then build a new array from the appropriate subset of elements, e.g.:

```
df $ct=ct $d2=d2
d2=$d2[i] d2[2]=$d2[j] d2[3]=$d2[k] ... calcdim setvalue('ct',$ct)
groupcopy('current','processed','acquisition')
```

Here, we store an entire array in a local variable! the first assignment to "d2" then resets the array (as we specified "d2" without array index), the following assignments build up a new array. With "calcdim" we then make sure "arraydim" is adjusted properly before we restore "ct" and do the "groupcopy".

For saving individual FID traces as ASCII files see the article in the last issue (Agilent MR News 2000-11-04).

[ Agilent MR News 2000-11-11 ]

2001-05-14:

FREE PC PROCESSING SOFTWARE FOR VARIAN DATA:

In Agilent MR News 1997-10-09 we posted a short note on the "MestRe-C" free PC processing software that can import VNMR data (VXR, GEMINI and XL data formats are not supported). A few weeks ago, Kirk Marat (University of Manitoba, Canada) posted an article through AMMRL, in which he announced the availability of SpinWorks 1.2, a free NMR processing, analysis and simulation

software for the PC, at
ftp://pauli.chem.umanitoba.ca/pub/marat/SpinWorks
Kirk Marat writes: The file is "SpinWorks1_2.zip" (2.7 MB, use "WinZip" or equivalent to extract the files). Complete documentation is included in the distribution (as "SpinWorks.doc"), but can also be downloaded separately in Word and Adobe PDF formats. To install, run "Setup.exe" in the unzipped distribution and follow the instructions. In most cases, using the defaults is fine. The package provides the following features:

- Basic 1D processing and printing for Varian, Bruker, JEOL and TECMAG data, including stacked display, integration, phasing, base-line correction, peak picking, etc.
- Spin simulation with iterative parameter adjustment. Equivalence factoring, X factoring (weak coupling), automatic assignment/deletion, and dipolar coupling are included. SpinWorks uses the NUMARIT algorithm as described in J.S. Martin & A.R. Quirt, J.Magn.Reson. 5, 318 (1971), modified and re-implemented in C++) at the University of Manitoba.
- An implementation of Ray Freeman's HOGWASH resolution enhancement (see J.Magn.Reson. 76, 476 (1988); 56, 463 (1984); 56, 294 (1984)).
- dynamic NMR simulation using both the venerable old DNMR3 program and Alex Bain's much newer MEXICO program; for more information on MEXICO see
http://www.chemistry.mcmaster.ca/faculty/bain/

[ Agilent MR News 2001-05-14 ]

2001-06-06:

BETTER DIRECTORY SORTING THROUGH IMPROVED FILE NAMING:

Several VNMR utilities create file names that have the creation date embedded in the name string. For instance, lines 38/39 of the VNMR macro "/vnmr/maclib/walkup" read

```
shell('date +%d.%m.%y'):$date
autodir=$dir+'/auto_'+$date
```

This leads to file names such as

```
auto_05.31.01
auto_06.01.01
auto_06.02.01
auto_07.03.99
auto_12.23.00
```

There are several potential problems with this naming scheme

- as shown above, "ls" doesn't automatically sort chronologically across the years (it's OK within the same year)
- the scheme is not year 2000 compliant, i.e., for certain dates it is not clear which part of the name represents the day of the month, etc.
- multiple dots may cause problems when exporting such data to Windows-based systems, or when burning an ISO 9660 CD-R disk.

This can easily be remedied by changing the first of the above macro lines to

```
shell('date +%Y-%m-%d'):$date
```

which will instead create names with ISO 8601 (yyyy-mm-dd) date format (see also Agilent MR News 1999-01-07), and the above names would list as follows:

```
auto_1999-07-03
auto_2000-12-23
auto_2001-05-31
auto_2001-06-01
auto_2001-06-02
```

[ Agilent MR News 2001-06-06 ]

2001-06-09:

ADDITIONAL GLP FEATURES AVAILABLE FOR VNMR USERS:

In the context of GLP it is desirable to have clear references about the user ("owner") who acquired a data set and about the system which was used to generate the data ("origin", covering system type, a unique identifier, such as the host-ID, the software version, release date and revision/patch level, as well as the operating system version). The VNMR user library contribution "maclib/glp" which is available at
http://www.varianinc.com/products/nmr/apps/usergroup/userlib.html
has been modified to automatically create and fill two new VNMR parameters "origin" and "owner" when an acquisition is started. These parameters are defined as follows:

a) "origin":
- Instrument type and frequency, e.g., "UNITY INOVA-500"
- The (Sun) host-ID as obtained by the UNIX "hostid" command, e.g., "HostID 80f86b0b". This guarantees a clear and unique identifier that ties the acquisition to a particular Sun host workstation (and cannot be tampered with by the user). The instrument serial number is currently NOT available from inside the software (there are many older instruments that don't even have a proper serial number).
- The VNMR version and revision date, e.g., "VNMR 6.1C 2000-08-18"
- The current VNMR patch level (if applicable), e.g., "patched all103"
- The operating system, e.g., "Solaris 2.6", "Solaris 8", etc.

b) "owner":
- The minimal form is the UNIX user name and the UID/GID info.
- If the system administrator has filled the "GECOS" field in "/etc/passwd" by entering the user's real name when defining a new user with the Solaris "admintool", that real name is extracted by using "finger", and added to the parameter "owner".

Examples:

```
origin: MERCURY-300; HostID 8dc86d0f; VNMR 6.1C 2000-08-18 Solaris 8
owner:  John Doe (vnmr1); UID/GID: 1001/101
```

Varian's JCAMP-DX exporting software (available from the VNMR user library in the contribution "bin/parhandler") has also been modified to use these two new parameters, if present in the VNMR data set that is to be converted.

[ Agilent MR News 2001-06-09 ]

2002-04-06:

"FIXING" THE FID FORMAT AFTER ADDITION / SUBTRACTION:

Independent of the format of the source data - which can be 16-bit integer ("dp='n'"), 32-bit integer ("dp='y'"), or 32-bit floating point ("dsp='i'") - the addition or subtraction of FIDs in VNMR ALWAYS results in floating point data in exp5, in order to avoid math overflow problems. Note that "add", "sub" and "addi" may involve arbitrary (fractional) FID (component) multipliers.

Even though such floating point FIDs are fully within the VNMR data format definition (as explained in chapter 5 "Parameters and Data" of the VNMR User Programming Manual), there are apparently third party software packages that can import 1D VNMR FIDs in integer format, but fail with VNMR FIDs in floating point format. In VNMR, you can check what format the current FID is, by typing

```
ddff(1)
```

A floating point FID will be indicated by fractional numbers in floating point notation, the header info indicates "4 bytes per point", and the status typically is 25 = 1 (data) + 8 (floating point) + 16 (complex data), see the VNMR User Programming Manual for more information.

VNMR does not have a standard solution for converting floating point FIDs back into integer format - but we found a solution that works: after using "add" (or acquiring a FID with "dsp='i'), join the relevant experiment (exp5 in the case of "add" / "sub"), then write out the FID as an ASCII file:

```
writefid('fid_float.txt')
```

That text file contains the numeric of the FID, one complex pair per line:

```
25.815  -138.39
176.473  122.165
-3.095  -24.6975
...     ...
```

For the next step we need to find the maximum value in this FID. You can do this by visually inspecting that text file with any editor, or by using "nawk" in a UNIX shell as follows:

```
cat fid_float.txt | nawk 'BEGIN {max=0} \
        {if ($1>max) max=$1; if (-$1>max) max=-$1; \
         if ($2>max) max=$2; if (-$2>max) max=-$2} \
        END {print max}'
```

Now, you can create a 32-bit integer format ASCII file from this:

```
cat fid_float.txt | \
   nawk '{printf("%d %d\n",$1/MAX*(2^31-1),$2/abs(MAX)*(2^31-1))}' \
   > fid_int.txt
```

Alternatively, you could pack this into a shell script "~/bin/fixfid":

```
#!/bin/sh
max=`nawk < $1 'BEGIN {max=0}
  {
    if ($1>max) max=$1
```

```
            if (-$1>max) max=-$1
            if ($2>max) max=$2
            if (-$2>max) max=-$2
          }
          END {print max}'`
        nawk < $1 '
          {
            printf("%d %d\n",$1/'$max'*(2^31-1),$2/'$max'*(2^31-1))
          }'
```

Make sure the script is executable and present in the C shell command list:

```
        chmod 755 ~/bin/fixfid
        rehash
```

now you can do the conversion with

```
        fixfid fid_float.txt > fid_int.txt
```

Now, the conversion is simply a matter of "re-importing" that integer ASCII FID into VNMR:

```
        rm(curexp+'/acqfil/fid')
        makefid('fid_int.txt')
```

and then you can save the integer FID using "svf". Deleting the FID file in the current experiment is necessary for two reasons:

- if the FID file exists, "makefid" does not alter the FID headers, i.e., the headers would in this case still indicate floating point data;
- to ensure that the original data are not altered through "makefid", see also Agilent MR News 1997-09-19.

With "ddff" you can verify that the status field in the FID headers now is changed to 21 = 1 (data) + 4 (32-bit word length) + 16 (complex data).

If you prefer 16-bit integer data, you DON'T need to change the above shell script, but you can use "compressfid" in VNMR to downscale and reformat the 32-bit integer FID.

[ Agilent MR News 2002-04-06 ]

2005-04-18:

IMPORTING BINARY VnmrJ / VNMR FIDS INTO THIRD PARTY SOFTWARE:

Occasionally we receive inquiries about problems with importing VNMR / VnmrJ data into third party software, sometimes claiming that we have altered the FID format without proper notice or documentation. The editor thought that this is worth a clarification.

The key to importing VNMR / VnmrJ FIDs is that in order to determine the format and the structure of the data importing software should look at the BINARY FID file ("*.fid/fid"), NOT at the parameter file ("*.fid/procpar"), even though the access to the latter (an ASCII file) may appear to be more straightforward. Detailed information on the Varian data format is found in chapter 5 "Parameters and Data" of the User Programming Manual - the article below describes the key features of the binary Varian FID format.

Why should importing software not look at VnmrJ / VNMR parameter, but rather focus on decoding the binary FID file? There are several issues:

- The Varian parameter file ("*.fid/procpar") primarily describes how a data set was acquired NOT how it is saved on disk: e.g., the parameter "dp" may be set to "'y'", indicating an acquisition with "double precision" (32-bit) word length; however, a user may then decide to save that FID (particularly in the case of large nD FIDs) as single-precision (16-bit) data by applying "compressfid". In this case, the parameter set will still contain "dp='y'", indicating how the data were acquired - but in apparent disagreement with the actual data format.
- There are certain parameter settings such as "fsq='y'" (UNITY INOVA) or "dsp='i'" (any system) that currently cause FIDs to be saved in 32-bit floating point format - and there may be other post-acquisition processing steps that also cause a conversion to floating point format (e.g., "add" and "sub", possibly user-written functions) - WITHOUT annotation in the parameter set. We are NOT sloppy in our "parameter bookkeeping" - such annotation is NOT NEEDED, as the FID format is COMPLETELY and ACCURATELY documented inside the binary data file itself, see also the article below.
- There have been cases where nD and arrayed experiments were aborted by the user, or through a hard- or software failure of some kind, and where the parameter "celem" was not updated (see bug "celem.6101"), possibly leaving a disagreement between the parameter set and the number of blocks in the actual FID data.

The conclusion: to determine the data structure and type clearly it is far

safer for importing software to look at the binary data file and its headers (see the article below), rather than trying to extract this information from the parameter file.

The parameter file ("*.fid/procpar") of course is still required for data importing, as it contains all relevant information about the experimental conditions and the variables controlling the pulse sequence. A comprehensive discussion of the format of the VnmrJ / VNMR parameter files is also found in chapter 5 ("Parameters and Data") of the User Programming Manual.

[ Agilent MR News 2005-04-18 ]

VnmrJ / VNMR FID DATA FORMAT:

With very minor exceptions (which are irrelevant in the context of data importing) we have NOT changed the binary data format since the introduction of VNMR almost 20 years ago, and this data format is described exhaustively in section 5 "Parameters and Data" of the User Programming Manual. For FIDs, the key is that FID files have a 32-byte data header with the following structure:

```
struct fileHeader
{
  long nblocks, ntraces, np, ebytes, tbytes, bbytes;
  short vers_id, status;
  long nblockheaders;
};
```

where the elements are defined as follows:

- "nblocks" is the number of data blocks in the file,
- "ntraces" is the number of (FID) "traces" per block (high resolution spectra with "nf=1" have one FID / "trace" per block, multi-echo FIDs such as imaging FIDs acquired in "compressed" mode may have multiple FIDs / "traces" per block),
- "np" is the number of (real, not complex) points per "trace" (typically equal to the "np" parameter),
- "ebytes" is the number of bytes per "element" (2 for 16-bit data, 4 for 32-bit data),
- "tbytes" is the number of bytes per "trace", i.e.,
  `fheader.ebytes * fheader.np`
- "bbytes" is the number of bytes per block, i.e., in "C speak":
  `fheader.ebytes * fheader.np * fheader.ntraces + sizeof(struct blockHeader)`
  ("struct blockHeader" is discussed below).
- "vers_id" may be used for the VnmrJ version ID (you may find this set to zero - it's not really used in our software);
- "status" indicates the status of the data; the value of "status" is a flag field similar to the protection bits in the VNMR parameter definition:
  - bit 0 (0x1) indicates the presence of data
  - bit 1 (0x2) would indicate "spectrum" (as opposed to "FID")
  - bit 2 (0x4) indicates 32-bit (as opposed to 16-bit) data
  - bit 3 (0x8) indicates (32-bit, IEEE format) floating point data (in which case the value of bit 2 is irrelevant)
  - bit 4 (0x10) indicates complex (as opposed to real) data
  - bit 5 (0x20) indicates hypercomplex data (transformed nD data only)
  - bit 6 (0x40) is unused up to VnmrJ 1.1D

  for the other bit fields see the User Programming Manual. In a typical VnmrJ 1.1D FID you may find the "status" field to have a value of 0x15, i.e., 0x1 (data present) + 0x4 (32-bit, i.e., "dp='y'") + 0x10 (complex)
- "nblockheaders" is the number of block headers (always 1 for FIDs)

The block header that follows has a size of 28 bytes and is defined as

```
struct blockHeader
{
  short scale, status, index, mode;
  long ctcount;
  float lpval, rpval, lvl, tlt;
};
```

Where the elements have the following meaning:

- "scale" is the number of downscaling operations (non-circular, bit-wise right-shifts, or divisions by 2) while accumulating data (particularly when acquiring with a very large "nt" and "dp='n'") in the given block (see the article "How Does the Autogain Work?" in Agilent MR News 1999-09-11);
- "status" in FIDs is usually the same as in the file header, see above;
- "ctcount" is the value of "ct" for the associated FID block;

 - "lvl" and "tlt" are used to store the d.c. offsets (as determined via the
   noise sampling prior to acquisition) in channels A and B in the case of
   "nt=1", see Agilent MR News 2001-02-02.

For explanation on the other fields see the User Programming Manual.

  The first block header is followed by the first data block, and each of the subsequent data blocks is preceded by its own block header (or the number of block headers in special cases of processed nD data, as indicated in the file header). In ALL cases - and most certainly in the case of FIDs - the structure of the binary data (FID) file, i.e.,

 - the number of blocks,
 - the number of traces (FIDs / echoes) per block,
 - the number of "elements" (numbers) per trace (FID), and
 - the format of an element (32-bit IEEE floating point vs. 32-bit integer vs.
   16-bit integer)

is COMPLETELY and ACCURATELY described in the data headers. The byte order is the one used with the Motorola MC68xxx, the Motorola PowerPC (PPC) and Sun's SPARC family of processors, even if the data were acquired using VnmrJ_LX on a system with an Intel / Linux based PC as spectrometer host, see Agilent MR News 2005-01-22.

[ Agilent MR News 2005-04-18 ]

==========