Agilent VnmrJ 3.2 Command and Parameter Reference Guide

Agilent Technologies

Notices

Manual Part Number

91000171b

Edition

Rev. B, August 2011

Printed in USA

Agilent Technologies, Inc.
5301 Stevens Creek Boulevard
Santa Clara, CA 95051 USA

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Contents

Notational Conventions

A

B

C

D

E

F

G

H

I

K

L

M

O

P

Q

R

S

T

V

W

X

Y

Z

Notational Conventions

The *VnmrJ Command and Parameter Reference* describes in detail the commands, macros, and parameters in VnmrJ software. Information new to VnmrJ in this version is shown by a change bar (as shown to the left of this paragraph).

Title line codes

Each entry has a letter in parentheses in the title line that identifies the type of entry:

(C)	VnmrJ command
(M)	VnmrJ macro command (from the `maclib` directory)
(O)	MAGICAL programming operator
(P)	VnmrJ parameter
(U)	UNIX command (not executable within VnmrJ)
(C,U) (M,U)	Executable from UNIX or VnmrJ (note that syntax is different)

Applicability

An entry with applicability information applies only to the system or accessory listed. If the entry does not include applicability information, the entry applies to all systems.

Command and macro syntax

Each command and macro entry includes the syntax used when entering it into the system. The following examples illustrate this syntax:

`halt`

If no parentheses are shown, enter the command or macro exactly as shown, e.g., enter `halt`.

`delexp(exp_num)`

If parentheses are shown, enter the command or macro name as shown, but replace arguments with a value, e.g., if `exp_num` is `5`, enter `delexp(5)`.

`rttmp(file)`

Arguments can be a string (e.g., name of file or solvent), number, variable, or parameter (e.g., `pw`),. If a string, enclose it with single quote marks, *e.g., if* `file` *is* `samp02`, *enter* `rttmp('samp02')`.
If number, variable, or parameter, do *not* use marks.

`rl<(frequency)>`

Angle brackets (< and >) indicate optional input, e.g., if `frequency` not needed or the default value of `frequency` is acceptable, enter `rl`, but if `frequency` has a value such as 10, enter `rl(10)`.

`md(<from_exp,>to_exp)`

Arguments can also be optional. Use a comma to separate arguments, e.g., `md(2,3)`. Unless stated otherwise, the order of arguments is often important.

`nll<('pos')>`

A keyword is frequently used as an argument. In the syntax, keywords are shown in single quotes and are entered exactly as shown, e.g., to use the optional keyword `'pos'` for `nll`, enter `nll('pos')`.

`dc2d('f1'|'f2')`

A vertical bar indicates an OR condition, e.g., either `'f1'` or `'f2'` can be an argument to `dc2d`.

`sin(angle)<:n>`

Some commands return values to a calling macro. This is shown by a colon followed by one or more variables, e.g., if `angle` is variable `x` and `n` is variable `rt`, then `sin(x):rt` returns the value of `sin(x)` to the calling macro via the variable `rt`.

`z(reset1,reset2,...)`

Three dots indicate the sequence of arguments continues. Unless a limit is given, you can enter one argument, two, three, or as many as needed.

Parameter Syntax

Parameter syntax is always in the form `parameter_name=value`. If `value` is a string, enclose it in single quote marks; otherwise, no marks are used, e.g., `auto='y'`, `plotter='ThinkJet'`, `spin=5`. Note that some parameters are not user-enterable.

Notational Conventions

Throughout all Agilent NMR manuals, `typewriter-like characters` identify commands, parameters, directories, file names, and text displayed on the screen.

Because pressing the Return key is required at the end of almost every command or line of text you type on the keyboard, assume this use of the Return key unless stated otherwise.

Other Sources of Information

For further information about an entry, refer to the manual listed under "See also." For general coverage on VnmrJ, refer to the following manuals (each manual is also online):

Automation User Guide
Spectroscopy User Guide
VnmrJ Installation and Administration
VnmrJ Imaging User Guide

A

aa	Abort acquisition with error (C)
abort	Terminate action of calling macro and all higher macros (C)
abortallacqs	Reset acquisition computer in a drastic situation (C)
abortoff	Terminate normal functioning of abort in a macro (C)
aborton	Restore normal functioning of abort in a macro (C)
abs	Find absolute value of a number (C)
AC1S-AC11S	Autocalibration macros (M)
ACbackup	Make backup copy of current probe file (M)
acct	Writes records for operator login and logoff (M)
ACreport	Print copy of probe file after autocalibration (M)
acos	Find arc cosine of number (C)
acosy	Automatic analysis of COSY data (C)
acosyold	Automatic analysis of COSY data, old algorithm (C)
acq_errors	Acquisition Done and Error Codes
acqdequeue	Dequeue an acquisition
acqdisp	Display message on the acquisition status line (C)
acqi	Interactive acquisition display process (C)
acqmeter	Open Acqmeter window (M)
Acqmeter	Open Acqmeter window (U)
acqmode	Acquisition mode (P)
acqreserve	Reserve the acquisition console for the current owner
acqstat	Open Acquisition Status window (M)
Acqstat	Open Acquisition Status window (U)
acqstatus	Acquisition status (P)
acquire	Acquire data (M)

actionid	Current study queue node id (P)
activestudy	Active study name (P)
add	Add current FID to add/subtract experiment (C)
addi	Start interactive add/subtract mode (C)
addnucleus	Add new nucleus to existing probe file (M)
addpar	Add selected parameters to current experiment (M)
addparams	Add parameter to current probe file (M)
addprobe	Create new probe directory and probe file (M)
adept	Automatic DEPT analysis and spectrum editing (C)
aexppl	Automatic plot of spectral expansion (M)
ai	Select absolute-intensity mode (C)
aig	Absolute-intensity group (P)
alfa	Set alfa delay before acquisition (P)
alock	Automatic lock control (P)
ampmode	Independent control of amplifier mode (P)
amptype	Amplifier type (P)
analyz	Calculate standard peak height (M)
analyze	Generalized curve fitting (C)
annotation	Display annotation specified by the parameter "template" or the default.
ap	Print out "all" parameters (C)
ap	"All" parameters display control (P)
apa	Plot parameters automatically (M)
aph	Automatic phase adjustment of spectra (C)
aph0	Automatic phase of zero-order term (C)
aphb	Auto phasing for Bruker data (C)
aphx	Perform optimized automatic phasing (M)
appdir	Application directory information
appdirs	Starts Applications Directory Editor (M)
appmode	Application mode (P)
apptype	Application type (P)
Apt	Set up parameters for APT experiment (M)
aptaph	Automatic processing for APT spectra (M)

array	Easy entry of linearly spaced array values (M)
array	Parameter order and precedence (P)
arraydim	Dimension of experiment (P)
array2csv	Formats Array into Comma Separate Variable
array2string	Formats Array into String
array2strsv	Formats Array into String Separated Variable
asin	Find arc sine of number (C)
asize	Make plot resolution along f_1 and f_2 the same (M)
assign	Assign transitions to experimental lines (M)
at	Acquisition time (P)
atan	Find arc tangent of a number (C)
atan2	Find arc tangent of two numbers (C)
atcmd	Call a macro at a specified time (M)
atext	Append string to current experiment text file (M)
attval	Calculate pulse width (M)
atune	ProTune Present (P)
au	Submit experiment to acquisition and process data (M)
AuCALch3i	Set up autocalibration with CH3I sample (M)
AuCALch3i1	Get autocalibration with CH_3I sample (M)
AuCALch3oh	Set up autocalibration with Autotest sample (M)
AuCALch3oh1	Get autocalibration with Autotest sample (M)
Aucalibz0	Automatic Hz to DAC calibration for Z0 (M)
AuCdec	Carbon decoupler calibration macro (M)
AuCgrad	Carbon/proton gradient ratio calibration macro (M)
AuCobs	Carbon observe calibration macro (M)
audiofilter	Audio filter board type (P)
Aufindz0	Automatic adjustment of Z0 (M)
Augcal	Probe gcal calibration macro (M)
Augmap	Automated gradient map generation (M)
Augmapz0	Automatic lock gradient map generation and z0 calibration (M)
AuHdec	Proton decoupler calibration (M)
AuHobs	Proton observe calibration macro (M)

Aumakegmap	Auto lock gradient map generation (M)
AuNuc	Get parameters for a given nucleus (M)
auto	Prepare for an automation run (C)
auto	Automation mode active (P)
auto_au	Controlling macro for automation (M)
autoaa	Abort an automation run with no error
Autobackup	Back up current probe file (M)
autodept	Automated complete analysis of DEPT data (M)
autodir	Automation directory absolute path (P)
autogo	Start automation run (C)
autolist	Set up and start chained acquisition (M)
automerge	Merges overniteQ with daytimeQ
Automkdir	Creates Data Directory from Template
autoname	Create path for data storage (C)
autoname	Prefix for automation data file (P)
autoq	Utility commands for the automation queue
autora	Resume suspended automation run (C)
autosa	Suspend current automation run (C)
autoscale	Resume autoscaling after limits set by scalelimits macro (M)
autostack	Automatic stacking for processing and plotting arrays (M)
autotest	Open Auto Test Window (C)
autotime	Displays approximate time for automation (M)
av	Set abs. value mode in directly detected dimension (C)
av1	Set abs. value mode in 1st indirectly detected dimension (C)
av2	Set abs. value mode in 2nd indirectly detected dimension (C)
averag	Calculate average and standard deviation of input (C)
awc	Additive weighting const. in directly detected dimension (P)
awc1	Additive weighting const. in 1st indirectly detected dimension (P)
awc2	Additive weighting const. in 2nd indirectly detected dimension (P)
axis	Provide axis labels and scaling factors (C)
axis	Axis label for displays and plots (P)
axisf	Axis label for FID displays and plots (P)

aa — Abort acquisition with error (C)

Syntax `aa`

Description Aborts an experiment that has been submitted to acquisition. If the experiment is active, it is aborted immediately, all data is discarded, and the experiment is interpreted as an error. Any data collected from an earlier block size transfer is retained. If any `werr` processing is defined, that processing occurs, followed by any queued experiments. The `login` name, and the FID directory path in `file` are used as keys to find the proper experiment to abort.

In some circumstances, there is a delay between the time `go` is entered and the acquisition is started. During this time, instructions based on the selected pulse sequence are being generated. This is signified by the letters "`PSG`" appearing in the upper left corner of the status window. An `aa` command issued under these circumstances reports that no acquisition is active but it instead stops the instruction generation process and the message "PSG aborted" appears.

See also *NMR Spectroscopy User Guide*

Related

`file`	File name of a parameter set (P)
`go`	Submit experiment to acquisition (C)
`halt`	Abort acquisition with no error (C)
`werr`	Specify action when error occurs (C)
`werr`	When error (P)

abort — Terminate action of calling macro and all higher macros (C)

Syntax `abort`

Description Terminates the action of the calling macro and all higher levels of nested macros. `abort` is used only in macros and not entered from the keyboard. It generates an error condition, which is the reason why the calling macro and any parent (nested) macros above will also be aborted. To exit from the execution of a macro without generating an error, use `return`.

See also *VnmrJ User Programming*

Related

`abortoff`	Terminate normal functioning of `abort` in a macro (C)
`aborton`	Restore normal functioning of `abort` in a macro (C)
`return`	Terminate execution of a macro (C)

abortallacqs — Reset acquisition computer in a drastic situation (C)

Syntax `abortallacqs`

Description Reboots the acquisition system from the host computer. Wait at least 30 seconds before attempting new acquisitions.

See also *NMR Spectroscopy User Guide*

abortoff Terminate normal functioning of abort in a macro (C)

Syntax `abortoff`

Description Changes the action of an `abort` command in a macro. Normally, `abort` (or any command aborting with an error condition) terminates the action of the calling macro and all higher levels of nested macros; however if the `abortoff` command is executed prior to a macro containing the `abort` command, only the macro containing `abort` terminates and execution continues to the next macro. The operation of the `abortoff` command is nullified by the `aborton` command. `abortoff` is used only in macros and not entered from the keyboard.

See also *VnmrJ User Programming*

Related `abort` Terminate action of calling macro and all higher macros (C)
`aborton` Restore normal functioning of `abort` in a macro (C)

aborton Restore normal functioning of abort in a macro (C)

Syntax `aborton`

Description Nullifies the operation of a `abortoff` command and restores the normal functioning of the `abort` command. `aborton` is used only in macros and not entered from the keyboard.

See also *VnmrJ User Programming*

Related `abortoff` Terminate normal functioning of `abort` in a macro (C)

abs Find absolute value of a number (C)

Syntax `abs(number)<:value>`

Description Finds the absolute value of a number. Absolute value is a nonnegative number equal in numerical value to the given number (e.g., `abs(-6.5)` is 6.5).

Arguments `number` is the given real number.

`value` is the return value with the absolute value of the given number. The default is to display the value in the status window.

Examples
```
abs(-25)
abs(n):abs_val
```

See also *VnmrJ User Programming*

AC1S-AC11S Autocalibration macros (M)

Syntax ACnS, where n is a number from 1 to 11.

Description Performs automatic system calibration. When finished with the calibration routines, the current probe file is updated. If the probe is new to the system (i.e., all values in the probe file are zero), system power levels are determined followed by calibration. If power levels are listed in the current probe file, these values are used. The macro AC1S determines ^{1}H pw90, AC5S begins ^{13}C calibration, including decoupler power calibrations. AC10S performs ^{19}F calibration, and AC11S performs ^{31}P calibration.

See also *NMR Spectroscopy User Guide*

ACbackup Make backup copy of current probe file (M)

Syntax ACbackup

Description Called by the autocalibration macros AC1S-AC11S to back up the probe file after calibration ends. This macro is not usually called by the user.

See also *NMR Spectroscopy User Guide*

Related AC1S-AC11S Autocalibration macros (M)

acct Writes records for operator login and logoff (M)

Applicability VnmrJ

Syntax acct('start'|'done')

Description acct writes operator login and logoff records to the system adm/tmp/macrorecords.txt file used by the accounting package.

See also *VnmrJ Installation and Administration* manual

Related

operator	operator name (P)
operatorlogin	Sets work space and parameters for the operator (M)
vnmr_accounting	Open Accounting window (U)

ACreport Print copy of probe file after autocalibration (M)

Syntax ACreport

Description Called by the autocalibration macros AC1S-AC11S to print a copy of the probe file before beginning a new autocalibration run.

See also *NMR Spectroscopy User Guide*

Related AC1S-AC11S Autocalibration macros (M)

acos Find arc cosine of number (C)

Syntax acos(value)<:n>

Description Finds the arc cosine (also called the inverse cosine) of a number.

Arguments value is a number in the range of ±–1.0 to +1.0.

n is a return argument giving the arc cosine, in radians, of value. The default is to display the arc cosine value in the status window.

Examples
```
acos(.5)
acos(value):acos_val
```

See also *VnmrJ User Programming*

Related sin Find sine value of an angle (C)

acosy Automatic analysis of COSY data (C)

Syntax acosy

Description Automatically analyzes a 2D COSY data set with fn=fn1 and sw=sw1. In this algorithm, a fuzzy pattern recognition technique is used to detect peaks and cluster the cross peaks into groups. Symmetry measures and chemical shifts for all cross peaks are calculated. Connectivities and the correlation table are displayed on the computer screen. This method is less sensitive to the threshold and rejects most artifacts in the peak list.

See also *NMR Spectroscopy User Guide*

Related

acosyold	Automatic analysis of COSY data (C)
fn	Fourier number in 1st indirectly detected dimension (P)
fn1	Fourier number in directly detected dimension (P)
ll2d	Automatic and interactive 2D peak picking (C)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 1st indirectly detected dimension (P)

acosyold Automatic analysis of COSY data, old algorithm (C)

Syntax acosyold

Description Analyzes COSY data using an old algorithm.

See also *NMR Spectroscopy User Guide*

Related

acosy	Automatic analysis of COSY data (C)
fn	Fourier number in 1st indirectly detected dimension (P)
fn1	Fourier number in directly detected dimension (P)
ll2d	Automatic and interactive 2D peak picking (C)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 1st indirectly detected dimension (P)

acq_errors Acquisition Done and Error Codes

Applicability VnmrJ 3.1

Description Whenever wbs, wnt, wexp, or werr processing occurs, the acquisition condition which initiated that processing is available from the parameter acqstatus. This acquisition condition is represented by two numbers, a "done" code and an "error" code. The done code is set in acqstatus[1] and the error code is set in acqstatus[2]. Macros may take different action depending on the acquisition condition. The done codes and error codes are listed below. As an example, a

Description werr macro could specify special processing if the maximum number of transients of accumulated. The appropriate test in the macro would be:

```
if (acqstatus[2] = 200) then
"do special processing,  e.g. dp='y' au"
endif
```

The acquisition error messages printed by Vnmr may be modified by creating an "acqerrmsgs" file with substitute messages. Each line in the file contains an error code followed by the text of the desired message. Error codes that do not occur in the acqerrmsgs file will continue to produce the standard messages. Vnmr first searches for the acqerrmsgs file in the user's "$vnmruser/templates" directory; if the file is not there Vnmr looks the the system directory "$vnmrsystem/user_templates". Entries are taken only from one file or the other, their contents are not "merged". A typical entry in the file would be:

```
301Can't spin the spinner
```

Error codes marked with an asterisk (*) are not used on Mercury and GEMINI 2000

Done Codes:

11. FID Complete.
12. Block Size Complete. (error code indicates BS # completed)
13. Soft Error.
14. Warning.
15. Hard Error.
16. Experiment Aborted.

17. Setup Completed. (error code indicates type of setup completed)

101. Experiment Complete.

102. Experiment Started.

Error codes:

Note: WARNINGS - Experiment acquisition continues.

SOFTERRORS - Experiment acquisition is stopped.

WARNINGS:

101. Low Noise Signal.

102. High Noise Signal.

103. ADC overflow occurred.

104. Receiver overflow occurred.*

SOFTERROR:

200. Maximum Transient Completed for Single Precision Data.

WARNINGS or SOFTERRORS: (User selectable)

201. Lost Lock during experiment. (`LOCKLOST`)

Spinner Errors:

301. Sample failing to spin after three attempts of repositioning. (`BUMPFAIL`)

302. Spinner did not regulate in the allowed time period. (`RSPINFAIL`)*

303. Spinner went out of regulation during the experiment. (`SPINOUT`)*

395. Unknown Spinner device specified. (`SPINUNKNOWN`)*

396. Spinner device is not powered up. (`SPINNOPOWER`)*

397. RS232 Cable not connected between console and Spinner device.

(`SPINRS232`)*

398. Spinner does not acknowledge commands. (`SPINTIMEOUT`)*

VT Errors:

400. VT did not regulate in the given time 'vttime' after being set.

401. VT went out of regulation during the experiment. (`VTOUT`)

402. VT is in manual mode after the automatic command given. (see oxford manual)*

403. VT Safety Sensor has reached limit. (see oxford manual)*

404. VT can not turn on cooling gas. (see oxford manual)*

405. VT main sensor on bottom limit. (see oxford manual)*

406. VT main sensor on top limit. (see oxford manual)*

407. VT sc/ss error. (see oxford manual)*

408. VT oc/ss error. (see oxford manual)*

495. Unknown VT device specified. (`VTUNKNOWN`)*

496. VT device is not powered up. (`VTNOPOWER`)*

497. RS232 Cable not connected between console and VT device. (`VTRS232`)*

498. VT does not acknowledge commands. (`VTTIMEOUT`)

SOFTERROR:

Sample Changer Errors:

501. Sample changer has no sample to retrieve.

502. Sample changer arm unable to move up during retrieve.

503. Sample changer arm unable to move down during retrieve.

504. Sample changer arm unable to move sideways during retrieve.

505. Invalid sample number during retrieve.

506. Invalid temperature during retrieve.

507. Gripper abort during retrieve.

508. Sample out of range during automatic retrieve.

509. Illegal command character during retrieve.*

510. Robot arm failed to find home position during retrieve.*

511. Sample tray size is not consistent.*

512. Sample changer power failure during retrieve.*

513. Illegal sample changer command during retrieve.*

514. Gripper failed to open during retrieve.*

515. Air supply to sample changer failed during retrieve.*

525. Tried to insert invalid sample number.*

526. Invalid temperature during sample changer insert.*

527. Gripper abort during insert.*

528. Sample out of range during automatic insert.

529. Illegal command character during insert.*

530. Robot arm failed to find home position during insert.*

531. Sample tray size is not consistent.*

532. Sample changer power failure during insert.*

533. Illegal sample changer command during insert.*

534. Gripper failed to open during insert.*

535. Air supply to sample changer failed during insert.*

593. Failed to remove sample from the magnet.*

594. Sample failed to spin after automatic insert.

595. Sample failed to insert properly.

596. Sample changer not turned on.

597. Sample changer not connected to RS-232 interface.

598. Sample changer not responding.*

Shimming Errors:

601. Shimming User Aborted.*

602. Lost Lock while Shimming.*

604. Lock Saturation while Shimming.*

608. A Shim Coil DAC limit hit while Shimming.*

Auto-Lock Errors:

701. User Aborted.(ALKABORT)*

702. Auto Lock Failure in finding resonance of sample. (ALKRESFAIL)

703. Auto Lock Failure in lock power adjustment. (ALKPOWERFAIL)*

704. Auto Lock Failure in lock phase adjustment. (ALKPHASFAIL)*

705. Auto Lock Failure, lock lost in finial gain adjustment. (ALKGAINFAIL)*

Auto-Gain Errors:

801. Auto-Gain failure, gain driven to zero, reduce pulse width (pw).

(AGAINFAIL)

HARDERRORS:

901. Incorrect PSG version for Acquisition.

902. Sum-to-Memory Error, Number of points acquired not equal to np.

903. Fifo Underflow Error. (A delay too small?). *

904. Requested number of data points (np) to acquire is too large for acquisition.*

905. Acquisition Bus Trap (Experiment maybe lost). *

SCSI Errors

1001. Recoverable SCSI read transfer from Console Occurred. *

1002. Recoverable SCSI write transfer from Console Occurred. *

1003. Unrecoverable SCSI read transfer Error. *

1004. Unrecoverable SCSI write transfer Error. *

Host disk errors

1101. Error opening disk file. (most likely a Unix premission problem.)*

1102. Error on closing disk file.*

1103. Error on reading from disk file.*

1104. Error on writing to disk file.*

RF Monitor errors (only on Inova systems with RF monitor)

1400. An RF monitor trip occurred but the error status is OK

1401. Reserved RF monitor trip A occurred

1402. Reserved RF monitor trip B occurred

1404. Excessive reflected power at quad hybrid

1405. STOP button pressed at operator station

1406. Power for RF Monitor board (RFM) failed

1407. Attenuator control or readback failed

1408. Quad reflected power monitor bypassed (Warning)

1409. Power supply monitor for RF Monitor board (RFM) bypassed (Warning)

1410. Ran out of memory to report RF monitor errors

1411. No communication with RF monitor system

1421. Reserved RF monitor trip A1 occurred on observe channel

1422. Reserved RF monitor trip B1 occurred on observe channel

1423. Reserved RF monitor trip C1 occurred on observe channel

1424. RF Monitor board (PALI/TUSUPI) missing on observe channel

1425. Excessive reflected power on observe channel

1426. RF amplifier gating disconnected on observe channel

1427. Excessive power detected by PALI on observe channel

1428. RF Monitor system (TUSUPI) heartbeat stopped on observe channel

1429. Power supply for PALI/TUSUPI failed on observe channel

1430. PALI asserted REQ_ERROR on observe channel (should never occur)

1431. Excessive power detected by TUSUPI on observe channel

1432. RF power amp: overdrive on observe channel

1433. RF power amp: excessive pulse width on observe channel

1434. RF power amp: maximum duty cycle exceeded on observe channel

1435. RF power amp: overheated on observe channel

1436. RF power amp: power supply failed on observe channel

1437. RF power monitoring disabled on observe channel (Warning)

1438. Reflected power monitoring disabled on observe channel (Warning)

1439. RF power amp monitoring disabled on observe channel (Warning)

1461. Reserved RF monitor trip A2 occurred on decouple channel

1462. Reserved RF monitor trip B2 occurred on decouple channel

1463. Reserved RF monitor trip C2 occurred on decouple channel

1464. RF Monitor board (PALI/TUSUPI) missing on decouple channel

1465. Excessive reflected power on decouple channel

1466. RF amplifier gating disconnected on decouple channel

1467. Excessive power detected by PALI on decouple channel

1468. RF Monitor system (TUSUPI) heartbeat stopped on decouple channel

1469. Power supply for PALI/TUSUPI failed on decouple channel

1470. PALI asserted REQ_ERROR on decouple channel (should never occur)

1471. Excessive power detected by TUSUPI on decouple channel

1472. RF power amp: overdrive on decouple channel

1473. RF power amp: excessive pulse width on decouple channel

1474. RF power amp: maximum duty cycle exceeded on decouple channel

1475. RF power amp: overheated on decouple channel

1476. RF power amp: power supply failed on decouple channel

1477. RF power monitoring disabled on decouple channel (Warning)

1478. Reflected power monitoring disabled on decouple channel (Warning)

1479. RF power amp monitoring disabled on decouple channel (Warning)

acqdequeue Dequeue an acquisition

Syntax `acqdequeue<:$ret>` - dequeue acquisition from current experiment

`acqdequeue<('go_id')><:$ret>` - dequeue an acquisition

Applicability VnmrJ 3.1

Description When a `go`, `ga`, or `au` command is issued, instructions are sent to the acquisition system to run that experiment. If another experiment is already running, the request is queued. When the prior experiment finishes, the queued acquisition will start. The `acqdequeue` command will remove an experiment from this queue. The `acqdequeue` command will not stop an experiment that is already started. An optional return argument will be set to 1 if the experiment is successfully dequeued; otherwise it will be set to 0.

Arguments When a `go`, `ga`, or `au` command is issued, a unique identifier is added to the parameter set, in the processed tree. This parameter is named '`go_id`'. This parameter can be used as an argument for the `acqdequeue` command. If no argument is given, the value of this parameter in the current experiment's processed tree is used.

acqdisp Display message on the acquisition status line (C)

Syntax `acqdisp(message)`

Description Displays the message specified on the acquisition status line. `acqdisp` is used primarily by the acquisition process to update the screen.

Arguments `message` is a text string, up to 8 characters long.

See also *NMR Spectroscopy User Guide*

acqi Interactive acquisition display process (C)

Syntax `acqi<('par'|'disconnect'|'exit'|'standby')><:$ret>`

Description Opens the Acquisition window for interactive locking and shimming on the lock signal, FID, or spectrum. When using a spectrometer, `acqi` normally automatically starts. On all systems, if the console has been recently rebooted, enter `su` before running `acqi`.

If `acqi` is connected to the console and you start an acquisition (`su`/`go`/`au`), `acqi` automatically disconnects.

The pulse sequence and parameter set for the FID/spectrum display can be selected by entering `gf`. Note that if clicking the FID button in `acqi` causes `acqi` to "disconnect," the common cause is that `gf` had not been executed.

The FID display is controlled by the parameters `lsfid`, `phfid`, and `dmgf`. These display parameters are automatically sent to `acqi` when `acqi` is first invoked. These parameters may subsequently be changed and sent again to `acqi` with the command `acqi('par')`. If `phfid` is not set to "Not Used" for the FID display in `acqi`, a slide control will be available in `acqi` for the interactive adjustment of the `phfid` parameter. The slide will be in the IPA set of adjustments. If the parameter `dmgf` exists and is set to `'av'`, the FID display in `acqi` displays the square root of the sum of the squares of the real and imaginary channels.

The spectrum display is controlled by parameters `sp`, `wp`, `dmg`, `rp`, `lp`, `rfl`, `rfp`, `vs`, `vp`, `sw`, and `fn`. These parameters are automatically sent to `acqi` when `acqi` is first invoked. These parameters can subsequently be changed and sent again to `acqi` with the command `acqi('par')`. The preparation macro `gf` also calls `acqi('par')`, thereby causing these parameters to be sent to `acqi`. If `fn` is greater than 64K, it is lowered to 64K.

A convenient method of setting these parameters is to acquire a spectrum with `go`, then `ft` and adjust the display with the `ds` command options. Once the display is set the way you want, enter `gf`. The same display should then appear when the spectrum display is selected from `acqi`. Note that weighting parameters are not used in the *acqi* spectrum display.

The manual *NMR Spectroscopy User Guide* has a step-by-step description of using `acqi`.

Arguments
'par' causes the current values of parameters lsfid, phfid, dmgf, sp, wp, dmg, rp, lp, rfl, rfp, vs, sw, and fn to be sent to acqi.

'disconnect' causes acqi to be disconnected. Clicking the Close button in acqi is equivalent, and puts acqi in the standby mode. Lock parameters, the spin parameter, and the shim values are sent back to the current experiment when acqi is “disconnected.” If the experiment has the load parameter set to 'y', then the shim values are not delivered to the experiment.

'exit' causes an exit from acqi. Clicking the exit button in the Acquisition window is equivalent.

$ret is a return value with the success or failure of running acqi. The default is a warning displayed in the status window if acqi fails.

'standby' starts acqi and puts it into the standby mode.

Examples
```
acqi
acqi('par')
acqi('disconnect')
acqi('exit')
acqi:$ok
```

See also *NMR Spectroscopy User Guide*

Related

Acqstat	Bring up the acquisition status display (U)
dmg	Display mode in directly detected dimension (P)
dmgf	Absolute-value display of FID data or spectrum in acqi (P)
ds	Display a spectrum (C)
fn	Fourier number in directly detected dimension (P)
ft	Fourier transform 1D data (C)
gf	Prepare parameters for FID/spectrum display in acqi (M)
go	Submit an experiment to acquisition (C)
load	Load status of displayed shims (P)
lkof	Track changes in lock frequency (P)
lp	First-order phase in directly detected dimension (P)
lsfid	Number of complex points to left-shift the np FID (P)
phfid	Zero-order phasing constant for np FID (P)
rfl	Ref. peak position in 1st indirectly detected dimension (P)
rfp	Ref. peak frequency in directly detected dimension (P)
rp	Zero-order phase in directly detected dimension (P)
sp	Start of plot in directly detected dimension (P)
spin	Sample spin rate (P)
sw	Spectral width in directly detected dimension (P)
vp	Vertical position of the spectrum (P)
vs	Vertical scale (P)
wp	Width of plot in directly detected dimension (P)

acqmeter Open Acqmeter window (M)

Syntax `acqmeter<(remote_system)>`

Description Opens the Acqmeter window and shows a time line of lock level, temperature (VT), and/or spinner speed. When first opened, only lock level is displayed. By clicking anywhere in the lock level window with the right mouse button, a menu pops up with choices to close the lock level window, show a temperature (VT) window, show a spinner window, open a properties window, or close the Acqmeter window. Click on the choice desired in the menu with either the left or right mouse button. In the properties window, the host, font, color, and graphical mode can be changed. Continue to click in any Acqmeter window with the right mouse button to open the menu and then open or close windows, or close the Acqmeter window, as desired.

Arguments `remote_system` is the host name of a remote machine on the same network. The default is the local machine. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the `/etc/hosts` file).

Examples `acqmeter`
`acqmeter('nmr500')`

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display (C)
Acqmeter	Open Acqmeter window (U)

Acqmeter Open Acqmeter window (U)

Syntax `Acqmeter <remote_system> <-f file> <&>`

Description Opens the Acqmeter window and shows a time line of lock level, temperature (VT), and/or spinner speed. When first opened, only lock level is displayed. By clicking anywhere in the lock level window with the right mouse button, a menu pops up with choices to close the lock level window, show a temperature (VT) window, show a spinner window, open a properties window, or close the Acqmeter window. Click on the choice desired in the menu with either the left or right mouse button. In the properties window, the host, font, color, and graphical mode can be changed. Continue to click in any Acqmeter window with the right mouse button to open the menu and then open or close windows, or close the Acqmeter window, as desired.

Arguments `remote_system` is the host name of a remote machine on the same network. The default is the local machine. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the `/etc/hosts` file).

`-f file` is the name of a template file in the directory `$vnmruser/vnmrsys/templates/acqstat` used to set the attributes of the Acqmeter window when it opens. This allows customizing the Acqmeter window for different users and experiments. The default name of the file is `default`.

& (ampersand) character added to the command makes `Acqmeter` into a background process. For example, if "lab" is the remote machine host name, entering the command `Acqmeter lab &` displays the acquisition status of the "lab" remote machine as a background process. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the `/etc/hosts` file).

Examples
```
Acqmeter &
Acqmeter nmr400 &
Acqmeter gem300 -f inova500.lisa &
```

See also *NMR Spectroscopy User Guide*

Related

`acqi`	Interactive acquisition display (C)
`acqmeter`	Open Acqmeter window (M)

acqmode — Acquisition mode (P)

Description A global parameter specifying the normal acquisition mode for acquiring, locking, fid shimming, and prescan in VnmrJ.

Values `''` (empty string) normal acquisition

`'lock'` lock acquisition

`'fidscan'` fid shimming acquisition

`'prescan'` prescan acquisition

See also *VnmrJ Imaging, User Guide, NMR Spectroscopy User Guide*

acqreserve — Reserve the acquisition console for the current owner

Syntax `acqreserve`

Applicability VnmrJ 3.1

Description `acqreserve` controls reservation of the NMR acquisition console, allowing a user sole access. It reserves the console for the current user, as specified by the owner parameter. This user / owner will have access to the acquisition commands to acquire data, lock, shim, set temperature, etc. If the console is reserved, any VnmrJ session with a different user / owner will be forced into a datastation mode. Access to acquisition related commands and acquisition related panels will be prevented in this datastation mode.

By default, a console reservation will be removed when the user / owner exits from the current VnmrJ session. Using the acqreserve('on','noAutoOff') option causes the reservation to remain intact, even after the user / owner exits. For example, they will maintain the console reservation while they travel and start a new session at a remote site. If they do not exit from the first VnmrJ session and start a second session somewhere else, that second session will share the reservation. The reservation does not need to be turned off from the VnmrJ that started it. The capitalization in the second argument is ignored. The argument 'noautooff' also works. The invocation acqreserve('autooff') is used when the user exits. This will turn off the reservation, as long as it was not turned on with the 'noAutoOff' option.

A force option will turn the reservation on or off, even if the current user / owner is not the one that made the original reservation. A record will be kept of forced reservation events.

If acqreserve is never used, or after acqreserve('off') is issued, access to the console will be available on a first come first served basis. As soon as the console becomes "Idle", any user not in datastation mode will be able to access it.

Arguments acqreserve takes up to three optional arguments. The first argument is 'on', 'off', or 'autooff'. If no argument is given, the default is 'on'. acqreserve('on') makes the reservation. acqreserve('off') removes the reservation. acqreserve('autooff') is described below. The other optional arguments are 'noAutoOff' and 'force'. They can be provided in either order, following the 'on', 'off', or 'autooff' argument.

acqstat Open Acquisition Status window (M)

Syntax acqstat<(remote_system)>

Description Opens the Acquisition Status window, which displays acquisition information such as the current acquisition task, experiment number, spinner status, and temperature status. When the host computer is attached to a spectrometer, this window should open automatically when VnmrJ is started. In the properties window, the host, font, color, and graphical mode can be changed. For a complete description of these windows, refer to the manual *NMR Spectroscopy User Guide*.

Arguments remote_system is the host name of a remote machine on the same network. The default is the local machine. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the /etc/hosts file).

Examples acqstat
acqstat('u500')

See also *NMR Spectroscopy User Guide*

Related
`Acqstat` Open the Acquisition Status window (U)
`showstat` Display information about status of acquisition (C,U)

Acqstat Open Acquisition Status window (U)

Syntax `Acqstat <remote_system> <-f file> <&>`

Description Opens the Acquisition Status window, which displays acquisition information such as the current acquisition task, experiment number, spinner status, and temperature status. When the host computer is attached to a spectrometer, this window should open automatically when VnmrJ is started. In the properties window, the host, font, color, and graphical mode can be changed. For a complete description of these windows, refer to the manual *NMR Spectroscopy User Guide*.

Arguments `remote_system` is the host name of a remote machine on the same network. The default is the local machine. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the `/etc/hosts` file).

`-f file` is the name of a template file in the directory `$vnmruser/vnmrsys/templates/acqstat` used to set the attributes of the Acquisition Status window when it opens. This allows customizing the Acquisition Status window for different users and experiments. The default name of the file is `default`.

& (ampersand) character added to the command makes `Acqstat` into a background process. For example, if “lab” is the remote machine host name, entering the command `Acqstat lab &` displays the acquisition status of the “lab” remote machine as a background process. To activate the remote feature, the local and remote machines must be on the same Ethernet LAN (local area network) and the local machine must be able to get the Internet address of the remote machine (usually in the `/etc/hosts` file).

Examples
```
Acqstat &
Acqstat nmr400 &
Acqstat gem300 -f inova500.lisa &
```

See also *NMR Spectroscopy User Guide*

Related
`Acqstat` Open the Acquisition Status window (U)
`showstat` Display information about status of acquisition (C,U)

acqstatus Acquisition status (P)

Description Whenever wbs, wnt, wexp, or werr processing occurs, the acquisition condition that initiated that processing is available from the parameter acqstatus. This acquisition condition is represented by two numbers, a “done” code and an “error” code. The done code is set in acqstatus[1] and the error code is set in acqstatus[2]. Macros can take different actions depending on the acquisition condition.

The done codes and error codes are listed below and in the file acq_errors in /vnmr/manual. For example, a werr macro could specify special processing if the maximum number of transients is accumulated. The appropriate test in the macro would be:

```
if (acqstatus[2] = 200) then
“do special processing, e.g. dp='y' au”
endif
```

Done codes:

11. FID complete
12. Block size complete (error code indicates bs number completed)
13. Soft error
14. Warning
15. Hard error
16. Experiment aborted
17. Setup completed (error code indicates type of setup completed)
101. Experiment complete
102. Experiment started

Error codes:

Warnings
101. Low-noise signal
102. High-noise signal
103. ADC overflow occurred
104. Receiver overflow occurred*

Soft errors
200. Maximum transient completed for single-precision data
201. Lost lock during experiment (LOCKLOST)

300. *Spinner errors:*
301. Sample fails to spin after three attempts at repositioning
302. Spinner did not regulate in the allowed time period (RSPINFAIL)*
303. Spinner went out of regulation during the experiment (SPINOUT)*
395. Unknown spinner device specified (SPINUNKNOWN)*
396. Spinner device is not powered up (SPINNOPOWER)*
397. RS-232 cable not connected from console to spinner (SPINRS232)*
398. Spinner does not acknowledge commands (SPINTIMEOUT)*

400. *VT (variable temperature) errors:*
400. VT did not regulate in the given time vttime after being set
401. VT went out of regulation during the experiment (VTOUT)
402. VT in manual mode after automatic command (see Oxford manual)*
403. VT safety sensor has reached limit (see Oxford manual)*

404. VT cannot turn on cooling gas (see Oxford manual)*
405. VT main sensor on bottom limit (see Oxford manual)*
406. VT main sensor on top limit (see Oxford manual)*
407. VT sc/ss error (see Oxford manual)*
408. VT oc/ss error (see Oxford manual)*
495. Unknown VT device specified (VTUNKNOWN)*
496. VT device not powered up (VTNOPOWER)*
497. RS-232 cable not connected between console and VT (VTRS232)*
498. VT does not acknowledge commands (VTTIMEOUT)

500. *Sample changer errors:*
501. Sample changer has no sample to retrieve
502. Sample changer arm unable to move up during retrieve
503. Sample changer arm unable to move down during retrieve
504. Sample changer arm unable to move sideways during retrieve
505. Invalid sample number during retrieve
506. Invalid temperature during retrieve
507. Gripper abort during retrieve
508. Sample out of range during automatic retrieve
509. Illegal command character during retrieve*
510. Robot arm failed to find home position during retrieve*
511. Sample tray size is not consistent*
512. Sample changer power failure during retrieve*
513. Illegal sample changer command during retrieve*
514. Gripper failed to open during retrieve*
515. Air supply to sample changer failed during retrieve*
525. Tried to insert invalid sample number*
526. Invalid temperature during sample changer insert*
527. Gripper abort during insert*
528. Sample out of range during automatic insert
529. Illegal command character during insert*
530. Robot arm failed to find home position during insert*
531. Sample tray size is not consistent*
532. Sample changer power failure during insert*
533. Illegal sample changer command during insert*
534. Gripper failed to open during insert*
535. Air supply to sample changer failed during insert*
593. Failed to remove sample from magnet*
594. Sample failed to spin after automatic insert
595. Sample failed to insert properly
596. Sample changer not turned on
597. Sample changer not connected to RS-232 interface
598. Sample changer not responding*

600. *Shimming errors:*
601. Shimming user aborted*
602. Lost lock while shimming*
604. Lock saturation while shimming*
608. A shim coil DAC limit hit while shimming*

700. *Autolock errors:*
701. User aborted (ALKABORT)*
702. Autolock failure in finding resonance of sample (ALKRESFAIL)
703. Autolock failure in lock power adjustment (ALKPOWERFAIL)*

704. Autolock failure in lock phase adjustment (ALKPHASFAIL)*
705. Autolock failure, lock lost in final gain adjustment (ALKGAINFAIL)*

800. *Autogain errors.*
801. Autogain failure, gain driven to 0, reduce pw (AGAINFAIL)

Hard errors
901. Incorrect PSG version for acquisition
902. Sum-to-memory error, number of points acquired not equal to np
903. FIFO underflow error (a delay too small?)*
904. Requested number of data points (np) too large for acquisition*
905. Acquisition bus trap (experiment may be lost)*

1000. *SCSI errors:*
1001. Recoverable SCSI read transfer from console*
1002. Recoverable SCSI write transfer from console**
1003. Unrecoverable SCSI read transfer error*
1004. Unrecoverable SCSI write transfer error*

1100. *Host disk errors:*
1101. Error opening disk file (most likely a UNIX permission problem)*
1102. Error on closing disk file*
1103. Error on reading from disk file*
1104. Error on writing to disk file*

See also *NMR Spectroscopy User Guide*

Related	`react`	Recover from error conditions during `werr` processing (M)
	`werr`	Specify action when error occurs (C)
	`werr`	When error (P)

`acquire` Acquire data (M)

Description Macro to acquire data. It uses `execpars` to select the prep and prescan method, executes them, and then begins acquisition.

See also *NMR Spectroscopy User Guide*

Related	`execpars`	Set up the exec parameters (M)
	`execprescan`	Execute prescan macro (P)
	`xmnext`	Find next prescan or next experiment in study queue (M)
	`xmwexp`	Processing macro for end of acquisition in study queue (M)

`actionid` Current study queue node id (P)

Applicability Liquids, Imaging

Description Specifies the currently selected study queue node id.

See also VnmrJ Imaging, User Guide, NMR Spectroscopy User Guide

Related

xmaction	Perform study queue action (M)
xmnext	Find next prescan or next experiment in study queue (M)
xmselect	Action when study queue node is selected (M)

activestudy Active study name (P)

Applicability Liquids, Imaging

Description A global parameter that specifies the currently active study name. In the Walkup interface, it specifies the currently active automation run.

Values `'s_20050601'` active study name

`'auto_2005.06.01'` active automation run name

`'null'` no active study or automation run

See also VnmrJ Imaging, User Guide and NMR Spectroscopy User Guide

Related

acquire	Acquire data (M)
autodir	Automation directory absolute pathname (P)
cqinit	Initialize liquids study queue (M)
studyid	Study identification (P)
xmaction	Perform study queue action (M)
xmselect	Action when study queue node is selected (M)

add Add current FID to add/subtract experiment (C)

Syntax (1) `add<(multiplier<,'new'>)>`
(2) `add('new')`
(3) `add('trace',index)`

Description Adds the last displayed or selected FID to the current contents of the add/subtract experiment (`exp5`). The parameters lsfid and phfid can be used to shift or phase rotate the selected FID before it is combined with the data in the add/subtract experiment. A multi-FID add/subtract experiment can be created by using the `'new'` keyword. Individual FIDs in a multi-FID add/subtract experiment can subsequently be added to using the `'trace'` keyword followed by the index number of the FID.

Arguments `multiplier` is a value that the FID is to be multiplied by before being added to the add/subtract experiment (`exp5`). The default is 1.0.

`'new'` is a keyword to create a new FID element in a add/subtract experiment.

`'trace'` is a keyword to use the next argument (`index`) as the number of the FID to add to in an add/subtract experiment. The default is to add to the first FID in a multi-FID add/subtract experiment.

index is the index number of the FID to be used as a target in a multi-FID add/subtract experiment.

Examples
```
add
add(0.75)
add('new')
add('trace',2)
```

See also *NMR Spectroscopy User Guide*

Related

clradd	Clear add/subtract experiment (C)
lsfid	Number of complex points to left-shift ni interferogram (P)
phfid	Zero-order phasing constant for np FID (P)
select	Select a spectrum without displaying it (C)
spadd	Add current spectrum to add/subtract experiment (C)
sub	Subtract current FID from add/subtract experiment (C)

addi — Start interactive add/subtract mode (C)

Syntax addi

Description Starts the interactive add/subtract mode. Before entering addi, start the process with clradd and spadd, then display a second spectrum on the screen. This may involve changing experiments, selecting a second member of an array of spectra, a different trace of a 2D spectrum, or displaying a spin simulated spectrum. The Fourier numbers (fn) *must* be the same in the two spectra to be manipulated. The width (sw) of the two spectra need *not* be identical, although adding spectra of different widths will probably not be meaningful. Having selected the second spectrum and ensuring it is in nm mode, enter addi to begin the interactive process.

After addi is invoked, spectrum 1, the spectrum selected by the spadd command, appears in the center of the display. Spectrum 2, the spectrum that was active when addi was entered, appears on the bottom. The sum or difference of these spectra appears on top of the screen. When addi is first entered, this spectrum will be the sum (1 + 2) by default. The spectra is manipulated using the mouse.

The select button toggles between different modes of control.

- When the label at the screen bottom reads "active: current", all of the parameters (except wp) control spectrum 2, and spectrum 2 can be phased, scaled, or shifted relative to spectrum 1.
- After clicking on select, the label at the screen bottom reads "active: addsub", and now all of the parameters except wp control spectrum 1.
- Clicking select again toggles the label to read "active: result", and now parameter changes affect only the sum or difference spectrum.

Note that wp always controls all spectra, because differential expansions of the two spectra are not supported. Note also that the colors of the labels change to match the colors of the different spectra.

The sum/difference spectrum displayed on the screen while addi is active is strictly a temporary display. Once all manipulations have been performed, and assuming the sum/difference is something you wish to perform further operations with (such as plotting), it must be saved into the add/subtract experiment (exp5) by clicking on save. At this point, spectrum 1, which was in the add/subtract experiment, is overwritten by the sum or difference spectrum, and addi ceases operation. In most cases, you will next want to enter jexp5 ds to display the difference spectrum on the screen, ready for further manipulation (expansion, line listing, etc.) and plotting. If you wish to continue with the add/subtract process by adding in a third spectrum, display that spectrum in the usual way and enter addi again.

See also *NMR Spectroscopy User Guide*

Related		
	clradd	Clear add/subtract experiment (C)
	jexp	Join existing experiment (C)
	nm	Select normalized intensity mode (C)
	spadd	Add current spectrum to add/subtract experiment (C)
	spmin	Take minimum of two spectra in add/subtract experiment (C)
	spsub	Subtract current spectrum from add/subtract experiment (C)
	wp	Width of plot in directly detected dimension (P)

addnucleus Add new nucleus to existing probe file (M)

Applicability ALL

Description Entries for nuclei not in the default probe file are appended to the end of the file. The argument should correspond to a nucleus in the nuctable.

Syntax `addnucleus('nucleus')`

Arguments `nucleus` – name followed by atomic number, e.g. C13 not 13C.

Examples `addnucleus('Si29')`

See also *NMR Spectroscopy User Guide*

Related		
	addprobe	Create new probe directory and probe file (M)
	deletenucleus	Removes nucleus entry to probe file (M)
	getparam	Receive parameter from probe file (M)
	probe	Probe type (P)
	setparams	Write parameter to current probe file (M)

addpar — Add selected parameters to current experiment (M)

Syntax addpar<('2d'|'3d'|'3rf'|'4d'|'downsamp'|'fid'|
'image'|'ll2d'|'lp'<,dim>|'oversamp'|'ss')>

Applicability The '3d', '3rf', '4d', 'fid', and 'image' arguments work on all systems but are only useful if system has the proper hardware.

Description Creates selected parameters in the current experiment.

Arguments If no argument is entered, addpar displays instructions for its use.

'2d', '3d', '3rf', '4d', 'downsamp', 'fid', 'image', 'll2d', 'lp', 'oversamp', and 'ss' are keywords (only one keyword is used at a time) specifying the parameters to be created:

- '2d' specifies creating ni, phase, and sw1, which can be used to acquire a 2D data set (functions the same as macro par2d).
- '3d' specifies creating d3, ni2, phase2, and sw2, which can used to acquire a 3D data set (functions the same as macro par3d).
- '3rf' specifies retrieving the ap and dg2 display templates for third rf channel and 3D parameters (functions the same as macro par3rf).
- '4d' specifies creating the acquisition parameters d4, ni3, phase3, and sw3, which can be used to acquire a 4D data set (functions the same as macro par4d).
- 'downsamp' specifies creating the parameters downsamp, dscoef, dslsfrq, dsfb, and filtfile for digital filtering and downsampling (functions the same as macro pards).
- 'fid' specifies creating FID display parameters axisf, crf, deltaf, dotflag, vpf, and vpfi if the parameter set is older and lacks these parameters (functions the same as macro fidpar).
- 'll2d' specifies creating th2d and xdiag for the ll2d 2D peak picking program (functions the same as macro parll2d).
- 'lp' specifies creating lpalg, lpopt, lpfilt, lpnupts, strtlp, lpext, strtext, lptrace, and lpprint for linear prediction in the acquisition dimension (functions the same as macro parlp). The display template for the dglp macro is also created if necessary.
- 'oversamp' specifies creating parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp for oversampling and digital filtering (functions the same as macro paros).
- 'ss' specifies adding parameters ssorder, ssfilter, ssntaps, and sslsfrq for time-domain solvent subtraction (functions the same as macro parfidss).

dim specifies the dimension when adding linear prediction parameters: 1 for the first implicit dimension or 2 for the second implicit dimension. Default is the acquisition dimension. Therefore, addpar('lp') creates the parameters listed above; addpar('lp',1) creates lpalg1, lpopt1, lpfilt1, lpnupts1, strtlp1, lpext1, strtext1, lptrace1, and lpprint1; and addpar('lp',2) creates lpalg2, lpopt2, lpfilt2, lpnupts2, strtlp2, lpext2, strtext2, lptrace2, and lpprint2. Each separate dimension of a multidimensional data set can have its own unique parameters.

Examples
```
addpar
addpar('3d')
addpar('lp',1)
```

See also *NMR Spectroscopy User Guide; VnmrJ Imaging NMR*

Related

def_osfilt	Default value of osfilt (P)
fidpar	Add parameters for FID display in current experiment (M)
osfilt	Oversampling filter for real-time DSP (P)
par2d	Create 2D acquisition parameters (M)
par3d	Create 3D acquisition parameters (M)
par3rf	Get display templates for 3rd rf channel parameters (M)
par4d	Create 4D acquisition parameters (M)
pards	Create digital filtering and downsampling parameters (M)
parfidss	Set up parameters for time-domain solvent subtraction (M)
paros	Create oversampling and digital filtering parameters (M)
parll2d	Create parameters for 2D peak picking (M)
parlp	Create parameters for linear prediction (M)

addparams Add parameter to current probe file (M)

Syntax `addparams(param,value,nucleus<,'tmplt'><,'system'>)`

Description Adds a new parameter and its value for a specified nucleus to the probe file or to the probe template.

Arguments `param` is the name of the parameter to be added.

`value` is a string with the value to be written for the parameter.

`nucleus` is the nucleus to add in the probe file.

`'tmplt'` is a keyword to add the parameter to the local template. The default is the probe file.

`'system'` is a keyword to add the parameter to the system-level template or probe file, provided that you have write permission to that file. The default is to add the parameter to the local template or probe file.

Examples
```
addparams('ref_pwr','53',tn)
addparams('ref_pwx','00',dn,'tmplt')
addparams('ref_pwx2','00',dn2,'tmplt','system')
```

See also *NMR Spectroscopy User Guide*

Related

getparam	Receive parameter from probe file (M)
setparams	Write parameter to current probe file (M)
updateprobe	Update probe file (M)

addprobe Create new probe directory and probe file (M)

Syntax `addprobe(probe_name<,'stdar'|'system'><,'stdpar'>)`

Description Creates a new probe directory and a probe file. Default nuclei included in this file are ^{1}H, ^{19}F, ^{13}C, and ^{15}N. The information is saved in the user's directory `vnmrsys/probes`.

Arguments `probe_name` is the name to be given to the probe directory and probe file.

`'stdpar'` and `'system'` are keywords for the second and third arguments:

- If the second argument is `'stdpar'`, calibration values from the standard parameter sets (`stdpar/H1.par`, `stdpar/C13.par`, etc.) will be read and written into the probe file.
- If the second argument is `'system'` and the user has write permission into the VnmrJ system probes directory (typically `/vnmr/probes`), then a system-level probe directory will be made.
- If the second argument is `'system'` and the third argument is `'stdpar'`, then both actions in the preceding bullets will occur.
- The default is the probe file is created with all parameters initialized to zero.

Examples
```
addprobe('idpfg')
addprobe('idpfg','stdpar')
addprobe('idpfg','system','stdpar')
```

See also *NMR Spectroscopy User Guide; VnmrJ Walkup*

Related

addnucleus	Add new nucleus to existing probe file (M)
deletenucleus	Removes nucleus entry to probe file (M)
getparam	Receive parameter from probe file (M)
probe	Probe type (P)
setparams	Write parameter to current probe file (M)

adept Automatic DEPT analysis and spectrum editing (C)

Syntax `adept<(<'noll'><,'coef'><,'theory'>)>`

Description Automatically analyzes a set of four DEPT spectra and edits the spectra so that the spectra is arrayed as follows:

- #4 is CH_3 carbons only
- #3 is CH_2 carbons only
- #2 is CH carbons only
- #1 is all protonated carbons

Because `adept` modifies the transformed data, it should not be repeated without retransforming the data between calls. `adept` produces a text file `dept.out` in the current experiment directory, which contains the result of the analysis.

Arguments The following keyword arguments can be supplied in any order:

`'noll'` causes the line listing to be skipped. If `'noll'` is not supplied as an argument, `adept` first performs a line listing. In that case, the threshold parameter `th` must be set properly before starting `adept`.

`'coef'` causes the combination coefficients to be printed.

`'theory'` causes theoretical coefficients to be used. The default is optimized coefficients.

Examples
```
adept
adept('coef')
adept('theory','noll')
```

See also *NMR Spectroscopy User Guide*

Related

`autodept`	Automated complete analysis of DEPT data (M)
`Dept`	Set up parameters for DEPT experiment
`deptproc`	Process DEPT data (M)
`padept`	Perform `adept` analysis and plot resulting spectra (C)
`pldept`	Plot DEPT data, edited or unedited (M)
`th`	Threshold (P)

aexppl — Automatic plot of spectral expansion (M)

Syntax `aexppl<(expansion_factor)>`

Description Plots automatically expansions of given regions. Regions have to be defined first by using the `region` command or by using the cursors in `ds`.

Arguments `expansion_ factor` is a spectral expansion factor in units of Hz/mm. The default is 2 Hz/mm.

Examples
```
aexppl
aexppl(20)
```

See also *NMR Spectroscopy User Guide*

Related

`ds`	Display a spectrum (C)
`region`	Divide spectrum into regions (C)

ai — Select absolute-intensity mode (C)

Syntax `ai`

Description Selects the *absolute-intensity display mode* in which the scale is kept constant from spectrum to spectrum to allow comparison of peak heights from one spectrum to another. The alternative is the normalized-intensity display mode (`nm`) in which spectra are scaled so that the largest peak in the spectrum is `vs` mm high. The modes are mutually exclusive—the system is always in either nm or ai mode. Enter aig? to determine which mode is currently active.

See also *NMR Spectroscopy User Guide*

Related
aig	Absolute intensity group (P)
nm	Select normalized-intensity mode (C)
vs	Vertical scale (P)

aig Absolute-intensity group (P)

Description Contains the result of the ai or nm command. aig is not set in the usual way but can be queried (aig?) to determine which display mode is active.

Values 'ai' indicates the absolute-intensity display mode is active.

'nm' indicates the normalized-intensity display mode is active.

See also *NMR Spectroscopy User Guide*

Related
ai	Select absolute intensity mode (C)
dmg	Display mode in directly detected dimension (P)
nm	Select normalized-intensity mode (C)
?	Display individual parameter value (C)

alfa Set alfa delay before acquisition (P)

Description After the final event in the pulse sequence, including any receiver gate times occurring following the final pulse, acquisition occurs after a delay. This delay includes a fixed part, alfa, and a variable part, 1/(beta*fb).

- On systems with 4-pole Butterworth filters, beta is 2.
- On systems with 8-pole Butterworth (200-kHz) filters, beta is 3.8.
- On systems with 8-pole elliptical filters, beta is 1.29.
- On Systems with 4-pole Bessel filters, beta is 2.3 (only systems with 2-MHz and 5-MHz Analog-to-Digital Converter boards use this filter).

Because the total delay before acquisition is the sum of alfa and 1/(beta*fb), it is possible to shorten the delay beyond “normal” values by setting alfa negative (to a maximum of 1/(beta*fb)). The macros hoult and calfa frequently result in such negative values of alfa.

To set alfa to a negative number, use either the setvalue command to enter a specific value of alfa, or use the setlimit command to allow entry of negative values of alfa directly from the keyboard.

Values 0 to 100,000,000; in µs.

See also *NMR Spectroscopy User Guide*

Related

calfa	Recalculate alfa so that first-order phase is zero (M)
fb	Filter bandwidth (P)
hoult	Set parameters alfa and rof2 according to Hoult (M)
rof2	Receiver gating time following pulse (P)
setlimit	Set limits of a parameter in a tree (C)
setlp0	Set parameters for zero linear phase (M)
setvalue	Set value of any parameter in a tree (C)

alock Automatic lock control (P)

Description Governs Autolock control following the insertion of a sample with change or sample, and following initiation of an acquisition with the go, ga, or au. Manual adjustment of lock power, gain, and phase is possible using the acqi command.

Values Possible values are 'a', 'auto', 'n', 's', 'samp', 'u', or 'y', where:

'a' or 'auto' selects the optimizing Autolock function, which performs a lock capture and an automatic lock power and gain adjustment before data acquisition begins (lock phase is *not* optimized).

'n' leaves the lock in its current state.

's' or 'samp' selects the optimizing Autolock function, which performs a lock capture and an automatic lock power and gain adjustment before data acquisition begins (lock phase is *not* optimized) but only if the sample has just been changed.

'u' turns lock off so that the experiment runs unlocked.

'y' turns on the software Autolock function, which searches for the correct Z0 value only.

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display process (C)
au	Submit experiment to acquisition and process data (C)
change	Submit a change sample experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (C)
gf	Prepare parameters for FID/spectrum display in acqi (M)
go	Submit experiment to acquisition (C)
lock	Submit an Autolock experiment to acquisition (C)
sample	Submit change sample, Autoshim experiment to acquisition (M)

ampmode Independent control of amplifier mode (P)

Description Gives override capability over the default selection of amplifier modes. Unless overridden, the usage of rf channels determines whether the amplifier for a channel is in pulse, CW (continuous wave), or idle mode:

- Observe channel is set to the pulse mode.
- Other used channels are set to the CW mode.
- Any unused channels are set to the idle mode.

The `ampmode` parameter can be used to override this selection.

`ampmode` does not normally exist but can be created by the user with the command `create('ampmode','flag')`.

Values List of characters in which the mode of the first amplifier is determined by the first character, the mode of the second amplifier by the second character, and so on. For each amplifier, one of the following characters is used:

- `'c'` selects CW mode.
- `'i'` selects idle mode.
- `'p'` selects pulse mode.
- `'d'` selects default behavior.

For example, `ampmode='ddp'` selects default behavior for the first two amplifiers and forces the third channel amplifier into pulse mode. Additional filtering is usually required when an amplifier in the same band as the observe amplifier is placed in the CW mode.

See also *VnmrJ User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`dn`	Nucleus for the first decoupler (P)
`tn`	Nucleus for observe transmitter (P)

amptype Amplifier type (P)

Description Specifies the type of amplifier on each rf channel of the spectrometer. The value is set in the Spectrometer Configuration window (opened from `config`) using the label Type of Amplifier.

For each channel, the types are Class C, Linear Full Band, Linear Low Band, Linear Broadband, or, for the fourth channel only, Shared. Selecting Shared means that the amplifier is fully configured for the third channel, and that the fourth channel shares this amplifier with the third channel.

When a type is selected for a channel, a letter (one of the values described below) is added to the value of `amptype`. For example, a system already set to Linear Full Band on the observe transmitter channel and the first decoupler channel would have `amptype='aa'`.

Selecting the third channel as Linear Low Band would set `amptype='aal'`. Finally, selecting Shared for the fourth channel would set `amptype='aaln'`.

Values `'a'` indicates the channel uses a linear full-band amplifier. A full-band amplifier has two outputs: 12 MHz to ^{31}P, and $^{19}F/^{1}H$.

`'b'` indicates the system uses a linear broadband amplifier.

`'c'` indicates the system uses a class C amplifier.

`'l'` indicates the channel uses a linear low-band amplifier. A low-band amplifier has one output from 12 MHz to ^{31}P only.

`'n'` indicates the fourth channel shares a linear amplifier with the third.

See also *NMR Spectroscopy User Guide, VnmrJ User Programming*

Related `config` Display current configuration and possibly change it (M)

`analyz` Calculate standard peak height (M)

Syntax `analyz($option,$title)`

Description Macro to calculate average peak height and standard deviation and/or average phase and standard deviation.

Arguments $option ='n' for amplitude and phase, 'a' for amplitude only, and 'p' for phase only. The $title option puts a title on the plot.

Examples `analyz` – Does analysis for both amplitude and phase
`analyz('p')` – Does analysis for phase only
`analyz('n','Stability')` – Does analysis for amplitude and phase and puts title “Stability” on the plot.

`analyze` Generalized curve fitting (C)

Syntax (curve fitting) `analyze('expfit',xarray<,options>)`
(regression) `analyze('expfit','regression'<,options>)`

Description Provides interface to curve fitting program `expfit` (using the curve fitting syntax), supplying `expfit` with input data in the form of the text file `analyze.inp` in the current experiment. `expfit` can be called from UNIX with the syntax:

```
expfit options <analyze.inp >analyze.list
```

`expfit` does a least-squares curve fitting to the data supplied in `analyze.inp`. Macros are available for the specialized uses of `analyze`, such as the `'T1'` and `'kinetics'` options. These macros avoid the need to select options and get the correct file format.

In the regression mode (using the regression syntax above), the type of curve fitting, (`'poly1',...`) must be selected. The regression section in the manual *NMR Spectroscopy User Guide* gives the input file format and describes the menus that permit choices indirectly through menu buttons.

The text file `analyze.inp` for the options `'T1'`, `'T2'`, `'kinetics'`, `'contact_time'`, and `'regression'` contains the following lines (note that (1), (2), (3), etc. do not appear in the file but are used to identify lines in the explanation):

```
(1)   <text line>
(2)   <text line>
(3)   npeaks npairs <xscale> <yscale>
(4)   <NEXT npairs1>
(5)   peaks
(6)   x y
(6)   x y
   ...
(4)   <NEXT npairs2>
(5)   peaks
(6)   x y
(6)   x y
   ...
```

Line-by-line explanation:

(1) Optional descriptive text line, for regression only. Omit line otherwise.

(2) Optional y-axis title, for regression only. Omit line otherwise.

(3) Line containing an integer for the number of peaks (`npeaks`) followed by another integer for the number of (`x,y`) pairs per peak (`npairs`). If regression, the x-scale type and y-scale type are also listed.

(4) In the regression mode, a line beginning with the keyword `NEXT` is inserted at the start of each data set when the number of pairs per peak is variable. In this case, the number of (`x,y`) pairs for the peak (`npair1`, `npair2`, etc.) is also given on the line.

(5) Peak index.

(6) Data pairs, one to a line, are listed by peak in the following order:

```
x y    (first peak, first pair)
x y    (first peak, second pair)
...
x y    (second peak, first pair)
x y    (second peak, second pair)
...
```

In the regression mode, the line beginning with `NEXT` is inserted at the start of the data for each peak when the number of pairs per peak is variable. In this case, the header contains the maximum number of pairs for any peak.

For `'T1'`, `'T2'`, `'kinetics'`, and `'contact_time'`, information from the file `fp.out` and values of the arrayed parameter `xarray` are used to construct the file; thus, it is necessary to run `fp` prior to `analyze`.

For regression, `analyze.inp` is made by running `expl('regression')`. If the regression mode is not selected, `analyze.inp` may be slightly different.

In addition to output to the standard output, which is usually directed to `analyze.list`, `expfit` makes a file `analyze.out`, which is used by `expl` to display the results of the analysis.

User-supplied analysis programs can be called by `analyze` in place of `expfit`. Such programs should read their input from `stdin` and write the output listing to `stdout`. No `analyze.out` file needs to be generated unless display by `expl` is desired. Use the program `expfit` as a model.

Arguments `'expfit'` is a required first argument.

`xarray` is the name of the parameter array holding x-values in `'T1'`, `'T2'`, `'kinetics'`, and `'contact_time'`, and is used only with these options.

`'regression'` sets regression mode and signifies generalized curve fitting with choices `'poly1'`, `'poly2'`, `'poly3'`, and `'exp'`.

`options` are any of the following keywords:

- `'T1'` sets T_1 analysis (the default).
- `'T2'` sets T_2 analysis.
- `'kinetics'` sets kinetics analysis, with decreasing peak height. The last four arguments are used to add a time offset between the array elements. In the example below, the time increment is `(d1 + d2 + at) * nt`.
- `'increment'` sets kinetics analysis, with increasing peak height. The last four arguments are the same as in the kinetics case.
- `'list'` makes an extended listing for each peak.
- `'diffusion'` sets a special analysis for diffusion experiments.
- `'contact_time'` sets a special analysis for solids cross-polarization spin-lock experiments.
- `'poly1'` sets a linear fitting. It is used in regression mode only.
- `'poly2'` sets a quadratic fitting. It is used in regression mode only.
- `'poly3'` sets a cubic fitting It is used in regression mode only.
- `'exp'` sets exponential curve fitting. It is used in regression mode only.

Examples
```
analyze('expfit','d2','T1','list')
analyze('expfit','pad',kinetics','list',d1,d2,at,nt)
analyze('expfit','p2','contact_time','list')
analyze('expfit','regression','poly1','list')
```

See also *NMR Spectroscopy User Guide*

Related

`contact_time`	MAS cross-polarization spin-lock contact time (M)
`expl`	Display exponential or polynomial curves (C)
`pexpl`	Plot exponential or polynomial curves (C)
`kini`	Kinetics analysis, increasing intensity (M)

t1	T_1 exponential analysis (M)
t2	T_2 exponential analysis (M)

annotation — Display annotation specified by the parameter "template" or the default

Syntax `annotation(<template>, <x, y, width, height>)`

pannotation(<template>, <x, y, width, height>)

Applicability VnmrJ 3.1

Description "`annotation`" and "`pannotation`" will display or plot annotation specified by the parameter "`template`" or the default.

Arguments `template`: The name of template of annotation to be displayed. The default name is 'default'.

x, y: The origin point on the screen or plotter, in `mm`.

width: The width on the screen or plotter, in `mm`.

height: The height on the screen or plotter, in `mm`.

ap — Print out "all" parameters (C)

Applicability VnmrJ

Syntax `ap('template_name',<'filename'>)`

Description Print a parameter list. The *User Programming* Manual describes the rules for building a template for the `ap` commands. The string parameter `ap` normally controls how the command, `ap`, displays the parameters. Use command `paramvi('ap')` to modify the `ap` parameter. The `ap` command writes the parameter list to a file if `filename` is provided as the second argument.

Arguments `template_name` template name must be the first argument.

`filename` optional, name of file to which the parameters are written.

Examples `ap('ap','apout')` — writes the parameter list using defined by the `ap` parameter to the file `apout`.

`ap('newap')`

See also *NMR Spectroscopy User Guide; VnmrJ User Programming*

Related

addpar	Add selected parameters to the current experiment (M)
ap	"All" parameters display control (P)
dg	Display group of acquisition/processing parameters (C)
hpa	Plot parameters on special preprinted chart paper (C)
pap	Plot out "all" parameters (C)
paramvi	Edit a variable and its attributes with `vi` text editor (C)
ppa	Plot a parameter list in "English" (M)

ap "All" parameters display control (P)

Description: Controls the display of the ap and pap commands to print and plot a parameter list. Use paramvi('ap') to modify the string value of ap.

See also: *NMR Spectroscopy User Guide*; *VnmrJ User Programming*

Related:

ap	Print out "all" parameters (C)
dg	Display group of acquisition/processing parameters (C)
pap	Plot out "all" parameters (C)
paramvi	Edit a variable and its attributes with vi text editor (C)

apa Plot parameters automatically (M)

Syntax: apa

Description: Selects automatically the appropriate command on different plotter devices to plot the parameter list.

See also: *VnmrJ User Programming*

Related:

hpa	Plot parameters on special preprinted chart paper (C)
ppa	Plot a parameter list in "English" (M)

aph Automatic phase adjustment of spectra (C)

Syntax: aph<:$ok,$rp,$lp>

Description: Automatically calculates the phase parameters lp and rp required to produce an absorption mode spectrum and applies these parameters to the current spectrum. Values calculated do *not* depend on the initial values of lp and rp.

Arguments: $ok is 1 if the phase adjustment succeeds, or 0 if the adjustment fails.

$rp is the calculated value of rp. If $rp is requested as a return value, rp is returned but not applied to the current spectrum.

$lp is the calculated value of lp. If $lp is requested as a return value, lp is returned but not applied to the current spectrum.

See also: *NMR Spectroscopy User Guide*

Related:

aph0	Automatic phase of zero-order term (C)
aphx	Perform optimized automatic phasing (M)
lp	First-order phase in directly detected dimension (P)
rp	Zero-order phase in directly detected dimension (P)

aph0 Automatic phase of zero-order term (C)

Syntax `aph0<:$ok,$rp,$lp>`

Description Automatically adjusts only the zero-order frequency-independent term `rp` and does not rely on the frequency-dependent term `lp` being previously adjusted. In favorable circumstances, spectra may be obtained in such a way that only `rp` is expected to change. In these cases, if `lp` has been determined for one spectrum, then `rp` only can be computer-adjusted for subsequent spectra by `aph0` ("aph-zero"). Note that `aph0` does not correctly phase an exactly on-resonance peak.

Arguments `$ok` is 1 if the phase adjustment succeeds, or 0 if the adjustment fails.

`$rp` is the calculated value of `rp`.

`$lp` is the current value of `lp`.

See also *NMR Spectroscopy User Guide*

Related

`aph`	Automatic phase adjustment of spectra (C)
`aphx`	Perform optimized automatic phasing (M)
`lp`	First-order phase in directly detected dimension (P)
`rp`	Zero-order phase in directly detected dimension (P)

aphb Auto phasing for Bruker data (C)

Syntax `aphb<(threshold)>`

Description Phases Bruker data using the autophasing program.

Arguments `threshold` determines if a data point is large enough to qualify it as part of a peak. If no argument is given, or if the value is equal to or less than 0, the threshold is calculated from the spectrum.

Examples

```
aphb
aphb(2)
```

See also *NMR Spectroscopy User Guide*

Related

`aph`	Automatic phase adjustment of spectra (C)
`aph0`	Automatic phase of zero-order term only (C)

aphx Perform optimized automatic phasing (M)

Syntax `aphx`

Description Optimizes parameters and arguments for the `aph` command. `aphx` first performs an `aph` then calculates a theoretical value for `lp`. If `lp` set by the `aph` is different from the calculated value by 10 per cent, the calculated value is used and an `aph0` is performed.

See also	*NMR Spectroscopy User Guide*	
Related	`aph`	Automatic phase adjustment of spectra (C)
	`aph0`	Automatic phase of zero-order term only (C)
	`lp`	First order phase along directly detected dimension (P)

appdir Application directory information

Syntax `appdir('info'):$num` - Applications directories information

`appdir('info',n):$label,$path` - Application directory information

Applicability VnmrJ 3.1

Description An application directory is a directory where VnmrJ can look for templates, `maclib`, `manual`, `menujlib`, `parlib`, `probes`, `psg`, `psglib`, `seqlib`, `shims`, `tablib`, `shapelib`, `gshimlib`, and `mollib` directories. It will not look for `expN` directories, global, or other files or directories. The `exists` command has been enhanced to search for other files and directories in the applications directories, allowing users flexibility to customize their applications. The `appdirs` macro starts an editor to set applications directories.

Arguments The `appdir('info')` command will tell you the number of application directories that are currently enabled. This value can be returned to a parameter as in `appdir('info'):$num`

The label and path of a specific application directory can be returned

by supplying a number after the '`info`' keyword. The number must be between

1 and the total number of applications directories ($num from above).

Examples The following macro lists the current applications directories:

```
clear
write('alpha','Applications Directories')
appdir('info'):$num
$i = 0
while ($i < $num) do
  $i=$i+1
  appdir('info',$i):$label,$path
  if ($label = '') then
    write('alpha','%d: "appdir %d" has path
"%s"',$i,$i,$path)
  else
    write('alpha','%d: "%s" has path
"%s"',$i,$label,$path)
  endif
endwhile
```

See also The "`which`" macro for another example of the use of the `appdir` command

appdirs Starts Applications Directory Editor (M)

Applicability ALL

Syntax `appdirs`

Description The **appdirs** macro brings up an editor to set the applications directories.The top section of the editor has rows consisting of a menu and two entry boxes.

Values **Menu selections**:

Enabled — enable an application directory.

Disabled — disable an application directory.

Remove(d) — initial setting for other row and the and empty entry boxes.

Set an application directory menu to Remove(d) to completely remove it.

Fields in each row:

Applications directory path.

A comment can be added to the second entry box.

Radio- button choices:

Save as private applications directories — sets the applications directories for the current operator only.

Reset to system default applications directories — removes any private applications directories and return to the standard default set.

Save the applications directories for global use — available only to users with write permission for VnmrJ system files. A name must be provided for this choice. This will affect all users the administrator has set that name as their appdirs setting. The Agilent default names are Experimental, Walkup, Imaging, and LcNmrMs.

Buttons:

OK — exit the editor and apply the selections made in the editor.

Cancel — exit the editor and abort the editor session, making no changes to the applications directories.

See also *VnmrJ Installation and Administration*

Related `exists` Checks if parameter, file, or macro exists and file type (C)

appmode Application mode (P)

Description A global parameter that allows selection of specialized system applications modes, such as imaging, by setting the global parameters `sysmaclibpath`, `sysmenulipath`, and `syshelppath`.

For example, in `/vnmr/maclib` is a subdirectory `maclib.imaging` that contains macros used primarily with imaging applications. Similarly, in `/vnmr/menulib` is a subdirectory `menulib.imaging` for imaging- related menus. By separating the imaging macros and menus into subdirectories, access to imaging-specific macros and menus is more convenient. This separation also allows minor modifications to some macros and menus while retaining the names that are in common use or required by other VnmrJ commands.

The value of appmode are set from either the System settings dialog in the Utilities menu or the VnmrJ Admin interface.

Values 'standard' sets standard application mode.
'imaging' sets imaging application mode.
'autotest' sets autotest application mode

apptype Application type (P)

Description Specifies the application type, the group of pulse sequences to which a pulse sequence belongs. It is used by the execpars macros to specify the actions executed by the protocol for a pulse sequence. The actions are common to the group of pulse sequences specified by the apptype.

Values See the execpars directory in /vnmr.

See also *VnmrJ Imaging, User Guide* and *NMR Spectroscopy User Guide*

Related

cqexp	Load experiment from protocol (M)
execpars	Set up the exec parameters (M)
execsetup	Execute setup macro (P)
execprep	Execute prepare macro (P)
execprescan	Execute prescan macro (P)
execpreocess	Execute processing macro (P)
execplot	Execute plotting macro (P)
sqexp	Load experiment from protocol (M)

Apt Set up parameters for APT experiment (M)

Description Converts a parameter set to the APT (attached proton test) experiment.

See also *NMR Spectroscopy User Guide*

Related

aptaph	Automatic processing for APT spectra (M)
capt	Automated carbon and APT acquisition (M)
hcapt	Automated proton, carbon, and APT acquisition (M)

aptaph Automatic processing for APT spectra (M)

Syntax aptaph

Description Automatically phases APT spectra.

See also *NMR Spectroscopy User Guide*

Related `Apt` Set up parameters for APT pulse sequence (M)

`array` Easy entry of linearly spaced array values (M)

Syntax `array<(parameter<,number_steps,start,step_size)>`

Description Arrays a parameter to the number of steps, starting value and step size given by the user. All values of the array will satisfy the limits of the parameter.

If `array` is typed with none or only some of its arguments, you enter an interactive mode in which you are asked for the missing values.

Arguments `parameter` is the name of the parameter to be arrayed. The default is an interactive mode in which you are prompted for the parameter. Only numeric parameters can be arrayed.

`number_steps` is the number of values of the parameter. The default is an interactive mode in which you are prompted for the number of steps.

`start` is the starting value of the parameter array. The default is an interactive mode in which you are prompted for the starting value.

`step_size` is the magnitude of the difference between elements in the array. The default is an interactive mode in which you are prompted for the step size.

Examples
```
array
array('pw')
array('tof',40,1400,-50)
```

See also *NMR Spectroscopy User Guide*

`array` Parameter order and precedence (P)

Description Whenever an array of one or more parameters is set up, the string parameter `array` tells the system the name of the parameter or parameters that are arrayed and the order and precedence in which the arraying is to take place. The parameter `array` is automatically updated when acquisition parameters are set. “Diagonal arrays” (those corresponding to using parentheses in the parameter `array`) must be entered by hand.

Values `''` (two single quotes with no space between) indicates no parameter is arrayed.

`'x'` indicates the parameter `x` is arrayed.

`'x,y'` indicates the parameters `x` and `y` are arrayed, with `y` taking precedence. That is, the order of the experiments is x_1y_1, x_1y_2,... x_1y_n, x_2y_1, x_2y_2,... x_2y_n,... x_my_n, with a total of $m \times n$ experiments being performed.

'y,x' indicates the parameters x and y are arrayed, with x taking precedence. That is, the order of the experiments is x_1y_1, x_2y_1,... x_ny_1, x_1y_2, x_2y_2,... x_my_2,... x_my_n, with total of $m \times n$ experiments being performed.

'(x,y)' indicates the parameters x and y are jointly arrayed. The number of elements of the parameters x and y must be identical, and the order of experiments is x_1y_1, x_2y_2,... x_ny_n, with n experiments being performed.

Joint arrays can have up to 10 parameters. Regular multiple arrays can have up to 20 parameters, with each parameter being either a simple parameter or a diagonal array. The total number of elements in all arrays can be $2^{32}-1$.

See also *NMR Spectroscopy User Guide*

Related `array` Easy entry of linearly spaced array values (M)

arraydim Dimension of experiment (P)

Description After calcdim calculates the dimension of an experiment, the result is put into the parameter arraydim. If an experiment is arrayed, arraydim is the product of the size of the arrays.

See also *NMR Spectroscopy User Guide*

Related
- calcdim Calculate dimension of experiment (C)
- celem Completed FID elements (P)

array2csv Formats Array into Comma Separate Variable

Description This macro converts an array into a comma separated variable.

Syntax `array2csv('parameter'):$csv`

Examples array2csv('ni'):$increments

Related array2strsv, array2string, string2array, strsv2array, csv2array

array2string Formats Array into String

Description This macro converts an array into a string variable.

Syntax `array2string('parameter'):$string`

Examples array2string ('d3'):$delay

`array2strsv` Formats Array into String Separated Variable

Description This macro formats an array into a string separated variable.

Syntax `array2strsv('parameter'):$strsv`

Examples `array2strsv` ('ni'):$increments

`asin` Find arc sine of number (C)

Syntax `asin(value)<:n>`

Description Finds the arc sine (also called the inverse sine) of a number.

Arguments `value` is a number in the range of ±1.0.

`n` is a return argument giving the arc sine, in radians, of `value`. The default is to display the arc sine value in the status window.

Examples `asin(.5)`
`asin(val):asin_val`

See also *VnmrJ User Programming*

Related `sin` Find sine value of an angle (C)

asize — Make plot resolution along f_1 and f_2 the same (M)

Syntax `asize`

Description Adjusts the 2D display parameters (`sc`, `wc`, `sc2`, and `wc2`) so that the displayed resolution along both f_1 and f_2 is the same. It is not suggested for heteronuclear experiments where the chemical shift spread of one nucleus is much greater than that of the other.

See also *NMR Spectroscopy User Guide*

Related

`sc`	Start of chart (P)
`sc2`	Start of chart in second direction (P)
`wc`	Width of chart (P)
`wc2`	Width of chart in second direction (P)

assign — Assign transitions to experimental lines (M)

Syntax (1) `assign<('mark')>`
(2) `assign(transistion_number,line_number)`

Description Assigns the nearest calculated transition to the lines from a `dll` or `nll` listing after `spinll` has placed them in `slfreq`. All lines may not be assigned and transitions must be greater than `sth`. The next `spins('iterate')` determines new parameters to minimize the differences in position of the assigned pairs.

Arguments `'mark'` makes assign use the lines selected with the mark button in place of `dll`. The results of the `mark` operation are stored in the file `mark1d.out`, which is cleared by the command `mark('reset')`.

`transition_number` is a single calculated transition number that is assigned to a line from the `dll` listing.

`line_number` is the index of the line from the `dll` listing. Setting `line_number=0` removes an assignment from a calculated transition.

Examples

```
assign
assign('mark')
assign(4,0)
```

See also *NMR Spectroscopy User Guide*

Related

`dll`	Display listed line frequencies and intensities (C)
`mark`	Determine intensity of the spectrum at a point (C)
`nll`	Find line frequencies and intensities (C)
`slfreq`	Measured line frequencies (P)
`spinll`	Set up `slfreq` array (M)
`spins`	Perform spin simulation calculation (C)
`sth`	Minimum intensity threshold (P)

at — Acquisition time (P)

Description Length of time during which each FID is acquired. Since the sampling rate is determined by the spectral width sw, the total number of data points to be acquired (2*sw*at) is automatically determined and displayed as the parameter np. at can be entered indirectly by using the parameter np.

Values Number, in seconds. A value that gives a number of data points that is not a multiple of 2 is readjusted automatically to be a multiple of 2.

See also *NMR Spectroscopy User Guide; VnmrJ User Programming*

Related

np	Number of data points (P)
sw	Spectral width in directly detected dimension (P)

atan — Find arc tangent of a number (C)

Syntax atan(value)<:n>

Description Finds the arc tangent (also called the inverse tangent) of a number.

Arguments value is a number between π/2 and –π/2.

n is a return argument giving the arc tangent, in radians, of value. The default is to display the arc tangent value in the status window.

Examples

```
atan(.5)
atan(val):atan_val
```

See also *VnmrJ User Programming*

Related

sin	Find sine value of an angle (C)

atan2 — Find arc tangent of two numbers (C)

Syntax atan2(y,x)<:n>

Description Finds the arc tangent (also called the inverse tangent) of the quotient of two numbers.

Arguments y and x are two numbers, where the quotient y/x is between π/2 and –π/2 and x is not equal to zero.

n is a return argument giving the arc tangent, in radians, of y/x. The default is to display the arc tangent value in the status window.

Examples

```
atan2(1,2)
atan2(val):atan2_val
```

See also *VnmrJ User Programming*

Related

sin	Find sine value of an angle (C)

atcmd Call a macro at a specified time (M)

Syntax
```
atcmd('macro','timespec')
atcmd('macro','timespec','active')
atcmd('macro','timespec','start')
atcmd('macro','timespec','active','start')
atcmd('macro','cancel')
atcmd('macro','list')
atcmd
```

Description
atcmd will call a macro at the specified time. It only functions on a spectrometer. If the 'active' argument is given, the macro will be executed by the Vnmr process that specified atcmd. If that process is no longer active, the macro will be removed from the database. If the 'active' argument is not give, then a background Vnmr will be started to execute the macro. This background Vnmr will not be started in an experiment. Therefore, the macro will need to execute jexp or run commands or macros which do not need experiment parameters. It will have access to global and systemglobal parameters. The bootup macro will not be executed automatically. It can be called from the atcmd macro.

Arguments
When called with arguments, atcmd updates the database with the supplied information. It does not start the process that calls the macros at the specified times. atcmd with no arguments starts the program that calls the macros at the specified times.

timespec -- has the format hh:mm <mon tue wed thur fri sat sun> A 24 hour clock is used -- midnight is 0:0, noon is 12:00.

day -- If the optional day field is used, the command will be repeated on that day at the appointed time. The day fields are case insensitive. For monday, wednesday, and friday only a single character is needed. More can be used. For tuesday, thursday, saturday, and sunday, at least two characters must be given.

cancel -- If the cancel argument is given, it will cancel all the commands that match the supplied macro. For example, if you specify cmda to be run at 8:00 on mon and 9:00 on tue, then atcmd('cancel','cmda') will cancel both of them. If the macro is '', the cancel option will cancel all atcmd macros.

list -- The list argument lists the timespec for all the atcmds that match the supplied macro. If the macro is '', the list option lists all of the atcmd macros and their timespecs. Optional arguments can be returned. The first is the number of atcmds. The macro and timespec for each atcmd can be returned.

When the command specified by atcmd is executed in background, it will be executed using the environment of the user who requested the atcmd. Also, the background VnmrJ will initially not be joined to a specific experiment.

Examples
atcmd('echo(`good morning`)','8:00 mon tue wed thu fri')
Displays a welcome message every weekday at 8:00 am.

```
atcmd('echo(`What are you doing here on a
weekend?`)','8:00 Sat Sun')
```
Questions your intentions on the weekend.

```
atcmd('startNightQueue','22:00')
```
Runs the macro startNightQueue at 22 hr. (10:00pm).

```
atcmd('startNightQueue','cancel')
```
Cancels the scheduled startNightQueue cmd

```
atcmd('','cancel')
```
Cancels all scheduled commands

```
atcmd('','list')
```
Lists all scheduled commands

atext — Append string to current experiment text file (M)

Syntax	`atext(string)`
Description	Adds a line of text to the current experiment text file.
Arguments	`string` is a single line of text.
Examples	`atext('T1 Experiment')`
See also	*NMR Spectroscopy User Guide*

Related

`ctext`	Clear the text of the current experiment (C)
`text`	Display text or set new text for current experiment (C)
`write`	Write formatted text to a device (C)

attval — Calculate pulse width (M)

Syntax	`attval (pw,tpwr)`
Description	Calculates the pulse width and B_1 field at every transmitter power. A low transmitter power should be used where the amplifier is not in compression. Calculation is not valid where amplifier is in compression.
Arguments	`pw` is the pulse width. `tpwr` is the transmitter power.
Examples	`attval(7.0,59)`

atune — ProTune Present (P)

Description	Hardware configuration parameter specifying if ProTune is or is not present. Parameter is set in the System Configuration window.

Arguments 'y' ProTune is present
'n' ProTune not is present

See also *VnmrJ Installation and Administration*

Related	wtune	Specify when to tune (P)
	tupwr	Transmitter power used in tuning (P)

au Submit experiment to acquisition and process data (M)

Syntax `au<(<'nocheck'><,'next'><,'wait'>)>`

Description Performs the experiment described by the current acquisition parameters, checking the parameters loc, spin, gain, wshim, load, and method to determine the necessity to perform various actions in addition to simple data acquisition. This may involve a single FID or multiple FIDs, as in the case of arrays or 2D experiments. au causes the data to automatically be processed according to the following parameters:

- wbs specifies what happens after each block.
- wnt specifies what happens after each FID is collected.
- wexp specifies what happens when the entire acquisition is complete (which may involve several complete FIDs in the case of 1D arrays or 2D experiments).

Before starting the experiment, au executes the two user-created macros if they exist. The first is usergo, a macro that allows the user to set up general conditions for the experiment. The second is a macro whose name is formed by go_ followed by the name of the pulse sequence (from seqfil) to be used (e.g., go_s2pul, go_dept). This macro allows a user to set up experiment conditions suited to a particular sequence.

Arguments 'nocheck' is a keyword to override checking if there is insufficient free disk space for the complete 1D or 2D FID data set to be acquired.

'next' is a keyword to put the experiment started with au('next') at the head of the queue of experiments to be submitted to acquisition.

'wait' is a keyword to stop submission of experiments to acquisition until wexp processing of the experiment, started with au('wait'), is finished.

Examples
```
au
au('wait')
```

See also *NMR Spectroscopy User Guide*

Related	auto_au	Controlling macro for automation (M)
	change	Submit a change sample experiment to acquisition (M)
	ga	Submit experiment to acquisition and FT the result (M)
	gain	Receiver gain (P)
	go	Submit experiment to acquisition (M)
	go_	Pulse sequence setup macro called by go, ga, and au (M)
	load	Load status of displayed shims (P)

loc	Location of sample in tray (P)
lock	Submit an Autolock experiment to acquisition (C)
method	Autoshim method (P)
sample	Submit change sample, Autoshim experiment to acquisition (M)
seqfil	Pulse sequence name (P)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
spin	Sample spin rate (P)
su	Submit a setup experiment to acquisition (M)
usergo	Experiment setup macro called by go, ga, and au (M)
wbs	Specify action when bs transients accumulate (C)
wexp	Specify action when experiment completes (C)
wnt	Specify action when nt transients accumulate (C)
wshim	Conditions when shimming is performed (P)

AuCALch3i Set up autocalibration with CH3I sample (M)

Syntax `AuCALch3i`

Description Retrieves standard proton parameter set and setup for automatic calibration of proton (observe and decouple), carbon (observe and decouple), gcal, and C/H gradient ratio. The AuCALch3i macro is the same as the AuCALch3i1 macro.

AuCALch3i1 Get autocalibration with CH_3I sample (M)

Syntax `AuCALch3i1`

Description Retrieves standard proton parameter set and setup for automatic calibration of proton (observe and decouple), carbon (observe and decouple), gcal, and C/H gradient ratio. The AuCALch3i1 macro is the same as the AuCALch3i macro.

AuCALch3oh Set up autocalibration with Autotest sample (M)

Syntax `AuCALch3oh`

Description Retrieves standard proton parameter set and setup for automatic calibration of proton (observe), carbon (decouple), gcal and C/H gradient ratio. The AuCALch30h macro is the same as the AuCALch30h1 macro.

AuCALch3oh1 Get autocalibration with Autotest sample (M)

Syntax	AuCALch3oh1
Description	Retrieves standard proton parameter set and setup for automatic calibration of proton (observe), carbon (decouple), gcal and C/H gradient ratio. The AuCALch30h1 macro is the same as the AuCALch30h macro.

Aucalibz0 Automatic Hz to DAC calibration for Z0 (M)

Applicability	Autocalibration routine
Syntax	Called by Augmapz0 calibration routine.
Description	Called by Augmapz0 calibration routine. Automatically calibrates lock frequency change per Z0 DAC unit change. The calibrated value is written out in the probe file as lkhzdac parameter
See also	*System Administration*.

Related

Augmapz0	Automatic lock gradient map generation and Z0 calibration (M)
Aufindz0	Automatic adjustment of Z0 (M)

AuCdec Carbon decoupler calibration macro (M)

Syntax	AuCdec
Description	Used by AuCALch3i and AuCALch3oh autocalibration routines to do carbon decoupler calibrations. Calibrates high-power pulse widths and dmf.
See also	System Administration

Related

AuCALch3i	Get autocalibration with CH_3I sample (M)
AuCALch3oh	Get autocalibration with Autotest sample (M)
dmf	Decoupler modulation frequency for first decoupler (P)

AuCgrad Carbon/proton gradient ratio calibration macro (M)

Syntax	AuCgrad
Description	Used by AuCALch3i1 and AuCALch3oh1 autocalibration routines for C/H gradient ratio calibrations.
See also	System Administration

Related

AuCALch3i1	Get autocalibration with CH_3I sample (M)
AuCALch3oh1	Get autocalibration with Autotest sample (M)

AuCobs Carbon observe calibration macro (M)

Syntax AuCobs

Description Used by AuCALch3i1 autocalibration routines for carbon observe calibrations.

See also *System Administration*

Related AuCALch3i1 Get autocalibration with CH_3I sample (M)

audiofilter Audio filter board type (P)

Description Sets the type of audio filter board used where the spectral width (sw) is less than 100 kHz. The filter type is set in the Spectrometer Configuration window (opened from config) using the label Audio Filter Type.

Values 'b' indicates the system has a 100-kHz Butterworth filter board (100 kHz Butterworth choice in the Spectrometer Configuration window.).

'e' indicates the system has a 100-kHz elliptical filter board (100 kHz Elliptical choice in the Spectrometer Configuration window).

'2' indicates the system has a 200-kHz Butterworth filter board (200 kHz Butterworth choice in the Spectrometer Configuration window).

'5' indicates the system has a 500-kHz elliptical filter board (500 kHz Elliptical choice in the Spectrometer Configuration window).

See also System Administration

Related config Display current configuration and possibly change it (M)
sw Spectral width in directly detected dimension (P)

Aufindz0 Automatic adjustment of Z0 (M)

Syntax Aufindz0

Description Finds z0 by doing lock 1D spectrum. The frequency is then used along with the lkhzdac value in the probe file to calculate the z0 value for a given solvent and autolocking is done. This requires previous calibration of the hzdac value done using the Aucalibz0 macro.

See also *System Administration*

Related Aucalibz0 Automatic Hz to DAC calibration for Z0 (M)

Augcal Probe gcal calibration macro (M)

Syntax Augcal

Description	Used by AuCALch3i1 and AuCALch3oh1 autocalibration routines for probe gcal calibrations.
See also	System Administration
Related	AuCALch3i1 — Get autocalibration with CH_3I sample (M) AuCALch3oh1 — Get autocalibration with Autotest sample (M) gcal — Gradient calibration constant (P)

Augmap — Automated gradient map generation (M)

Syntax	`Augmap`
Description	Automatically adjusts gradient level, offset, window, and pulse width to generate a z1–z4 gradient map using a 2-Hz D_2O sample. This macro is used by the Aumakegmap auto gradient map generation macro and is applicable only for a lock gradient map.
See also	System Administration
Related	Aumakegmap — Auto lock gradient map generation (M) gzsize — Number of z-axis shims used by gradient shimming (P)

Augmapz0 — Automatic lock gradient map generation and z0 calibration (M)

Syntax	`Augmapz0`
Description	Using the 2-Hz D_2O sample, the `augmapz0` macro automatically creates a lock gradient map, followed by Hz to DAC calibration of Z0 for the autolocking procedure.
See also	*System Administration*
Related	Aucalibz0 — Automatic Hz to DAC calibration for Z0 (M) Aufindz0 — Automatic adjustment of Z0 (M)

AuHdec — Proton decoupler calibration (M)

Syntax	`AuHdec`
Description	Used by AuCALch3i autocalibration routine to do proton decoupler calibrations. Calibrates high-power pulse widths and dmf.
See also	System Administration
Related	AuCALch3i — Get autocalibration with CH3I sample (M) dmf — Decoupler modulation frequency for first decoupler (P)

AuHobs — Proton observe calibration macro (M)

Syntax `AuHobs`

Description Used by AuCALch3i and AuCALch3oh autocalibration routines for proton observe calibrations.

Aumakegmap — Auto lock gradient map generation (M)

Syntax `Aumakegmap(<lk or hs or H1>)`

Description Generates z1–z4 lock gradient ('lk' argument), lock homospoil ('hs' argument), or ^{1}H gradient map ('H1' argument). If no argument is given, the defaults is 'lk', if gradtype='nnh' to 'hs'. The doped 2-Hz D_2O should be used for hs and lk maps. H1 map is typically done on the sample. Automatically adjusts gradient level, offset, window, and pulse width. The map name is automatically stored in the probe file.

AuNuc — Get parameters for a given nucleus (M)

Syntax `AuNuc(nucleus,solvent)`

Description Retrieves standard parameter set for a given nucleus and adds all required parameters for Tcl/dg driven parameters. If no parameter set exists in stdpar, then carbon parameters are retrieved and tn changed.

auto — Prepare for an automation run (C)

Applicability Systems with an automatic sample changer.

Syntax `auto<(automation_directory)>`

Description Prepares the automation directory for an automation run. auto aborts if the spectrometer is already in automation mode.

Arguments automation_directory is the name of the automation directory, either an absolute UNIX path (i.e.the first character is a “/”) or a relative path (the first character is not a “/”). The default is the value of the parameter autodir. If for some reason autodir is not defined, you are prompted to provide the location of the automation directory. If not given as an argument, you are prompted for the path. If the automation directory is not present, it is created with full access for all users. auto aborts if it fails to create this directory.

Examples

```
auto
auto('/home/vnmr1/autorun_620')
```

See also *NMR Spectroscopy User Guide, VnmrJ User Programming, VnmrJ Walkup*

Related
auto_au Controlling macro for automation (M)
autodir Automation directory absolute pathname (P)
autogo Start an automation run (C)
autoname Prefix for automation data file (P)

auto Automation mode active (P)

Applicability Systems with an automatic sample changer.

Description A global variable that shows whether or not an automation run is in progress. Macros typically test this parameter because actions can differ between the automation and non-automation modes. The value of auto is not enterable by the user. An automation experiment is initiated with the autogo command. The auto parameter is only set to 'y' for those macros and commands that are run as part of an automation experiment.

Values 'y' indicates automation mode is active.

'n' indicates automation mode is inactive

See also *NMR Spectroscopy User Guide, VnmrJ User Programming, VnmrJ Walkup*.

Related
auto_au Controlling macro for automation (M)
autogo Start an automation run (C)
autora Resume suspended automation run (C)
autosa Suspend current automation run (C)

autoaa Abort an automation run with no error

Syntax autoaa

Applicability VnmrJ 3.1

Description This command is used to abort an experiment that has been submitted to automation. The currently running experiment will not be interrupted, but when it is over, the automation run will be terminated.

Arguments The macro consists of autosa and aa, run sequentially.

See also For further information on autosa or aa, see the manual.

Related
halt halt acquisition with no error
autora resume the interrupted automation run

auto_au Controlling macro for automation (M)

Applicability Systems with an automatic sample changer.

Syntax auto_au

Description Reads sampleinfo file (defines an automation experiment) using the lookup facility, sets the solvent and loc parameters based on the SOLVENT and SAMPLE# fields of sampleinfo, runs exec on the entry in the MACRO field, and writes the experiment text based on the TEXT field. After that, auto_au examines the value of the wexp parameter:

- If wexp is set to 'procplot', then auto_au calls au.
- If wexp is set to 'autolist', then auto_au inserts 'auto' as the first argument to autolist and calls au('wait').
- If wexp is set to anything else, auto_au does not call au.

If no data is generated from the requested MACRO field, due to an error or some other reason, auto_au sets the STATUS field to "No Data Requested."

auto_au is used only during automation and should not be called directly. It provides a starting point for all automation experiments. As such, it is a convenient point for user customization of automation.

See also *NMR Spectroscopy User Guide, VnmrJ User Programming, VnmrJ Walkup*

Related		
	au	Submit experiment to acquisition and process data (M)
	auto	Prepare for an automation run (C)
	autolist	Set up and start chained acquisition (M)
	exec	Execute a VnmrJ command (C)
	loc	Location of sample in tray (P)
	lookup	Look up words and lines from a text file (C)
	solvent	Lock solvent (P)
	wexp	When experiment completes (P)

autoq Utility commands for the automation queue

Syntax autoq

Applicability VnmrJ 3.1

Arguments This command can contain the following arguments:

- autoq('add',pathname): adds the sampleinfo file at pathname to the automation queue (enterQ). The pathname may contain multiple sampleinfo entries. An implicit lock is placed on the queue. An autosa / autora pair is not needed.
- autoq('add',pathname,'priority'): adds the sampleinfo file at pathname to the automation queue (enterQ) with queue name 'priority'. The pathname may contain multiple sampleinfo entries. For the enterQ, 'priority' is interpreted as adding it to the top of the file. An implicit lock is placed on the queue. An autosa / autora pair is not needed.

- autoq('lock'): locks the automation queue (enterQ) so other processes can not access it.
- autoq('lock',seconds): locks the automation queue (enterQ) so other processes will not access it. By default, all locks expire after 5 seconds. A second argument can set the expiration time between 1 and 15 seconds.
- autoq('unlock'): removes the lock.
- autoq('get',pathname): gets the next sampleinfo file from the automation queue (enterQ) and places it at pathname. An implicit lock is placed on the queue. An `autosa` / `autora` pair is not needed. This option will generally not be needed by user macros. This function is currently performed by `Autoproc`.
- autoq('sendmsg',message): Send `"message"` to whatever Vnmr session is listening. This is often used by background automation if it wants to send a message to a foreground Vnmr.
- autoq('recvmsg','on'): Turn on receiving messages from an `autoq('sendmsg',message)` command.
- autoq('recvmsg','off'): Turn off receiving messages from an `autoq('sendmsg',message)` command.

Autobackup — Back up current probe file (M)

Syntax `Autobackup`

Description Makes a copy of the probe file before starting the calibrations and prints the current calibration file. `Autobackup` is called by the autocalibration routines `AuCALch3i1` and `AuCALch3oh1`

autodept — Automated complete analysis of DEPT data (M)

Syntax `autodept`

Description Processes DEPT spectra, plots the unedited spectra, edits the spectra, plots the edited spectra, and prints outs editing information.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

`adept`	Automatic DEPT analysis and spectrum editing (C)
`Dept`	Set up parameters for DEPT experiment
`deptproc`	Process DEPT data (M)
`padept`	Perform `adept` analysis and plot resulting spectra (C)
`pldept`	Plot DEPT data, edited or unedited (M)

autodir Automation directory absolute path (P)

Applicability Systems with an automatic sample changer or LC-NMR accessory.

Description When using a sample changer, `autodir` is a global variable that holds the absolute path of the currently active automation directory. When VnmrJ is started, `autodir` is set to the absolute path of the last automation run.

When using the LC-NMR accessory, `autodir` specifies a directory in which experiments using a stored queue are saved.

See also *NMR Spectroscopy User Guide*

Related

`auto`	Set up an automation directory (C)
`autoname`	Prefix for automation data file (P)
`globalauto`	Automation directory name (P)
`walkup`	Walkup automation (M)

autogo Start automation run (C)

Applicability Systems with an automatic sample changer.

Syntax `autogo<(file<,automation_directory>)>`

Description Starts an automation run. The `autogo` parameter cannot be entered while the spectrometer is in automation mode. You must have an `enter` queue prepared to start an automation run. The queue is checked to verify that it was prepared using the `enter` command (`autogo` aborts if an error in the format is found.) Your automation directory is also checked for the presence of a non-empty `enter` queue (`autogo` aborts if the current queue in the automation directory is present and not empty). Finally, `autogo` checks the automation directory and runs the `auto` command if this directory is not present or another problem is found. When `autogo` completes, the system is in automation mode and your automation run starts.

Arguments `file` is the file name of your `enter` queue. The default is that the system prompts you for the location of the `enter` queue.

`automation_directory` is the pathname of the automation directory. The default is the current value of the parameter `autodir`.

Examples

```
autogo
autogo('MySamples')
autogo('MySamples','/home/vnmr1/AutoRun_621')
```

See also *NMR Spectroscopy User Guide*

Related

`auto`	Set up an automation directory (C)
`autodir`	Automation directory absolute path (P)
`autoname`	Prefix for automation data file (P)
`enter`	Enter sample information for automation run (C)

autolist Set up and start chained acquisition (M)

Syntax `autolist(<options,>experiment1<,experiment2<,...>)`

Description Sets up parameters for chained experiments by executing the experiments given as arguments and then starting a chained acquisition. Note that the macro au is executed as part of autolist and should not be included in the arguments to autolist.

Arguments options is one or more of the following keywords:

- 'auto' is a keyword to add 'wait' to the au call (e.g, au('wait','next')).
- 'start' is a keyword to make the first experiment in the list as one that needs to be acquired rather than processed.

experiment1,experiment2,... are experiments written as strings (e.g., 'dept' or 'c13'). experiment1 is the current experiment and, when it finishes, the macro procplot is called to process the data. If experiment2 is listed, that experiment is executed and then the macro au('next') is performed. For subsequent experiments, the text, solvent and temp are used from the preceding experiment. Also, the wexp parameter is reset to 'autolist' with the first experiment removed.

Examples
```
autolist('h1','c13','dept')
autolist('h1','hcosy')
```

See also *NMR Spectroscopy User Guide*

Related

auto_au	Controlling macro for automation (M)
au	Submit experiment to acquisition and process data (M)
hc	Automated proton and carbon acquisition (M)
hcapt	Automated proton, carbon, and APT acquisition (M)
hccorr	Automated proton, carbon, and HETCOR acquisition (M)
hcosy	Automated proton and COSY acquisition (M)
procplot	Automatically process FIDs (M)
solvent	Lock solvent (P)
temp	Sample temperature (P)
wexp	When experiment completes (P)

automerge Merges overniteQ with daytimeQ

Description This option is useful for sorting the short runs from longruns and merging the longruns at the back of short runs before doing autogo on an existing enter file. Alterntively, this macro can be used in a cron job to merge overniteQ with daytimeQ in a current automation run at a specified time. An optional 2nd argument will suppress autosa/autora. Optional argument allows one to merge overniteQ with daytimeQ for an enter file - i.e., non walkup mode.

Arguments add Arguments

Automkdir Creates Data Directory from Template

See also This macro is executed by automation at runtime to create the directory path specified in the Preferences/Templates panel.

autoname Create path for data storage (C)

Applicability Automation

Syntax

```
autoname:$path
autoname(name_template):$path
autoname(name_template,sample_info_file):$path
autoname (name_template,sample_info_file,
<'keepspaces'|'replacespaces'>):$path
autoname(name_template,sample_info_file,
    <,'excluded_suffixes'<,'keepspaces'|'replace
    spaces'>):$path
```

```
Svfname:$path
Svfname(name_template):$path
Svfname(name_template, suffix):$path
Svfname(name_template, suffix,
    'excluded_suffixes'):$path Svfname(name_template,
suffix, 'excluded_suffixes',
    'keepspaces'|'replacespaces'):$path
chkname(name_template, 'characters', 'par or tmpl or
    str','replacechar'):$s1,$s2,$par,$req
chkname('fileChars', 'characters')
```

Description The `autoname` command determines the path for data storage during an automation run and uses the value of a naming template (the `autoname` parameter by default) and the contents of a sample info file (default is `sampleinfo` in the current experiment) to determine this path. The path name is stored in the return argument or displayed on line 3 if no return argument is present.

The name is prefaced with using the value of the parameter `autodir` or `userdir+'/data/'` if `autodir` is equal to ''.

The default `excluded_suffix` is`.fid`.

Arguments No argument provided. The command uses the default `autoname` parameter and `sampleinfo` in the current experiment directory for the path to the sample info file. If the `autoname` parameter does not exist or is set to '', the default template is`%SAMPLE#:%%PEAK#:%`.

`name_template` (no quotes) is string that contains keywords separated by substitution specifiers to represent the data storage path. Substitution specifiers in this template are either a percent sign (%) or a dollar sign ($). The keywords are obtained from the `sample_info_file` file, if it exists, using % substitution specifiers or VNMR parameters using $ substitution specifiers.

A template is passed directly using: `autoname('$owner$/$sample$'):$path`.

Percent sign (%) substitution specifier is used with the autoname command to scan the `sample_info_file` for the text specific by keyword between the first percent sign in the template string and the next percent sign. The text specified by the keyword between the % substitution specifiers is passed to `$path`.

The following percent substitutions (% keywords) for time and date are obtained from the system clock, not from the sample info file:

Keyword	*Format*	*Description*
`%DATE%`	`YYYYMMDD`	4 digit year 2 digit month 2 digit day
`%TIME%`	`HHMMSS`	2 digit each for hours, minutes, and seconds
`%YR%`	`YYYY`	4-digit year
`%YR2%`	`YY`	2-digit year
`%MO%`	`MM`	2-digit month
`%DAY%`	`DD`	2-digit day
`%HR%`	`HH`	2-digit hour
`%MIN%`	`MM`	2-digit month
`%SEC%`	`SS`	2-digit second

The following are some of the percent substitutions (% keywords) are obtained from the second argument, `sample_info_file`.

Keyword	*Description*
%USER%	user name
%MACRO%	macro name
%SAMPLE%	sample name
%SOLVENT%	solvent name

String parameters cannot not contain any of the following characters: ' ', '!', '"', '$', '&', '\', '', '(', ')', '*', ';', '<', '>', '?', '\\', '[', ']', '^', '`', '{', '}', '|', ',', '\0'

Version number is specified by %Rn% where n is an integer from 0 to 9 (default 2), as follows:

n=	*Description*
0	no revision digits are appended (all names must be uniquely constructed without these revision digits).
1 to **9**	revision number is padded with leading zeroes to form an n-digit number. If more places are needed than specified, more zeroes are used.
>9 (more than one digit)	Rnn is still used as a search string in the sampleinfo file. %Rn% must be specified at the end of the `name_template` string. The revision digits are always appended except if %R0% is used.
no %Rn%	default of %R2% is used

Specify the starting number to be used when constructing the version number by appending a colon **:** and start number after Rn.

The default starting value is 1. A zero is not allowed.

Dollar sign ($) substitution specifiers works in manner analogous to the percent substitution specifier, except that the text between the dollar signs is interpreted as the name of a VNMR parameter. The value of this parameter is substituted for the substitution specifier.

Numeric parameters are represented as a string and truncated to an integer value. The template, `pw=$pw$usec`, with vnmr parameter `pw` having a value of 12.3 produces `pw=12usec01` which is appended to`.fid` and passed to `$path`. The 01 following usec is added by the %R2% default setting.

`sample_info_file` (no quotes) is the name of a text file to read for the % substitutions passed to autoname. The file must exist.

Using the keyword `'replacespaces'` uses underscores (_) in place of spaces `' '` in the resulting path name or the keyword `'keepspaces'` retains spaces in the resulting path name.

The keyword, `'keepspaces' or 'replacespaces'` is an optional argument (includes quotes). The argument is accepted as the third or fourth argument.

Solaris and Linux operating systems default to `replacespaces.`

A comma separated list of excluded suffixes the new path name will not use or match is specified if the third keyword is not `'keepspaces'` or `'replacespaces'`.

Examples Using a $ substitution specifier:

`autoname(pw=$pw$usec):$path`

A $ substitution specifier, `pw=$pw$usec`, is the `name_template` and a relative path. The vnmr parameter, `pw`, has a value of 12.3 and the resulting filename is: `pw=12usec01.fid`. The path name is prefaced with the value of the parameter `autodir` if the name template generates a relative pathname.

Examples Using $ substitution specifiers and a comma separated list of suffixes:

`autoname('$seqfil$_$tn$_','/vnmr/conpar','.img'):$path`

The $ substitution specifier is; `$seqfil$_$tn$_` the dummy info filename is; `'/vnmr/conpar',` and the comma separated list of excluded suffixes is `.img`. The path name is prefaced with `seqfil_tn_index`. Each time a file is written to the directory the command changes the index by one (see %Rn% above). The suffix is both `.fid` and `.img`. The file is named `gems_H1_03.img` if target directory contains `gems_H1_01.fid` and `gems_H1_02.img`.

See also *NMR Spectroscopy User Guide, VnmrJ User Programming, VnmrJ Walkup*

Related		
	`autoname`	Temple determining the path where is data stored (P)
	`Svfname`	Determines the name used to store data (C)
	`svfname`	Specifies the filename template (P)

autoname Prefix for automation data file (P)

Applicability Automation

Description The `autoname` temple determines the resulting path where the data is stored for an entry in the automation run and uses the contents of a sample info file (the name by default is "sampleinfo" in the current experiment) to determine this path. The path name is stored in the return argument and displayed on line 3 if no return argument is present.

See also *NMR Spectroscopy User Guide, VnmrJ User Programming, VnmrJ Walkup*

Related `autoname` Determines path for data storage during an automation run (C).

autora Resume suspended automation run (C)

Applicability Systems with an automatic sample changer.

Syntax `autora`

Description Resumes a previously suspended automation run. No matter what caused the interruption (including `autosa`, power failure, or system boot-up), the system examines the condition of the automation file and resumes acquisition for all experiments that have not finished. If `autora` is executed while an automation run is in progress, it has no effect.

See also *NMR Spectroscopy User Guide*

Related `autosa` Suspend current automation run (C)

autosa Suspend current automation run (C)

Applicability Systems with an automatic sample changer.

Syntax `autosa`

Description Suspends the automation mode at the conclusion of the current experiment and changes the system to the manual mode. The currently running experiment is not interrupted.

See also *NMR Spectroscopy User Guide*

Related `autora` Resume suspended automation run (C)

autoscale — Resume autoscaling after limits set by scalelimits macro (M)

Syntax `autoscale`

Description Returns to autoscaling in which the scale limits are determined by the `expl` command such that all the data in the `expl` input file is displayed.

See also *NMR Spectroscopy User Guide*

Related

`expl`	Display exponential or polynomial curves (C)
`scalelimits`	Set limits for scales in regression (M)

autostack — Automatic stacking for processing and plotting arrays (M)

Syntax `autostack`

Description When processing and plotting arrayed 1D spectra, VnmrJ automatically determines whether the stacking mode is horizontal, vertical or diagonal from the number of traces and the number of lines in the spectrum. If this automatic function is not desirable (or makes an undesirable decision), it can be overridden by placing the `stack` macro in the experiment startup macro or by calling `stack` before processing (or reprocessing) a spectrum. `autostack` switches back to automatic determination of the stack mode by destroying the `stackmode` parameter.

See also *NMR Spectroscopy User Guide*

Related

`procarray`	Process arrayed 1D spectra (M)
`plarray`	Plot arrayed 1D spectra (M)
`stack`	Fix stacking mode for processing / plotting arrayed spectra (M)
`stackmode`	Stacking control for processing (P)

autotest — Open Auto Test Window (C)

Syntax `autotest`

Description Opens the Auto Test window.

See also *AutoTest Software* manual.

autotime — Displays approximate time for automation (M)

Syntax `autotime(<automation directory>)`

Description Displays approximate time for each experiment and for each location in an automation run. If no argument is given, time is calculated for the current automation run (enterQ).

See also *NMR Spectroscopy User Guide*

Related explist Display approximate time for current experiment chain (M)

av Set abs. value mode in directly detected dimension (C)

Syntax av

Description Selects the absolute-value spectra display mode by setting the parameter dmg to the string value 'av'. In the *absolute-value display mode*, each real point in the displayed spectrum is calculated as the square root of the sum of the squares of the real and imaginary points comprising each respective complex data point. All information, including noise, is always positive, and the relationship between signal and noise is linear.

For multidimensional data, av has no effect on data prior to the second Fourier transform. If pmode='full', av acts in concert with commands ph1, av1, or pwr1 to yield the resultant contour display for the 2D data.

See also *NMR Spectroscopy User Guide*

Related

av1	Set abs. value mode in 1st indirectly detected dimension (C)
av2	Set abs. value mode in 2nd indirectly detected dimension (C)
dmg	Display mode in directly detected dimension (C)
dmgf	Absolute-value display of FID data or spectrum in acqi (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
pa	Set phase angle mode in directly detected dimension (C)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
ph	Set phased mode in directly detected dimension (C)
ph1	Set phased mode in 1st indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr1	Set power mode in 1st indirectly detected dimension (C)
wft	Weigh and Fourier transform 1D data (C)
wft1d	Weigh and Fourier transform of 2D data (C)
wft2d	Weigh and Fourier transform 2D data (C)

av1 Set abs. value mode in 1st indirectly detected dimension (C)

Syntax av1

Description Selects the absolute-value spectra display mode along the first indirectly detected dimension by setting the parameter dmg1 to the value 'av1'. If the parameter dmg1 does not exist, av1 creates it and set it to 'av1'.

In the *absolute-value display mode*, each real point in the displayed trace is calculated as the square root of the sum of the squares of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the real-real and imaginary-real points from each respective hypercomplex data point are used in the summation. In this mode, all information, including noise, is always positive; and the relationship between signal and noise is linear.

The av1 command is only needed if mixed-mode display is desired. If the parameter dmg1 does not exist or is set to the null string, the display mode along the first indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter dmg). For the contour display of multidimensional data, the result of av1 is the same as for traces provided that pmode='partial' or pmode='' (two single quotes with no space between).

See also *NMR Spectroscopy User Guide*

Related

av	Set abs. value mode in directly detected dimension (C)
dmg1	Data display mode in 1st indirectly detected dimension (P)

av2 Set abs. value mode in 2nd indirectly detected dimension (C)

Syntax av2

Description Selects absolute-value spectra display mode for the second indirectly detected dimension by setting the parameter dmg2 to the value 'av2'. If dmg2 does not exist or is set to the null string, av2 creates dmg2 and set it equal to 'av2'.

In the *absolute-value display mode*, all information, including noise, is positive; and the relationship between signal and noise is linear. Each real point in the displayed trace is calculated as the square root of the sum of the squares of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the real-real and imaginary-real points from each respective hypercomplex data point are used in the summation.

The av2 command is only needed if mixed-mode display is desired. If the parameter dmg2 does not exist or is set to the null string, the display mode along the second indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter dmg). For the contour display of multidimensional data, the result of av2 is the same as for traces provided that pmode='partial' or pmode='' (two single quotes with no space between).

See also *NMR Spectroscopy User Guide*

Related
av Set abs. value mode in directly detected dimension (C)
dmg2 Data display mode in 2nd indirectly detected dimension (P)

averag Calculate average and standard deviation of input (C)

Syntax `averag(number1,number2,...):average,sd, number_arguments,sum_numbers,sum_squares`

Description Finds average, standard deviation, and other characteristics of a set of numbers.

Arguments `number1,number2,...` is a finite set of numbers.

`average` is the average of the numbers.

`sd` is the standard deviation of the numbers.

`number_arguments` is the number of `number1,number2,...` arguments.

`sum_numbers` is the sum of the numbers

`sum_squares` is the sum of squares of the numbers.

Examples `averag(3.4,4.3,3.5,5.4):r1,r2`

See also *VnmrJ User Programming*

awc Additive weighting const. in directly detected dimension (P)

Description Adds the current value of awc to each value of the weighting function along the directly detected dimension. This dimension is often referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, and so forth. awc is applied *after* the sinebell and exponential function, but *before* the Gaussian function. This allows using gf as a Gaussian apodization even when awc is non-zero. Typical value of awc is *'n'*.

See also *NMR Spectroscopy User Guide*

Related
awc1 Additive weighting const. in 1st indirectly detected dimension (P)
awc2 Additive weighting const. in 2nd indirectly detected dim. (P)
gf Gaussian function in directly detected dimension (P)

awc1 Additive weighting const. in 1st indirectly detected dimension (P)

Description Adds the current value of awc1 to each value of the weighting function along the first indirectly detected dimension This dimension is often referred to as the f_1 dimension of a multidimensional data set. awc1 is analogous to the parameter awc. The “conventional” parameters (lb, gf, etc.) operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

See also *NMR Spectroscopy User Guide*

Related awc Additive weighting const. in directly detected dimension (P)

awc2 Additive weighting const. in 2nd indirectly detected dimension (P)

Description Adds the current value of awc2 to each value of the weighting function along the second indirectly detected dimension This dimension is often referred to as the f_2 dimension of a multidimensional data set. awc2 is analogous to the parameter awc. The value of awc2 can be set with wti on the 2D interferogram data.

See also *NMR Spectroscopy User Guide*

Related
awc Additive weighting const. in directly detected dimension (P)
wti Interactive weighting (C)

axis Provide axis labels and scaling factors (C)

Syntax
```
axis('fn'|'fn1'|'fn2')
   <:$axis_label,$freq_scaling,$scaling_factor>
```

Description Displays or returns values of the axis labels and scaling factors to the calling macro. See the macro rl for an example of using this command.

Arguments 'fn'|'fn1'|'fn2' is the Fourier number parameter for the axis of interest.

$axis_label is the axis label (e.g., ppm, kHz, cm, or ppm(sc)).

$freq_scaling is the divisor needed to convert from units of Hz to the units defined by the axis parameter with any scaling. axis uses the current value of the axis parameter for that dimension and also checks for axis scaling using the corresponding scalesw, scalesw1, or scalesw2 parameter.

$scaling_factor is a second scaling factor, determined solely by the scalesw type of parameter. This last scaling factor is independent of the value of the axis parameter.

Examples `axis('fn')`
`axis('fn1'):$lab,$fr,$scl`

See also *VnmrJ User Programming*

Related
axis	Axis label for displays and plots (P)
rl	Set reference line (M)
scalesw	Scale spectral width in directly detected dimension (P)
scalesw1	Scale spectral width in 1st indirectly detected dimension (P)
scalesw2	Scale spectral width in 2nd indirectly detected dimension (P)

axis — Axis label for displays and plots (P)

Applicability Certain arguments work only if system has the proper hardware.

Description Specifies the units for the axis display and plot.

For 1D experiments, `axis` uses a single letter that includes `'h'` for Hz, `'p'` for ppm, and `'k'` for kHz (e.g., `axis='h'`).

For 2D experiments, `axis` uses two letters, with the first letter describing the detected spectral axis (f_2), and the second letter describing the indirectly detected axis (f_1). Thus `axis='ph'` is appropriate for a homonuclear 2D-J experiment, with a referenced ppm scale along the spectral axis and an axis in Hz (`'h'`) along the J-axis. `axis='pp'` is appropriate for COSY or NOESY experiments.

For 3D experiments, `axis` uses three letters with the first letter describing the detected spectral axis (f_3), the second letter describing the first indirectly detected axis (f_1), and the third letter specifying the second indirectly detected axis (f_2).

The special letter `d` is used to reference the indirectly detected axis to the parts per million of the decoupler channel, as appropriate for heteronuclear chemical shift correlation experiments, which would typically have `axis='pd'`. The letter `n` is used to suppress the axis display on one or both axes (e.g., `axis='nn'`, `axis='pn'`).

For systems with multiple decouplers, the characters `'1'`, `'2'`, and `'3'` can be used to reference an axis relative to the frequency of that decoupler. Setting `axis='p1'` is effectively the same as `axis='pd'`.

Values `'1'` sets the axis label for units of ppm relative to the first decoupler.
`'2'` sets the axis label for units of ppm relative to the second decoupler.
`'3'` sets the axis label for units of ppm relative to the third decoupler.

`'c'` sets the axis label for units of centimeters.
`'d'` sets the axis label for units of ppm relative to the first decoupler.
`'h'` sets the axis label for units of hertz.

`'k'` sets the axis label for units of kilohertz.
`'m'` sets the axis label for units of millimeters.
`'n'` sets no axis label display.

'p' sets the axis label for units of ppm relative to the observe transmitter.
'u' sets the axis label for units of micrometers.

See also *NMR Spectroscopy User Guide*

Related
axis	Provide axis labels and scaling factors (C)
axisf	Axis label for FID displays and plots (P)
dscale	Display scale below spectrum or FID (C)
pscale	Plot scale below spectrum or FID (C)

axisf Axis label for FID displays and plots (P)

Description Specifies the units for the FID axis display and plot. To create the FID display parameters axisf, dotflag, vpf, vpfi, crf, and deltaf (if the parameter set is older and lacks these parameters), enter addpar('fid').

Values 's' sets the axis label for units of seconds.

'm' sets the axis label for units of ms.

'u' sets the axis label for units of µs.

'n' sets no axis label display.

See also *NMR Spectroscopy User Guide*

Related
addpar	Add selected parameters to the current experiment (M)
axis	Axis label for displays and plots (P)
dscale	Display scale below spectrum or FID (C)
pscale	Plot scale below spectrum or FID (C)

B

bandinfo Shaped pulse information for calibration (M)

Applicability Information only useful on systems capable of shaped pulse generation.

Syntax `bandinfo<(shape,width<,ref_power>)>:duration,power`

Description Displays a table containing the duration and the predicted 90° pulse power setting for the pulse shape and bandwidth given by the arguments. No parameter settings are changed. The necessary data is contained in the `shapeinfo` file in the `shapelib` subdirectory.

Arguments If `bandinfo` is run without arguments, prompts operator for input

`shape` is the name of the shape. The default is system prompts for a name.

`width` is the bandwidth, in Hz, desired for the pulse.

ref_power is value of tpwr to which pw90 is set. The default is 55 dB.

duration is the duration, in µs, of the pulse.

power is the predicted 90° pulse power setting.

Examples
```
bandinfo
bandinfo('sinc',10):pw,tpwr
```

See also *User Programming*

Related

pulseinfo	Shaped pulse information for calibration (M)
pw90	90° pulse width (P)
tpwr	Observe transmitter power level with linear amplifiers (P)

banner — Display message with large characters (C)

Syntax `banner(message<,color>)`

Description Displays text as large-size characters on the graphics windows.

Arguments message is the text to be displayed. If the text includes a single quotation mark ('), it must be preceded by a backslash (\'). Multiline displays are available by inserting two backslashes (\\) between lines. Any undefined characters are displayed as a "bug" shape.

color is the color of text on a color display: 'red', 'yellow', 'green', 'cyan', 'blue', 'magenta', and 'white'. The default is 'yellow'.

Examples
```
banner('banner sample')
banner('Don\'t Touch','blue')
```

See also *User Programming*

bc — 1D and 2D baseline correction (C)

Description Makes 1D or 2D baseline correction using a spline or a second to twentieth order polynomial fitting of predefined baseline regions. bc defines every other integral (those integrals that disappear when intmod='partial') as baseline and attempts to correct these points to zero.

1D baseline correction

Syntax `bc<(n|'unbc'<,nsubregion<,minpoints<,minregion>>>)>`

Description Performs a 1D baseline correction. The nonintegrated parts of the spectrum (i.e., every odd region between integral reset points, or the integral gaps with intmod='partial') are divided into baseline subregions. The number of baseline subregions in each area are adjusted as possible, so that the subregions are more or less equal in size. Finally, the "center of gravity" (midpoint in *x and* average of the *y* values in the region) for each of the subregions is calculated.

Arguments n is an integer from 1 to 20 for the baseline correction step. A polynomial of the (n-1)th order is calculated “through” the “baseline points” using the Chebychev least-squares fitting algorithm, and that polynomial function is subtracted from the spectrum. The coefficients of the polynomial are written into the file cureexp+'/bc.out'. The default is 1(a spline fit).

'unbc' is a keyword to make bc read in the coefficients from the file written by the previous bc operation and reverse that operation. This option is only functional for polynomials with two or more coefficients performing baseline correction operations on 1D spectra or individual 2D traces (i.e., baseline corrections cannot be undone with the default spline correction).

nsubregion defines the number of subregions (minimum 3, maximum 400). By default, the total number of subregions is 20 (if fn<2048), 40 (if fn=2048 or fn=4096), or 80 (if fn>4096).

minpoints sets the minimum number of data points required in an integral gap for bc to regard it as baseline. Use this to exclude small, nonintegrated areas between close signals. The default is fn/1000 (but at least 3).

minregion defines the minimum number of subregions assigned to each baseline area. The default is 1.

Examples bc
bc(3)
bc('unbc')
bc(1,200,8,2)gives a spline correction using 200 baseline subregions, a gap of 8 data points between two (even) integral regions is regarded as baseline, and each baseline area is split into at least two subregions.

2D baseline correction

Syntax bc(trace_direction<,num_coeff><,trace_start>
<,trace_end>)

Description 2D baseline correction can be performed on three types of 2D data:

- f2 spectra (trace_direction='f2') after the first half of a 2D FT (wft1da).
- f2 traces (trace_direction='f2') after a full 2D FT (wft2da).
- f1 traces (trace_direction='f1') after a full 2D FT (wft2da).

Arguments trace_direction specifies the direction, 'f1' or 'f2', along which the 2D baseline correction is to take place.

num_coeff is the number of coefficients, from 1 to 20, used in the fitting procedure. The default value is 1, which gives a spline fit. A value of 2 gives a linear baseline fit ($a + bx$), a value of 3 gives a quadratic fit ($a + bx + cx^2$), etc. The maximum value (20) gives a 19th-order polynomial fit with 20 coefficients.

trace_start is the trace number for the spectrum on which the 2D baseline correction is to start. It must lie within the appropriate range or an error results.

`trace_end` is the trace number for the spectrum on which the 2D baseline correction is to end. It must lie within the appropriate range or an error results.

Examples
```
bc('f1')
bc('f2',3)
bc('f2',3,10,60)
```

See also *NMR Spectroscopy User Guide*

Related

dc	Calculate spectral drift correction (C)
fn	Fourier number in directly detected dimension (P)
intmod	Integral display mode (P)
trace	Mode for 2D data display (P)
wft1da	Weight and Fourier transform phase-sensitive data (M)
wft2da	Weight and Fourier transform phase-sensitive data (M)

beepoff Turn beeper off (C)

Description Turns off the beeper sound so that the system does not use sound to warn the user when errors occur. The default is the beeper is turned on.

See also *User Programming*

Related beepon Turn beeper on (C)

beepon Turn beeper on (C)

Syntax `beepon`

Description Turns on the beeper sound so that the user hears a sound when errors occur. The default is the beeper is turned on.

See also *User Programming*

Related beepoff Turn beeper off (C)

bigendian Determine system byte order (C)

Syntax `bigendian:$type`

Description The bigendian

Description command determines the system byte order for storing numbers. One architecture is Big Endian, used by Sun computers with the “Sparc” CPU’S. The other architecture is Little Endian, used by most PCs.

Return values to argument `$type`:

1 if it is a “Big Endian” system.

0 if it is a “Little Endian” system.

This command should rarely be used. Its only current use is when imaging .fdf files are created. The .fdf file headers can specify whether the data is stored as big or little endian.

binom — Set up parameters for BINOM pulse sequence (M)

Description: Sets up a binomial water suppression pulse sequence.

See also: *NMR Spectroscopy User Guide*

bioref — Bio-NMR Referencing (P)

Applicability: All

Syntax: `bioref='<y or n>'`

Description: Flag, global or local, for Bio-NMR Referencing. Setting the flag (bioref='y') sets the system to bio-NMR referencing (based on nuctables/nuctabrefBio) rather than standard IUPAC / organic chemistry referencing (based on nuctables/nuctabref). Bio-NMR referencing uses DSS for nuclei such as ^{13}C and liquid NH_3 for ^{15}N.

Creating bioref as a local parameter (create('bioref','flag') creates a local flag) permits its use for a specific case. The parameter can be created as a local parameter and saved with a standard parameter set (stdpar/N15) to enable bio-NMR referencing for a specific nucleus. The local value of the parameter takes precedence over the global parameter.

create('bioref','flag','global') — creates a global flag.
setenumeral('bioref',2,'y','n','global') — sets the possible values of a string parameter in a parameter tree.

Examples: bioref='y' sets referencing to use nuctables/nuctabrefBio

Related		
	create	Create new parameter in a parameter tree (C)

bootup — Macro executed automatically (M)

Syntax: `bootup<(foreground)>`

Description: Executed automatically when VnmrJ is started up. The bootup macro displays a message, looks for a macro login in the user’s local maclib directory and executes it (if found), starts Acqstat and acqi (acqi is not run if system is configured as a workstation), and then starts the menu system. This set of actions can be modified on a per user basis by constructing custom `bootup` or `login` macros in the user's `maclib` directory. A custom `login` macro is preferred because all custom `bootup` macros are overridden whenever a new VnmrJ release is installed.

Arguments `foreground` is 0 if VnmrJ is being run in the foreground or nonzero if being run in the background. This argument is passed to the `login` macro.

See also *User Programming*

Related `acqi` Interactive acquisition display process (C)
`Acqstat` Bring up the acquisition status display (U)

box Draw a box on a plotter or graphics display (C)

Syntax `box(<'keywords',>x1mm,x2mm,y1mm,y2mm <,'nolimit'>)<:r1,r2>`

Description Draws a box on a plotter or a graphics display.

Arguments `'keywords'` identifies the output device (`'graphics'|'plotter'`), drawing mode (`'xor'|'normal'`), and drawing capability (`'newovly'|'ovly'|'ovlyC'`).

- `'graphics'|'plotter'` is a keyword for the output device. The default is `'plotter'`. The output selected is passed to subsequent `pen`, `move`, or `draw` commands and remains active until a different output is specified.
- `'xor'`,`'normal'` is a keyword for the drawing mode when using the `'graphics'` output device. The default is `'normal'`. In the `'xor'` mode, if a line is drawn such that one or more points of the line are in common with a previous `'xor'` line, the common points are erased. In the `normal` mode, the common points remain. The mode selected is passed to subsequent `pen`, `move`, and `draw` commands and remains active until a different mode is specified.
- `'newovly'`, `'ovly'` and `'ovlyC'` are keywords that specify an interactive drawing capability that is slightly slower than the `'xor'` mode but more consistent in color. `'newovly'` clears any previous draws, boxes, and writes made with the `'ovly'` modes and draws the figure. `'ovly'` draws without clearing so that multi-segment figures can be created. `'ovlyC'` clears without drawing.

`x1mm` is the left edge of the box, `x2mm` is the right edge, `y1mm` is the bottom, and `y2mm` is the top. The location of the edges are given in plotter units (mm on most plots) and are scaled in mm for the graphics display. (If units are in Hz or ppm, you can use the `hztomm` command to convert units.)

`'nolimit'` allows the box to extend outside the limits determined by the parameters `sc`, `wc`, `sc2`, and `wc2`.

`r1`, `r2` return the location of the upper left corner of the box.

Examples	`box('plotter',20,100,40,150)` `box(25,105,45,155,'nolimit'):r1,r2`	
See also	*NMR Spectroscopy User Guide*	
Related	gin	Return current mouse position and button values (C)
	hztomm	Convert positions from Hz or ppm to plotter units (C)
	sc	Start of chart (P)
	sc2	Start of chart in second direction (P)
	wc	Width of chart (P)
	wc2	Width of chart in second direction (P)
	wcmax	Maximum width of chart (P)

boxes — Draw boxes selected by the mark command (M)

Syntax `boxes<('graphics'|'plotter')>`

Description Draws boxes on a plotter or a graphics display with the location of the edges given in Hz. The data to make the boxes is stored in the `mark2d.out` file produced by the mark command. If there is no data in `mark2d.out`, a box is drawn from the current cursor positions. The `boxes` command also numbers the boxes above the upper left corner.

Arguments `'graphics'|'plotter'` is a keyword to send output to the graphics display or to the plotter, respectively. The default is `'graphics'`.

Examples
```
boxes
boxes('plotter')
```

See also *NMR Spectroscopy User Guide*

Related

mark	Determine intensity of spectrum at a point (C)

bpa — Plot boxed parameters (M)

Syntax `bpa:$sc2_minimum`

Description Plots a box around the entire chart (assuming blank paper) and then plots “chemist-style” parameters in boxes along the lower edge of the chart. `bpa` is the same as ppa, but with a different layout. Both ppa and `bpa` behave somewhat naively if the pulse sequence is more complex, but they were designed primarily for chemists, not for spectroscopists.

Arguments `sc2_minimum` returns the minimum value for sc2 to plot a scale properly. To use the command pir, vp has to be set to a non-zero value.

See also *NMR Spectroscopy User Guide*

Related

apa	Plot parameters automatically (M)
pap	Plot out “all” parameters (C)

pir	Plot integral amplitudes below spectrum (C)
ppa	Plot a parameter list in “English” (M)
sc2	Start of chart in second direction (P)
vp	Vertical position of spectrum (P)

br24 Set up parameters for BR24 pulse sequence (M)

Applicability Systems with solids hardware.

Description Converts a FLIPFLOP, MREV8, or S2PUL parameter set into a BR24 solids line-narrowing multiple-pulse sequence.

See also *User Guide: Solid-State NMR*

Related

cylbr24	Set up parameters for cycled BR24 pulse sequence (M)
flipflop	Set up parameters for FLIPFLOP pulse sequence (M)
mrev8	Set up parameters for MREV8 pulse sequence (M)
s2pul	Set up standard two-pulse sequence (M)

bs Block size (P)

Description Directs the acquisition computer, as data are acquired, to periodically store a block of data on the disk, from where it can be read by the host computer.

CAUTION If bs='n', block size storage is disabled and data are stored on disk only at the end of the experiment. If the experiment is aborted prior to termination, data will be lost.

Values 1 to 32767 transients, 'n'

See also *NMR Spectroscopy User Guide*

Related

wbs	Specify action when *bs* transients accumulate (C)
wbs	When block size (P)

C

c13	Automated carbon acquisition (M)
c13p	Process 1D carbon spectra (M)
calcdim	Calculate dimension of experiment (C)
calfa	Recalculate alfa so that first-order phase is zero (M)
calibflag	Correct systematic errors in DOSY experiments (P)
calibrate	Start a dialog for autocalibration routines (M)
callacq	Utility macro to call Acq command (M)
capt	Automated carbon and APT acquisition (M)
Carbon	Set up parameters for 13C experiment (M)
cat	Display one or more text files in text window (C)
cattn	Coarse attenuator type (P)
cd	Change working directory (C)
cdc	Cancel drift correction (C)
cdept	Automated carbon and DEPT acquisition (M)
cdump	Prints the current graphics screen (M)
celem	Completed FID elements (P)
center	Set display limits for center of screen (C)
centerprobe	Calculates probe position relative to the ISO-Center.
centersw	Move cursor to center of spectrum (M)
centersw1	Move cursor to center of spectrum in 1st indirect dimension (M)
centersw2	Move cursor to center of spectrum in 2nd indirect dimension (M)
cexp	Create an experiment (M)
cf	Current FID (P)

chkname	Parse the template and return substituted strings and lists of parameters defined by the template
cfpmult	Calculate first-point multiplier for 2D experiments (M)
change	Submit a change sample experiment to acquisition (M)
checkstring	Find and replace unwanted characters (C)
chiliConf	Control flag set by ecc_on and ecc_off (P)
Cigar2j3j	Convert the parameter to a CIGAR2j3j experiment (M)
ckresloc	Macro to Reserve Specific Locations
ckstring	Utility to Check String Variables for Illeagal Characters
cla	Clear all line assignments (M)
cla	Calculated transition number (P)
clamp	Calculated transition amplitude (P)
cleanexp	Remove old files and directories from an experiment (M)
clear	Clear a window (C)
cleardosy	Delete temporarily saved data in current sub experiment (M)
clfreq	Calculated transition frequency (P)
clindex	Index of experimental frequency of a transition (P)
clradd	Clear add/subtract experiment (C)
color	Select plotting colors from a graphical interface (M)
cmdlineOK	Determine if an operator has a command line
coldprobe	Tells system a coldprobe is present.
combiplate	View a color map for visual analysis of VAST microtiter plate (U)
combishow	Display regions (red, green, and blue) in CombiPlate window (M)
compressfid	Compress double-precision FID data (M,U)
config	Display current configuration and possibly change it (M)
confirm	Confirm message using the mouse (C)
Console	System console type (P)
contact_time	MAS cross-polarization spin-lock contact time (M)
continflag	The command ddif creates a CONTIN display if continflag='y'.
continprepare	Called by the macro dosy to prepare the input file for the CONTIN programme.

continread	Called by the macro dosy to take the output of the CONTIN programme and create an input file for ddif.
continueMovie	Continue movie in either forward or backward direction (C)
convert	Convert data set from a VXR-style system (M,U)
convertbru	Convert Bruker data (M,U)
copy	Copy a file (C)
cos	Find cosine value of an angle (C)
Cosy	Convert the parameter to a COSY experiment (M)
cosyps	Set up parameters for phase-sensitive COSY pulse sequence (M)
cp	Copy a file (C)
cp	Cycle phase (P)
cpdone	Macro called upon study completion (M)
cpgo	Macro called upon study completion (M)
cpmgt2	Set up parameters for CPMGT2 pulse sequence (M)
cpos_cvt	Convert data set from a VXR-style system (M,U)
cptmp	Copy experiment data into experiment subfile (M)
cptmpltdefaults	Defaults for Save Data Template
cpx	Create pbox shape file (M)
cqexp	Load experiment from protocol (M)
cqfindz0	Run an experiment to find the value of z0 (M)
cqgmap	Perform gradient shimming utility functions (M)
cqinit	Initialize liquids study queue (M)
cqpars	Create study queue parameters for liquids (M)
cqplot	Macro to perform generic 2D plot (M)
cqprotocol	Macro to create protocols (M)
cqreset	Reset study queue parameters (M)
cqsavestudy	Macro to save study queue parameters (M
cqwtmenu	Macro to set weighting functions from a panel (M)
cr	Cursor position in directly detected dimension (P)
cr1	Cursor position in 1st indirectly detected dimension (P)
cr2	Cursor position in 2nd indirectly detected dimension (P)
crcom	Create user macro without using text editor (M)

create	Create new parameter in a parameter tree (C)
create(P)	Parameter used for RF transmitter board temperature compensation
createqcomp	Create qcomp parameter (M)
crf	Current time-domain cursor position (P)
crl	Clear reference line in directly detected dimension (M)
crl1	Clear reference line in 1st indirectly detected dimension (M)
crl2	Clear reference line in 2nd indirectly detected dimension (M)
crmode	Current state of the cursors in df, ds, or dconi programs (P)
crof2	Recalculate rof2 so that lp = 0 (M)
cryo_noisetest	Run Cold Probe conditioning experiments (M)
cryoclient	Start the CryoBay Monitor program (M, U)
csv2cpQ	Imports CSV Data (M)
ct	Completed transients (P)
ctext	Clear the text of the current experiment (C)
curexp	Current experiment directory (P)
curscan	Scan currently in progress (P)
curwin	Current window (P)
cutoff	Data truncation limit (P)
cyclenoe	Set up parameters for CYCLENOE pulse sequence (M)
cylbr24	Set up parameters for cycled BR24 pulse sequence (M)
cylmrev	Set up parameters for cycled MREV8 pulse sequence (M)
cz	Clear integral reset points (C)

c13 Automated carbon acquisition (M)

Syntax `c13<(solvent)>`

Description Prepares parameters for automatically acquiring a standard ^{13}C spectrum. The parameter wexp is set to 'procplot' for standard processing. If c13 is used as the command for automation via the enter command, the au is supplied automatically and should not be entered on the MACRO line of the enter program. However, it is possible to customize the standard c13 macro on the MACRO line by following it with additional commands and parameters. For example, `c13 nt=1` uses the standard `c13` setup but with only one transient.

Arguments `solvent` is the name of the solvent. In automation mode the solvent is supplied by the `enter` program. The default is `'CDCl3'`.

Examples
```
c13
c13('DMSO')
```

See also *NMR Spectroscopy User Guide*

Related

`au`	Submit experiment to acquisition and process data (M)
`c13p`	Process of 1D carbon spectra (M)
`enter`	Enter sample information for automation run (C)
`proc1d`	Processing macro for simple (non-arrayed) 1D spectra (M)
`procplot`	Automatically process FIDs (M)
`wexp`	When experiment completes (P)

c13p Process 1D carbon spectra (M)

Syntax `c13p`

Description Processes non-arrayed 1D carbon spectra using a set of standard macros. `c13p` is called by the `proc1d` macro, but can also be used directly. Fully automatic processing (up to a point where a spectrum could be plotted) is provided: Fourier transformation (using pre-set weighting functions), automatic phasing (`aphx` macro), automatic integration (`integrate` macro if required only), vertical scale adjustment (`vsadjc` macro), avoiding excessive noise (`noislm` macro), threshold adjustment (`thadj` macro), and referencing to the TMS signal if present (`setref` macro then `tmsref` macro).

See also *NMR Spectroscopy User Guide*

Related

`aphx`	Perform optimized automatic phasing (M)
`c13`	Automated carbon acquisition (M)
`integrate`	Automatically integrate 1D spectrum (M)
`noislm`	Limit noise in spectrum (M)
`proc1d`	Processing macro for simple (non-arrayed) 1D spectra (M)
`setref`	Set frequency referencing for proton spectra (M)
`thadj`	Adjust threshold (M)
`tmsref`	Reference spectrum to TMS line (M)
`vsadjc`	Adjust vertical scale for carbon spectra (M)

calcdim Calculate dimension of experiment (C)

Syntax `calcdim`

Description Calculates the dimension of an experiment and puts the result into the parameter `arraydim`. If an experiment is arrayed, `arraydim` is the product of the size of the arrays.

See also *NMR Spectroscopy User Guide*

Related `arraydim` Dimension of experiment (P)

calfa Recalculate alfa so that first-order phase is zero (M)

Syntax `calfa`

Description Based upon the current `alfa` and `lp` values, `calfa` calculates a new value for `alfa` so that the first-order phase parameter `lp` is rendered approximately 0. When digital filtering is active (`dsp='r'` or `dsp='i'`), `calfa` also adjusts `rof2` as well as `alfa`. For `calfa` to work properly, a trial spectrum must be obtained and phased to pure absorption. This spectrum provides `calfa` with the current `alfa` and `lp` values. `calfa` pertains to processing 2D data. Unless `lp` is approximately 0, `fpmult` will affect both the `dc` offset and the curvature of the spectrum.

See also *NMR Spectroscopy User Guide*

Related

`alfa`	Set `alfa` delay before acquisition (P)
`cfpmult`	Calculate first-point multiplier for 2D experiments (M)
`crof2`	Recalculate rof2 so that lp = 0 (M)
`dc`	Calculate spectral drift correction (C)
`dsp`	Type of DSP for data acquisition (P)
`fpmult`	First-point multiplier for `np` FID data (P)
`hoult`	Set parameters `alfa` and `rof2` according to Hoult (M)
`lp`	First-order phase in directly detected dimension (P)
`rof2`	Receiver gating time following pulse (P)

calibflag Correct systematic errors in DOSY experiments (P)

Syntax `calibflag`

Description Corrects systematic errors in DOSY experiments.

Values `'y'` corrects systematic deviations in DOSY analysis.

`'n'` omits gradient correction in DOSY analysis.

See also *NMR Spectroscopy User Guide*

Related `dosy` Process DOSY experiments (M)

calibrate Start a dialog for autocalibration routines (M)

Syntax `calibrate`

Description Starts a dialog for autocalibration routines.

callacq Utility macro to call Acq command (M)

Syntax `callacq(arg_string)`

Description Utility macro to construct a string to pass to psg via the `Acq()` command.This macro should be used only by users with advanced knowledge. A well-constructed argument string is required. The motivation for this macro is to make the '`go`' macro re-entrant, while still synchronizing with VnmrJ.

Arguments `arg_string` is a character string constructed from a macro.

Examples `callacq($callback)`

Related	
`go`	Submit experiment to acquisition (M)
`reqparcheck`	Flag which enables/disables required parameters (P)
`reqparclear`	Clears the parameters in required parameter list (M)
`reqparlist`	List of required parameters (P)
`reqpartest`	Tests whether required parameters are set (M)

capt Automated carbon and APT acquisition (M)

Syntax `capt<(solvent)>`

Description Prepares parameters for automatically acquiring a standard ^{13}C spectrum, followed by an APT experiment. In non-automation mode, the carbon and APT spectra are acquired in the experiment in which `capt` is entered. Following acquisition completes, the commands `rttmp('C13')` and `rttmp('apt')` can be used for further processing of the carbon and APT spectra, respectively.

Arguments `solvent` is name of the solvent used. In automation mode, the `enter` program supplies name. In non-automation mode, the default is `'cdcl3'`.

Syntax `capt au`
`capt('dmso')`

See also *NMR Spectroscopy User Guide*

Related	
`Apt`	Prepare parameters for APT experiment (M)
`c13`	Automated carbon acquisition (M)
`enter`	Enter sample information for automation run (C)
`rttmp`	Retrieve experiment subfile (M)

Carbon — Set up parameters for 13C experiment (M)

Description Set up parameters for ^{13}C experiment

cat — Display one or more text files in text window (C)

Syntax `cat(file1<,file2,...>)`

Description Displays the contents of one or more text files on the text window. It pauses after the window has filled and waits for the user to indicate whether it should display more or should terminate.

Arguments `file1,file2,...` are the names of the files to be displayed.

Examples `cat('/vnmr/manual/cat')`
`cat('/vnmr/manual/cat','/vnmr/manual/cattn')`

See also *NMR Spectroscopy User Guide*

cattn — Coarse attenuator type (P)

Applicability Systems with a coarse attenuator.

Description Identifies the type of coarse attenuator if this attenuator is present on the current rf channel. The value of `cattn` is set in the Spectrometer Configuration window (opened by entering `config`) using the label Coarse Attenuator.

Values 0 for no coarse attenuator, as in the case with class C amplifiers (Not Present choice in Spectrometer Configuration window).

79 for standard systems (79 dB choice in Spectrometer Configuration window).

127 for imaging attenuator (63.5 dB SIS choice in Spectrometer Configuration window).

63 for deuterium decoupler channel.

See also *VnmrJ Installation and Administration*

Related

`config`	Display current configuration and possibly change it (M)
`fattn`	Fine attenuator (P)
`tpwr`	Observe transmitter power level with linear amplifiers (P)

cd — Change working directory (C)

Syntax `cd<(directory)>`

Description Changes current working directory to another directory.

Arguments `directory` is the name of the directory that becomes the new current working directory. The change is made only if the directory name already exists and the user has permission to be in the directory. If no argument is included, `cd` changes the current working directory to the user's home directory.

Examples
```
cd
cd(userdir+'/exp1')
cd('/home/george/vnmrsys')
```

See also *NMR Spectroscopy User Guide*

Related		
	pwd	Display current working directory (C)

cdc Cancel drift correction (C)

Syntax `cdc`

Description Turns off the drift correction started by the dc command and resets the spectral drift correction parameters `lvl` (level) and `tlt` (tilt) to zero.

See also *NMR Spectroscopy User Guide*

Related		
	dc	Calculate spectral drift correction (C)
	dcg	Drift correction group (P)
	lvl	Zero-order baseline correction (P)
	tlt	First-order baseline correction (P)

cdept Automated carbon and DEPT acquisition (M)

Syntax `cdept<(solvent)>`

Description Prepares parameters for automatically acquiring a standard ^{13}C spectrum, followed by a DEPT experiment. In non-automation mode, the carbon and DEPT spectra are acquired in the experiment in which `cdept` was entered. Following the completion of the acquisition, the `rttmp('C13')` and `rttmp('dept')` commands can be used for further processing of the carbon and DEPT spectra, respectively.

Arguments `solvent` is name of the solvent used. In automation mode, the `enter` program supplies name. In non-automation mode, the default is `'cdcl3'`.

Examples
```
cdept au
cdept('DMSO')
```

See also *NMR Spectroscopy User Guide*

Related		
	adept	Automatic DEPT analysis and spectrum editing (C)
	c13	Automated carbon acquisition (M)
	dept	Prepare parameters for DEPT experiment (M)
	enter	Enter sample information for automation run (C)
	rttmp	Retrieve experiment subfile (M)

cdump Prints the current graphics screen (M)

Syntax `cdump('filename')`

Description `cdump` takes the current display and sends it to the current printer. If an optional `filename` is passed as an argument, the current display will be saved in the print subdirectory of the user's `vnmrsys` directory. This directory will be created if is does not already exist. If the `filename` passed to the `cdump` macro is an absolute pathname, i.e., it starts with a '/' character, that pathname will be used.

If the current display is saved as a file, the format of the file is specified by the `printformat` parameter. It can be set to the following values. as for PostScript formatted output.

`japed` for Joint Photographic Experts Group JFIF formatted output. `nag` for Portable Network Graphics formatted output.

celem Completed FID elements (P)

Description Indicates the current number of completed FIDs in an experiment. When `go` or `au` is entered, `celem` is set to 0. As each FID acquisition is completed, `celem` is updated to reflect this. This parameter is most useful in conjunction with `wbs`, `wnt`, `wexp`, and `werr` processing commands.

See also *NMR Spectroscopy User Guide*

Related

`arraydim`	Dimension of experiment (P)
`au`	Submit experiment to acquisition and process data (C)
`go`	Submit experiment to acquisition (C)
`ni`	Number of increments in 1st indirectly detected dimension (P)
`wbs`	Specify action when `bs` transients accumulate (C)
`werr`	Specify action when error occurs (C)
`wexp`	Specify action when experiment completes (C)
`wnt`	Specify action when `nt` transients accumulate (C)

center Set display limits for center of screen (C)

Description Sets parameters `sc` and `wc` (horizontal control) and parameters `sc2` and `wc2` (vertical control) to produce a display (and subsequent plot) in the center portion of the screen (and page). For 2D data, space is left for the scales.

See also *NMR Spectroscopy User Guide*

Related

`full`	Set display limits for a full screen (C)
`fullt`	Set display limits for full screen with room for traces (C)
`left`	Set display limits for left half of screen (C)

right	Set display limits for right half of screen (C)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

centerprobe Calculates probe position relative to the ISO-Center

Syntax `centerprobe`

Applicability VnmrJ 3.1

Description A macro that calculates the rf probe position relative to the ISO-center. Using the Z2 plot from the current gradient map this macro calculates how far out of center the probe is.

Arguments `centerprobe` will display the results on the graphics screen.

`centerprobe('plot')` will plot the results on the selected plotter.

centersw Move cursor to center of spectrum (M)

Description Sets cursor position parameter cr in the directly detected dimension for the center of the spectrum.

See also *NMR Spectroscopy User Guide*

Related

centersw1	Move cursor to center of spectrum in 1st indirect dimension (M)
centersw2	Move cursor to center of spectrum in 2nd indirect dimension (M)
cr	Cursor position in directly detected dimension (P)

centersw1 Move cursor to center of spectrum in 1st indirect dimension (M)

Description Sets cursor position parameter cr1 in the first indirectly detected dimension to the center of the spectrum.

See also *NMR Spectroscopy User Guide*

Related

centersw	Move cursor to center of spectrum (M)
cr1	Cursor position in 1st indirectly detected dimension (P)

centersw2 Move cursor to center of spectrum in 2nd indirect dimension (M)

Description Sets cursor position parameter cr2 in the second indirectly detected dimension to the center of the spectrum.

See also *NMR Spectroscopy User Guide*

Related		
	centersw	Move cursor to center of spectrum (M)
	cr2	Cursor position in 2nd indirectly detected dimension (P)

cexp Create an experiment (M)

Syntax `cexp(<experiment_dir,>experiment_number)`

Description Creates an experiment as a temporary workspace that can hold a complete 1D, 2D, or 3D data set. Up to 9999 experiments can be created. Experiment 5 is special because it is the add-subtract experiment. cexp creates the appropriate jexp macro so that the newly created experiment can be joined.

Arguments experiment_dir specifies the path of the directory in which the particular experiment is to be created. If experiment_dir is not entered, the default is the user directory specified by userdir.

experiment_number specifies the number, from 1 to 9999, of the experiment to be created.

Examples

```
cexp(3)
cexp('/data',2)
```

See also *NMR Spectroscopy User Guide*

Related		
	delexp	Delete an experiment (C)
	jexp	Join existing experiment (C)
	userdir	User directory (P)

cf Current FID (P)

Description Specifies which FID to operate on when working with multi-FID data. All subsequent operations such as Fourier transformation are applied to the selected data block.

When an experiment acquires nf number of data segments through explicit acquisition, cf indicates the cfth FID to use. For example, in the COSY-NOESY experiment with nf=2, cf=1 would select the COSY part of the experiment, and cf=2 would select the NOESY part.

Values 1 through the value of parameter nf.

See also *NMR Spectroscopy User Guide*

Related		
	nf	Number of FIDs (P)

cfpmult Calculate first-point multiplier for 2D experiments (M)

Description Calculates an fpmult value for the dataset, which is then used by wft2da. For 2D experiments, such as NOESY, run cfpmult on the transformed first increment, prior to entering wft2da, to minimize "f_2 ridges" in the final 2D spectrum. To do this manually for a 2D dataset, enter fpmult=1.0 wft(1) cdc in the command line and note whether the spectrum (essentially the baseline) moves up or down when dc is typed. Vary the value of fpmult until the dc correction (jump in the baseline) is as small as possible. With care, fpmult can be set to two decimal places. Typical values for fpmult range from 1.00 to 2.00. The default value is 1.0.

This calculation only needs to be performed for cosine-type experiments, such as NOESY, where both the t_2 FID and the t_1 interferogram decay. cfpmult might give incorrect values for first increments of experiments having baseline distortions (e.g., water suppression with 11-echo or 1331); in such cases, manual optimization of fpmult is more suitable.

When processing 2D data, unless the parameter lp is approximately 0, fpmult affects both the dc offset and the curvature of the spectrum. See the entries for alfa and calfa for more information.

See also *NMR Spectroscopy User Guide*

Related

alfa	Set alfa delay before acquisition (P)
calfa	Recalculate alfa so that first-order phase is zero (M)
crof2	Recalculate rof2 so that lp = 0 (M)
dc	Calculate spectral drift correction (C)
fpmult	First point multiplier for np FID data (P)
lp	First-order phase in directly detected dimension (P)
wft2da	Weight and Fourier transform phase-sensitive data (M)

change Submit a change sample experiment to acquisition (M)

Applicability Systems with automatic sample changer.

Description Removes the sample currently in the probe and loads the sample currently in sample location loc. change runs in the acquisition computer and is inoperative if loc is 0 and/or traymax is 'n' or 0. change also sets all hardware according to the current parameters.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (C)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
loc	Location of sample in tray (P)
lock	Submit an autolock experiment to acquisition (C)
sample	Submit change sample, Autoshim experiment to acquisition (M)

shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
su	Submit a setup experiment to acquisition (M)
traymax	Sample changer tray size (P)

checkstring Find and replace unwanted characters (C)

Syntax `checkstring('$VALUE',variable):variable`

Description `checkstring` is used panel to check and replace user-entered strings like `samplename`, `notebook`, or `page` for Unix-unfriendly characters:

" " (blank space) , ; : * ! ? (" ") [" "] {" "} <" "> ~ # $ & /

Data may be saved to unexpected directories (or not at all) with Save Data Setup (used for automatic saving of NMR data) if operating system special characters are used within a filename.

An error/warning message is issued and the respective character(s) is/are replaced with an underscore, _. Multiple consecutive characters are replaced by one single underscore. Example: samplename = 'special type of (new) sample' becomes '`special_type_of_new_sample`'.

chiliConf Control flag set by ecc_on and ecc_off (P)

Applicability Systems with Varian, Inc. Cold Probes

Description Control flag set by ecc_on and ecc_off macros

Values `E` — enable PSG control of ECC
`n` — disable PSG control of ECC

Related

ecc_on	Turns on eddy current compensation for Cold Probes (M)
ecc_off	Turns off eddy current compensation for Cold Probes (M)

chkname Parse the template and return substituted strings and lists of parameters defined by the template

Syntax
```
chkname(name_template, 'characters', 'par or tmpl or
str', 'replacechar'):$s1,$s2,$par,$req
chkname('fileChars', 'characters')
```

Applicability VnmrJ 3.1

Description The `chkname` command takes an argument that is a template type of the form used by the Svfname command. It parses the template and returns substituted strings and lists of parameters defined by the template. The chkname command will substitute parameters enclosed in the "$" substitution parameters. If the string of characters between the "$ pair" does not correspond to an existing parameter, the first $ character will be treated as a simple character. The chkname command will also substitute the time and date "% pairs" described above. It does this for all "% pairs" except the %Rn% or %Rn:number% specifiers. Like the Svfname command, the chkname command does not read a sample info file.

Arguments Following any substitutions, the chkname command will substitute characters in the resulting string, based on specified rules. The second argument to chkname specifies the characters that are allowed in the resulting string, or those that are disallowed in the resulting string. This is done because certain characters in filenames are either not allowed or they may make tools that use filenames difficult to use. Characters that often cause problems are:
' ', '!', '"', '$', '&', '\'', '(', ')', '*', ';', '<',
'>', '?', '\\', '[', ']', '^', '`', '{', '}', '|', ',',
'\0'
This second optional argument is either a keyword, a keyword plus modifiers, or a list of disallowed characters. Below are the keywords and the subset of characters they allow. The default is the keyword 'dir'.
'file' allows all alphanumeric characters [a-z, A-Z, and 0-9] and _ and . characters. These are typically used for file names.
'dir' allows all characters allowed by the 'file' keyword plus the directory specifier '/'.
'alnum' allows all alphanumeric characters [a-z, A-Z, and 0-9]
The alnum keyword can be followed by a list of other allowable characters.
'alnum_.' is identical to 'file'.
'alnum_./' is identical to 'dir'
'alnum_./:@%?=-&' might be used for email or web addresses.
'none' disables the character substitutions.
Supplying a list of characters is interpreted as disallowed characters. An example may be ' .,;:*!?()[]{}<>~#$%&/'
A special incantation of the chkname command will set the allowed characters selected by the 'file' and 'dir' keywords. The command chkname('fileChars','_') specifies the _ character in addition to alphanumerics as allowed characters for the 'file' keyword. The 'dir' keyword adds the '/' character to the 'file' set.
The third optional argument is the keyword 'par' or 'tmpl' or 'str' Actually, only the initial 'p', 't', or 's' is required, the longer name suggests the usage. The 'tmpl' (or 't', or 'template') keyword will do the "% pair" and "$ pair" substitutions before replacing disallowed characters. The 'str' (or 's', or 'string') keyword does not give any special meaning to the '%' or '$' characters. If they are found and they are in the disallowed list, they will be replaced with the replacement character. The 'par' (or 'p' or 'parameter') keyword is identical to the 'tmpl' keyword with respect to the first returned string. However, for the optional second returned string, it does not do any "$ pair" or "% pair" substitutions. It also does not replace the '$' characters with '#' characters (see below) as is the case with the 'tmpl' keyword. The 'par'

keyword is the default.
A fourth and final optional argument specifies the replacement character for any disallowed characters . The default is an underscore ('_'). A null string will remove any disallowed characters from the string.

The chkname command returns up to four values to the calling macro.

The first value returned is the expanded template. All places where a parameter is defined are substituted with the value of that parameter. If a used string parameter is an empty string, an empty string will be substituted for the template parameter. All "% pairs", except the %Rn% pairs will be substituted.

The second returned value depends on the value of the fourth optional argument to chkname. In the case of the 'tmpl' fourth argument, the second returned value is the same as the first returned value, except for the way in which empty string parameters are handled. In this case, instead of replacing the $parname$ with an empty string, it is replaced with the parameter name enclosed in # symbols. In the case of the 'str' fourth argument, the second returned value is identical to the first returned value. In the case of the default 'par' fourth argument, the second returned valued has no "$ pair" nor "% pair" substitutions.

The third returned value is a list of parameters defined by the template.

The fourth returned value is a subset all the defined parameters that are set to an empty string. This can be used to identify parameters that must be set before a template can be fully expanded.

In summary, the chkname command with a single argument will do the "$ pair" and "$ pair" substitutions (except for the %Rn% pairs) and replace any characters other than alphanumerics [a-zA-Z0-9] and '_', '.', and '/'. The chkname command should always be called before the Svfname command, which will do the final %Rn% pair substitution. The default values for the second, third, and fourth arguments will often be correct.

Examples

```
operator='vnmr1'
  comment='A special compound'
  samplename='C17H21NO4'
  ident=''
  $val='local'
  chkname('$samplename$ $$val$
$ident$/$operator$_$comment$_%R2%'):$s1,$s2,$p,$r


sets $s1 =
'C17H21NO4_local_/vnmr1_A_special_compound_%R2%'
       $s2 =
'C17H21NO4_local_#ident#/vnmr1_A_special_compound_%R2%'
       $p  = 'samplename $val ident operator comment'
```

```
      $r  = 'ident'

  chkname('$samplename$
$ident$/$operator$_$comment$_%R2%','file','par','.'):$s
1,$s2,$p,$r
  sets $s1 = 'C17H21NO4.vnmr1_A.special.compound_%R2%'
       $s2 =
'$samplename$.$ident$.$operator$_$comment$_%R2%'
       $p  = 'samplename ident operator comment'
       $r  = 'ident'
```

Cigar2j3j — Convert the parameter to a CIGAR2j3j experiment (M)

Syntax: Convert the parameter to a CIGAR2j3j experiment.

ckresloc — Macro to Reserve Specific Locations

Description: This macro checks the automation.conf file for any reserved locations. $ Operator specific reservations are in the following format: cppref_reserveloc_operator: 1 2 4. Similarly, dayQ and nightQ specific reservations are in the format: `cppref_DAYQ_ONLY`: 1 2 3 4 and cppref_NIGHTQ_ONLY: 41 42 43 neitherQ (i.e., blocked), respectively. Specific reservations are in the format: cppref_NEITHERQ: 23 38. Any location not specifically assigned are always allowed. If cppref_SMSLOCATION: is set to next, all locations except NEITHERQ are allowed.

Syntax: `ckresloc`

ckstring — Utility to Check String Variables for Illegal Characters

Description: This macro tests string variables for illegal characters.

Syntax: `ckstring('$VALUE',<argument2>):$return.`

Examples: ckstring('samplename',2):$samplename

Arguments: arg2=1 - Remove all special characters (default); arg2=2 - Removes all but forward slash; arg2=3 - Removes all but blank space: arg2=4 - Removes selected character.

cla — Clear all line assignments (M)

Syntax `cla`

Description Clears the line assignment parameters `clindex` and `slfreq` for spin simulation iteration, which matches simulated spectra to actual data.

See also *NMR Spectroscopy User Guide*

Related

`assign`	Assign transitions to experimental lines (M)
`dla`	Display line assignments (M)
`clindex`	Index of experimental frequency of a transition (P)
`slfreq`	Measured line frequencies (P)

cla — Calculated transition number (P)

Description A global arrayed parameter that stores the transition number of calculated transitions of the spin simulation program when they are above a threshold set by `sth`. In the iterative mode, the `cla` value of an assigned transition is associated with an experimental frequency whose index is the `clindex` value.

See also *NMR Spectroscopy User Guide*

Related

`clamp`	Calculated transition amplitude (P)
`clfreq`	Calculated transition frequency (P)
`clindex`	Index of experimental frequency of a transition (P)
`sth`	Minimum intensity threshold (P)

clamp — Calculated transition amplitude (P)

Description A global arrayed parameter that stores the transition amplitude of calculated transitions of the spin simulation program when they are above a threshold set by the parameter `sth`. Enter `dla('long')` to display `clamp`.

See also *NMR Spectroscopy User Guide*

Related

`cla`	Calculated transition number (P)
`clfreq`	Calculated transition frequency (P)
`clindex`	Index of experimental frequency of a transition (P)
`dla`	Display line assignments (C)
`sth`	Minimum intensity threshold (P)

cleanexp — Remove old files and directories from an experiment (M)

Syntax `cleanexp<(file1<,file2<,...>>)>`

Description Removes experiment subfiles from chained experiments that exist in an experiment directory. `cleanexp` only cleans the currently active experiment.

Arguments `file1, file2, ...` are specific experiment subfiles to be removed. If no argument is given, all files in `curexp/subexp` are removed.

Examples
```
cleanexp
cleanexp('H1','relayh')
```

See also *NMR Spectroscopy User Guide*

Related
`curexp` Current experiment directory (P)
`hccorr` Automated proton, carbon, and HETCOR acquisition (M)
`hcosy` Automated proton and COSY acquisition (M)

clear Clear a window (C)

Syntax `clear<(window_number)>`

Description Clears one of the four windows on the GraphOn terminal (status, input, graphics, text) or one of the two windows on the Sun (text and graphics).

Arguments `window_number` is the number (1 to 4) of the window to be cleared:

- `1` clears the status window (GraphOn only)
- `2` clears the graphics window
- `3` clears the input window (GraphOn only)
- `4` clears the text window (the default value).

Examples
```
clear
clear(2)
```

See also *User Programming*

cleardosy Delete temporarily saved data in current sub experiment (M)

Syntax `cleardosy`

Description Deletes any copies of DOSY data temporarily saved in the current sub experiment.

See also *NMR Spectroscopy User Guide*

Related
`dosy` Process DOSY experiments (M)

clfreq Calculated transition frequency (P)

Description	A global arrayed parameter that stores the transition frequency of calculated transitions of the spin simulation program when they are above a threshold set by the parameter sth. Enter dla to display clfreq.
See also	*NMR Spectroscopy User Guide*

Related

cla	Calculated transition number (P)
clamp	Calculated transition amplitude (P)
clindex	Index of experimental frequency of a transition (P)
dla	Display line assignments (M)
sth	Minimum intensity threshold (P)

clindex Index of experimental frequency of a transition (P)

Description	A global arrayed parameter where each value contains the index of an experimental frequency assigned to the associated calculated transition for use in iterative spin simulation. Use assign to make the assignments. A value of zero indicates no assignment.
See also	*NMR Spectroscopy User Guide*

Related

assign	Assign transitions to experimental lines (M)
cla	Clear line assignments (M)
cla	Calculated transition number (P)
dla	Display line assignments (M)

clradd Clear add/subtract experiment (C)

Description	Deletes the add/subtract experiment (exp5).
See also	*NMR Spectroscopy User Guide*

Related

add	Add current FID to add/subtract experiment (C)
sub	Subtract current FID from add/subtract experiment (C)

cmdlineOK Determine if an operator has a command line

Applicability VnmrJ 3.1

Description The cmdlineOK program queries whether the current operator has the command line enabled. This command is not typically used directly by an operator. It is used by the interface designer to determine if and how certain options should be presented.

The VnmrJ administrator interface is used to grant access to the

command line.
If the operator has access to the command line, the `cmdlineOK` program will return a 1 to the calling macro.
In the example below, `$ok` will be set to 1. It will be set to 0 if the command line is not available.
`cmdlineOK:$ok`
An optional argument can be provided. This will be the return value if command line access is not granted. For example,
`cmdlineOK(-1):$ok` will set `$ok` to `-1` if command line access is not granted. This can be used by the interface designed so that a button may be either "grayed out" or removed if command line access is not granted.

`coldprobe` Tells system a coldprobe is present

Applicability VnmrJ 3.1

Description The `coldprobe` macro tells the system that a coldprobe is present so that the `rof2` rule is enforced.

Arguments If a C13 observe coldprobe is being used, the value of `rof2` should not be less than 350 usec.

`color` Select plotting colors from a graphical interface (M)

Description Displays a window with color palettes for selecting colors for plotting the background of the display screen, spectrum, integral, FID, etc.

See also *NMR Spectroscopy User Guide*

Related

`pl`	Plot spectra (C)
`setcolor`	Set colors for graphics window and for plotters (C)

`combiplate` View a color map for visual analysis of VAST microtiter plate (U)

Syntax (From UNIX) `combiplate`

Description Opens the CombiPlate window, which provides a map of microtiter plate, allowing data to be viewed from individual sample wells. The window enables viewing integral region intensities by colors and color densities.

See also *NMR Spectroscopy User Guide*

Related

`combishow`	Display regions as red, green, and blue in CombiPlate window (M)
`dlivast`	Produce text file and process last wells (M)

combishow Display regions (red, green, and blue) in CombiPlate window (M)

Syntax `combishow(r,g,b)`

Description Displays integral regions shown on the spectrum as red (`r`), green (`g`), and blue (`b`) in the CombiPlate window. CombiPlate reads the regions automatically. 1, 2, or 3 integral regions can be designated. At least one integral region must be specified. Combishow displays spectra associated with individual wells.

See also *NMR Spectroscopy User Guide*

Related
`combiplate` View a color map for visual analysis of VAST microtiter plate (U)
`dlivast` Produce text file and process last wells (M)

compressfid Compress double-precision FID data (M,U)

Syntax `compressfid(<inFIDdir,>outFIDdir)`
(From UNIX) `compressfid -i inFIDdir -o outFIDdir -f`
(From UNIX) `compressfid -e exp_number -o outFIDdir -f`

Description Compresses double-precision FID data to single-precision and updates the parameter `dp` in the file `propcar`. `compressfid` can be run through a macro interface in VnmrJ or directly at the UNIX level. In entering FID directory names, leave off the `.fid` directory extension.

Arguments `inFIDdir` is the double-precision FID directory to be compressed. If `inFIDdir` is not entered, the default FID directory is `curexp/acqfil`.

`outFIDdir` is the FID directory to receive the output.

`exp_number` is the number of the experiment that contains the FID data.

`-i` specifies that the next argument is the input FID directory.

`-o` specifies that the next argument is the output FID directory.

`-e` specifies that the next argument is the number of the experiment that contains the FID data. The `-e` and the `-i` options are mutually exclusive.

`-f` specifies that any existing directory with the name `outFIDdir.fid` is to be overwritten. Note that the macro interface always overwrites any preexisting directory with the name specified by `outFIDdir.fid`.

Examples
```
compressfid('/vnmr/fidlib/fid1d',
   'testfid1d')compressfid('testfid1d')
```
(From UNIX) `compressfid -e 5 -o testfid1d -f`
(From UNIX) `compressfid -i /vnmr/fidlib/fid1d -o testfid1d -f`

See also *NMR Spectroscopy User Guide*

Related `dp` Double precision (P)

config Display current configuration and possibly change it (M)

Syntax `config <('display')>`

Applicability VnmrJ 3.2

Description The "config" command displays the current configuration. The configuration can be changed if the console is in use and the user has write access to the following:

- system global parameter file
- the “stdpar” link in the VNMR system directory
- the file that the “stdpark” link points to
- the “tests” link" in the VNMR system directory,
- the file that the “tests” link points to
- the VNMR system directory

In this situation, a window will appear on top of the VNMR windows. The single argument 'display' will PREVENT the interactive mode from operating.

Usually, the VNMR system manager will configure the system once, and the set the protection on the parameter file to permit read access only by other users.

In interactive mode, a separate panel appears. In non-interactive mode, the current choices are displayed in the text window. See the installation manual for details on the choices.

Arguments `'display'` is a keyword that the system administrator can use to make `config` run in the display mode rather than the interactive mode.

Examples
```
config
config('display')
```

See also *VnmrJ Installation and Administration*

Related

`amptype`	Amplifier type (P)
`audiofilter`	Audio filter type (P)
`cattn`	Coarse attenuator (P)
`Console`	System console type (P)
`fattn`	Fine attenuator (P)
`fifolpsize`	FIFO loop size (P)
`gradtype`	Gradients for X, Y, and Z axes (P)
`h1freq`	Proton frequency of spectrometer (P)
`latch`	Frequency synthesizer latching (P)
`lockfreq`	Lock frequency (P)
`numrfch`	Number of rf channels (P)
`overrange`	Frequency synthesizer overrange (P)

parmax	Parameter maximum values (P)
parmin	Parameter minimum values (P)
parstep	Parameter step size values (P)
ptsval	PTS frequency synthesizer value (P)
rfchtype	Type of rf channel (P)
rftype	Type of rf generation (P)
rfwg	RF waveform generator (P)
rotorsync	Rotor synchronization (P)
shimset	Type of shim set (P)
sysgcoil	System gradient coil (P)
system	System type (P)
traymax	Sample changer tray slots (P)
vttype	Variable temperature controller present (P)

confirm — Confirm message using the mouse (C)

Syntax `confirm(message):response`

Description Displays a dialog box with the specified message and two buttons: Confirm and Cancel. Clicking on the buttons with the mouse produces a return value.

Arguments `message` is a single-line muticharacter string to be shown in the dialog box.

`response` is 1 if the user clicks the left button of the mouse on the Confirm button or presses the Return key; `response` is 0 if the user clicks the mouse on the Cancel button.

Examples `confirm('Are you sure you want pw>100?'):$response`

See also *User Programming*

Console — System console type (P)

Description A global parameter that sets the type of system console The value is usually set using the Console label in the Spectrometer Configuration window (opened from `config`).

When `go`, `au`, or `ga` is entered, the value of the `Console` parameter is copied from the systemglobal parameter tree to the current experiment and named as the `console` parameter (lowercase c). If `console` does not exist in an old parameter set, `rt` via `fixpar` creates it and sets it to `''`. Both `console` and `Console` are type acquisition. Macros can use `Console` and `console` to take conditional action based on spectrometer type.

See also *VnmrJ Installation and Administration*

Related

au	Submit experiment to acquisition and process data (M)
config	Display current configuration and possibly change it (M)

fixpar	Correct parameter characteristics in experiment (M)
ga	Submit experiment to acquisition and FT the results (M)
rt	Retrieve FIDs (M)
go	Submit experiment to acquisition (M)
system	System type (P)

contact_timeMAS cross-polarization spin-lock contact time (M)

Applicability Systems with solids module.

Description Processes data obtained using an array of values for a pulse-length parameter. It runs the UNIX program `expfit`, which does an exponential curve fitting that determines the value of *Tch* and *T1rho*. The output is matched to the equation

*I = [S0 - (S0 - S_inf)*exp(-T/Tch))*exp(-T/T1rho)) + S_inf*

where *Tch* is the time constant of a spin-locked cross-polarization process, and *T1rho* is relaxation time of ^{13}C polarization in the proton rotating field.

The required input is file `fp.out` from the program `fp` and the values of the arrayed parameter. The output table is file `analyze.list` in the current experiment. The file `analyze.out` is used by the `expl` to display the results.

See also *User Guide: Solid-State NMR*

Related

expfit	Least-squares fit to polynomial or exponential curve (U)
expl	Display polynomial/exponential curves (C)
fp	Find peak heights (C)

continflag The command ddif creates a CONTIN display if continflag='y'.

Syntax `continflag`

Applicability VnmrJ 3.1

Description Tells the command `ddif` to create a 2D display using data produced by the CONTIN program. Set by the `dosy` macros, does not normally need to be set manually.

Arguments continflag = ‘y’

continflag = ‘n’

See also `dosyproc`

continprepare Called by the macro dosy to prepare the input file for the CONTIN programme.

Syntax `continprepare`

Applicability VnmrJ 3.1

Description `continprepare` takes the `dosy_in` file created in dosy and creates the file dosy_contin.in the format required by the CONTIN programme (http://s-provencher.com/index.shtml).

See also `continread`

`dosy`

`splmodprepare`

continread Called by the macro dosy to take the output of the CONTIN programme and create an input file for ddif.

Syntax `continread`

Applicability VnmrJ 3.1

Description `continread` takes a file dosy_contin.out as created by CONTIN programme, run by the continrun shell script from the `dosy` macro, and creates the files diffusion_display. contin and diffusion_spectrum used by `ddif` and `sdp` to display DOSY results.

Arguments `continread` takes no arguments

See also `ddif`

`dosy`

continueMovie Continue movie in either forward or backward direction (C)

Syntax `continueMovie(rate)`

Description Like `startMovie`, but can continueMovie can play a movie forward or back ward, and, instead of always starting from the beginning, it starts from the beginning if movie has not started yet, or continues from where it was stopped (by `stopMovie`). Movie direction is controlled by parameter `aipMovieSetting[3]=1` or `-1`.

Arguments `aipMovieRate`, or a number for the rate

See also `startMovie`, `stopMovie`, `resetMovie`.

convert — Convert data set from a VXR-style system (M,U)

Syntax `convert(VXR_file)`
(From UNIX) `cpos_cvt VXR_file`

Description Converts data stored on a VXR-style system (VXR, XL, or Gemini) to the format used in software. The macro `convert` loads the data from `VXR_file` into the current experiment and converts it to the new format. The UNIX command `cpos_cvt` writes the converted data in a subdirectory of the current working directory, using the original name of the data set.

Arguments `VXR_file` is the name of a VXR-style file to be converted to VnmrJ style

See also *NMR Spectroscopy User Guide*

Related `cpos_cvt` Convert data set from a VXR-style system (C,U)
`decomp` Decompose a VXR-style directory (C)

convertbru — Convert Bruker data (M,U)

Syntax (From UNIX) `convertbru file <options>`
`convertbru(file<,options>)`

Description A C-language program for converting 32-bit Bruker AMX data and 24- and 32-bit Bruker AM data into a 32-bit format compatible with the Varian `sread` program. After converting the Bruker data into the new format, the converted data can be read into VnmrJ using `sread` and can then be processed normally. The parameters `proc` and `proc1` are set appropriately by `sread`, so that `wft` or `wft2da` correctly processes the data.

Bruker AM parameters are converted to Varian parameters as shown in the table “AM Parameter Conversion.” Bruker parameter names that do not conflict with a Varian parameter name are converted under the original name: `td`, `fw`, `ds`, `o1`, `o2`, `ns`, `te`, `id`, `sfo1`, `sfo2`, and `ro`. Parameters `proc` and `proc1` are set to `'rft'` for all spectra (assuming TPPI data in both dimensions).

AM Parameter Conversion

Table 1.

Bruker	*Varian*	*Bruker*	*Varian*
sweeps completed	`ct`	`sp`	`satdly`
`td`	`np`	`dp`	`dpwr`
`dw`	`dw`	`te`	`temp=te–273`
`fw`	`fb=1.1*sw/2`	`id`	`sw1=1/id`
`ds`	`ss`	`sfo1`	`sfrq=sfo1+o1`
`sw`	`sw`	`sfo2`	`dfrq=sfo2+o2`

Table 1.

Bruker	*Varian*	*Bruker*	*Varian*
experiments done	ni	p#	p#
o1	tof	d#	d#
o2	dof	s#	s#
rd (or d1 if rd=0)	rd	ro	spin
pw (or p0 if pw=0)	pw	rg	gain
p1	pw90	date	date
de	de	time	time
ns	nt		

Bruker AMX parameters are converted to Varian parameters as shown in the table "AMX Parameter Conversion." All Bruker parameters are converted under their original names if the name doesn't conflict with the name of a Varian parameter. Arrayed Bruker parameters like P and D are converted to the names P# and D#, where # is the index into the array.

Because sread is limited to 8-character parameter names, the parameters routwd1# and routwd2# are converted to rtwd1# and rtwd2#.

The parameter proc is set to 'ft' when the Bruker parameter aq_mod is 1, and proc is set to 'rft' when aq_mod is 2. proc1 is always set to rft, assuming TPPI in t1.

If there is a file named info in the directory with the Bruker data, it is read in and put into the text file for the converted data set.

AMX Parameter Conversion

Table 2.

Bruker	*Varian*	*Bruker*	*Varian*
ns (from acqu)	nt	te	temp=te-273
ns (from acqus)	ct	sfo1	sfrq=sfo1
td (from acqus)	np	sfo2	dfrq=sfo2
td (from acqu2s)	ni	o1	tof
sw_h	sw	o2	dof
sw_h	dw=1.0e6/sw	ro	spin
sw_h (from acqu2s)	sw1	rg	gain
fw	fb=1.1*sw/2	date	date
ds	ss	date	time
rd (or d1 if rd=0)	rd	nucleus	tn
de	de	decnuc	dn

Table 2.

Bruker	*Varian*	*Bruker*	*Varian*
pw (or p0 if pw=0)	pw	pulprog	pslabel
p1	pw90	pulprog	seqfil

Arguments `file` is the input file name. For AMX data, `file` should be the name of the directory that contains the `acqus`, `acqu2s`, and `fid` or `ser` files. For AM data, `file` should be the name of the file containing the AM data. The `file` argument is not required to have a `.bru` extension, but if it does, the `.bru` extension is removed before creating the output file. Unless the `–cfile` option is present, the output file will have the same name as the input file, but with a `.cv` extension, and will be written into the current working directory.

`options` for AMX and AM data are the following, which can be entered in any order as long as `file` comes first (options are usually not necessary, but can be used to override the default actions of `convertbru`):

- `–bam` or `–bamx` specifies whether input is AM or AMX data. The default is determined from name of the input file given.
- `–cfile` specifies that the output file is given the name specified by `file` and is written with `.cv` appended to the name
- `–dxxx`, where `xxx` is the decoupler frequency (it must be a value between 10.0 and 640.0 MHz). The default is to read from data set.
- `–f` specifies that old output file is to be overwritten. The default is to not overwrite old files.
- `–olsb` or `–omsb` specifies whether the data has the least- or most-significant byte first. For AM data, the default is determined from data set. For AMX data, the default is `–olsb`.
- `–pxxx`, where `xxx` is the number of 24- or 32-bit words to skip before converting data. This option is for use with `–t` option to skip the header in AM data without converting it. Typical header sizes are 216 or 256 words. The default is 0.
- `–s3` or `–s4` specifies if AM data is 24-bit (3-byte) or 32-bit (4-byte). All AMX data is 32-bit. The default is determined from the data set.
- `–tall`, `–thdr`, or `–tdata` specifies whether `convertbru` should convert the header and the data, just the header, or just the data. The default is `–tall`.

Examples Convert AM data from a UNIX shell (in all these examples, the file name is arbitrarily named `br_data`):

- `convertbru br_data` determines the file format and converts the header and data in the file `br_data`.
- `convertbru br_data –d250.0 –cout` determines the file format, converts the header and data in the `br_data`, sets the decoupler frequency to 250.0 MHz, and writes to an output file named `out.cv` in the current working directory.

- `convertbru br_data -thdr` determines file format and converts only the header in the file `br_data`.
- `convertbru br_data –tdata –p256 –s3 –omsb` converts only the data in `br_data` after skipping the 256-word header. The data is converted assuming it is 24-bit AM data words with the most-significant byte first.

Convert AM data from VnmrJ:

- `convertbru('br_data','–tdata','–p256','–s3','–omsb')` converts only the data in `br_data` after skipping the 256-word header. The data is converted assuming it is 24-bit AM data words with the most-significant byte first.

Convert AMX data from a UNIX shell:

- `convertbru br_data –f` converts `acqus` and `acqu2s` files to ASCII, if needed, and then converts data and overwrites the existing `br_data.cv` file.

Convert AMX data from VnmrJ:

- `convertbru('br_data','–f')` converts `acqus` and `acqu2s` files to ASCII, if needed, and then converts data and overwrites the existing `br_data.cv` file.
- `convertbru('br_data','–c/home/vnmr1/bdata/data1')` converts `acqus` and `acqu2s` files to ASCII, if needed, and then converts the data and writes it to `/home/vnmr1/bdata/data1.cv`.

See also *NMR Spectroscopy User Guide*

Related	readbrutape	Read Bruker data files from 9-track tape (U)
	sread	Read converted data into VnmrJ (C)
	wft2da	Weight and Fourier transform phase-sensitive data (M)

copy — Copy a file (C)

Syntax `copy(<'–r',>from_file,to_file)<:$res>`

Description Makes a copy of a file and is identical to the `cp` command. All arguments are passed. Command will abort with no return value if an illegal file name is used.

Arguments `'–r'` — keyword requesting a recursive copy (i.e., copy a directory).

`from_file` — name of the file (or directory if `'–r'` used) to be copied.

`to_file` — name of the copy of the file (or directory). If the `from_file` argument has an extension (e.g., `.fid`), be sure the `to_file` argument has the same extension.

`:$res` — variable to hold the result of the copy process.

1 is returned if the copy is successful.

0 is returned if the copy failed.

Examples `copy('-r','/home/vnmr1/vnmrsys/seqlib','/vnmr/seqlib')`

```
copy('/home/vnmr1/vnmrsys/seqlib/d2pul', \
   '/vnmr/seqlib/d2pul')
```

See also *NMR Spectroscopy User Guide*

Related `cp` Copy a file (C)

cos — Find cosine value of an angle (C)

Syntax `cos(angle)<:n>`

Description Finds the cosine of an angle.

Arguments `angle` is the angle, given in radians.

`n` is the return value with the cosine of `angle`. The default is to display the cosine value in the status window.

Examples
```
cos(.5)
cos(val):cos_val
```

See also *User Programming*

Related `sin` Find sine value of an angle (C)

Cosy — Convert the parameter to a COSY experiment (M)

Description Convert the parameter to a COSY experiment.

See also *NMR Spectroscopy User Guide*

Related

`cosyps`	Set up parameters for phase-sensitive COSY pulse sequence (M)
`Dqcosy`	Set up parameters for double-quantum filtered COSY (M)
`relayh`	Set up parameters for RELAYH pulse sequence (M)

cosyps — Set up parameters for phase-sensitive COSY pulse sequence (M)

Description Sets up a phase-sensitive COSY (homonuclear correlation) experiment.

See also *NMR Spectroscopy User Guide*

Related

`Cosy`	Set up parameters for COSY pulse sequence (M)
`Dqcosy`	Set up parameters for double-quantum filtered COSY (M)
`relayh`	Set up parameters for RELAYH pulse sequence (M)

cp — Copy a file (C)

Syntax `cp(<'-r',>from_file,to_file)<:$res>`

Description Makes a copy of a file and is identical to the `copy` command. All arguments are passed. Command will abort with no return value if an illegal file name is used.

Arguments `'-r'` is a keyword requesting a recursive copy (i.e., copy a directory).

`from_file` is the name of the file (or directory if `'-r'` used) to be copied.

`to_file` is the name of the copy of the file (or directory). If the `from_file` argument has an extension (e.g., `.fid`), be sure the `to_file` argument has the same extension.

`:$res` variable to hold the result of the copy process.

1 is returned if the copy is successfully

0 is returned if the copy failed

Examples

```
cp('/home/vnmr1/vnmrsys/seqlib/d2pul',    \
   '/vnmr/seqlib/d2pul')
cp('-r','/home/vnmr1/vnmrsys/seqlib','/vnmr/seqlib')
```

See also *NMR Spectroscopy User Guide*

Related		
	`copy`	Copy a file (C)

cp — Cycle phase (P)

Description Sets the values that real-time variable oph is calculated as, either 0,1,2,3 (`cp='y'`) or 0 (`cp='n'`). The only circumstance where setting `cp='n'` may be useful is when displaying an FID with `acqi`. If there is an imbalance between the two receiver channels, the FID displayed for `acqi` may show alternating dc levels. The standard `gf` macro that prepares parameters for the FID display in `acqi` automatically handles this issue.

Values `'y'` makes oph calculate as 0,1,2,3; this is the typical value.
`'n'` makes oph calculate as 0.

See also *User Programming*

Related		
	`acqi`	Interactive acquisition display process (C)
	`go`	Submit experiment to acquisition (C)
	`gf`	Prepare parameters for FID/spectrum display in `acqi` (M)

cpdone — Macro called upon study completion (M)

Syntax

Applicability VnmrJ 3.1

Description This macro is called when a study is completed or paused. This system macro should never be edited. Changes should be implemented in the `usercpdone` macro.

See also User Guide: Automation-User Space Customization

Related `usercpdone` (m)

`cpgo` Macro called when acquisition is started (M)

Syntax

Applicability VnmrJ 3.1

Description This macro is called when an acquisition is started.

This system macro should never be edited.

Changes should be implemented in the usergo macro.

Examples User Guide: Automation-User Space Customization

See also

Related `usercpgo (m)`

`cpmgt2` Set up parameters for CPMGT2 pulse sequence (M)

Description Macro to set up a CPMGT2 (Carr-Purcell Meiboom-Gill T_2) experiment.

See also *NMR Spectroscopy User Guide*

Related `t2` T_2 exponential analysis (M)

`cpos_cvt` Convert data set from a VXR-style system (M,U)

Syntax (From UNIX) `cpos_cvt VXR_file`
`convert(VXR_file)`

Description Converts data stored on a VXR-style system (Gemini, VXR, or XL) to the format used in VnmrJ software. `cpos_cvt` writes the converted data in a subdirectory of the current working directory, using the original name of the data set. The command `convert` loads the data from `VXR_file` into the current experiment and converts it to the new format.

Arguments `VXR_file` is the file name in the VXR-style format to be converted to the VnmrJ style.

Related
`convert` Convert data set from a VXR-style system (C,U)
`decomp` Decompose a VXR-style directory (C)
`rt` Retrieve FIDs (C)

cptmp Copy experiment data into experiment subfile (M)

Syntax `cptmp<(file)>`

Description Copies the data (parameters, FID, and transformed spectrum) from the current experiment into a subdirectory inside `curexp+'/subexp'`.

Arguments `file` is the name of the subfile to receive the data. The default is to take the name from the transmitter nucleus (if `seqfil='s2pul'`) or to use the pulse sequence name.

Examples
```
cptmp
cptmp('cosy')
```

Related		
	curexp	Current experiment directory (P)
	rttmp	Retrieve experiment data from experiment subfile (M)
	seqfil	Pulse sequence name (P)
	svtmp	Move experiment data into experiment subfile (M)

cptmpltdefaults Defaults for Save Data Template

Examples This macro sets the default values used for creating the save-data template in the Preferences/Templates popup. It is called when the "Restore to Defaults" button on the Preferences/Templates popup is clicked.

cpx Create pbox shape file (M)

Syntax `cpx<(ref_pw90,ref_pwr)> or cpx<('g')>`

Description Calls UNIX command `Pbox`, which generates the specified pulse shape or decoupling/spin locking pattern, as defined by the `shapelib/Pbox.inp` file.

Arguments `ref_pw90` is the reference 90° pulse width

`ref_pwr` is the reference power level.

`'g'` is a keyword that is required only when generating gradient shapes and if the file type is not specified otherwise.

Examples
```
cpx
cpx('g')
cpx(pw90*compH,tpwr)
```

See also *NMR Spectroscopy User Guide*

Related		
	Pbox	Pulse shaping software (U)

cqexp Load experiment from protocol (M)

Applicability Liquids

Description Macro to load an experiment from a protocol.

Syntax `cqexp(experiment <, apptype>)`

The first argument is the experiment name, and the second argument is the apptype. If the apptype is not specified, the previous apptype is used.

Examples `cqexp('Proton', 'std1d')`

Related
`apptype` Application type (P)
`execpars` Set up the exec parameters (M)

cqfindz0 Run an experiment to find the value of z0 (M)

Applicability Liquids

Description A macro to run a deuterium experiment to find the correct value of `z0` for a given solvent. It requires an entry in the probe file for the number of deuterium Hz per DAC. Run the appropriate probe calibration for lk Hz per DAC to set the value in the probe file. The macro may be accessed through the Find `z0` button available on several panels.

Related
`solvent` Lock solvent (P)
`z0` Z0 field position (P)

cqgmap Perform gradient shimming utility functions (M)

Applicability Liquids

Description Macro runs gradient shimming utility functions.

Related
`gmapshim` Run gradient autoshimming, set parameters, map shims (M)

cqinit Initialize liquids study queue (M)

Applicability Liquids

Description Initializes the liquids study queue.

Related
`cqreset` Reset study queue parameters (M)
`sqfilemenu` Study queue file menu commands (M)

cqpars Create study queue parameters for liquids (M)

Applicability Liquids

Description A macro to create study queue parameters for the Walkup interface.

See also VnmrJ Walkup

Related `fixpar` Correct parameter characteristics in experiment (M)

cqplot Macro to perform generic 2D plot (M)

Applicability Liquids

Description A macro to perform generic 2D plotting, including 1D experiment traces. Usually called by other macros, and not used from the command line.

Related
`plot`	Automatically plot spectra (M)
`plot2D`	Plot results of 2D peak picking (C)
`plt2Darg`	Plot 2D arguments (P)

cqprotocol Macro to create protocols (M)

Applicability Liquids

Description A macro to create protocols for liquids applications. Called by the *Make protocols dialogs* in the Utilities menu.

cqreset Reset study queue parameters (M)

Applicability Liquids

Description Reset liquids study queue parameters. Usually called by other macros when starting a new study.

Related
`cqinit`	Initialize liquids study queue (M)
`sqfilemenu`	Study queue file menu commands (M)

cqsavestudy Macro to save study queue parameters (M)

Applicability Liquids

Description A macro to save study parameters in the liquids study queue. Usually called by other macros when starting a new study.

Related
`studypar` Study parameters (P)
`xmsubmit` Submit sample(s) to the study queue (M)
`xmendq` End a chained study queue (M)

cqwtmenu Macro to set weighting functions from a panel (M)

Applicability Liquids, Imaging

Description A macro to set weighting functions from a panel. It is used for both 1D and 2D weighting parameters. Called by processing parameter panels.

cr Cursor position in directly detected dimension (P)

Description Contains the current cursor position. The `rl` macro uses `cr` to set the reference line.

See also *NMR Spectroscopy User Guide*

Related
`centersw` Move cursor to center of spectrum (M)
`crf` Current time-domain cursor position (P)
`crl` Clear ref. line in directly detected dimension (M)
`delta` Difference of two frequency cursors (P)
`rl` Set reference line in directly detected dimension (M)

cr1 Cursor position in 1st indirectly detected dimension (P)

Description Contains the current cursor position along the first indirectly detected dimension. Analogous to the `cr` parameter except that `cr1` applies to the first indirectly detected dimension of a multidimensional data set. The `rl1` macro uses `cr1` to set the reference line along this dimension.

See also *NMR Spectroscopy User Guide*

Related
`centersw1` Move cursor to center of spectrum in 1st indirect dimension (M)
`cr` Cursor position in directly detected dimension (P)
`cr2` Cursor position in 2nd indirectly detected dimension (P)
`rl1` Set ref. line in 1st indirectly detected dimension (M)

cr2 Cursor position in 2nd indirectly detected dimension (P)

Description Contains the current cursor position along the second indirectly detected dimension. Analogous to the cr parameter except that cr2 applies to the second indirectly detected dimension of a multidimensional data set. The rl2 macro uses cr2 to set the reference line along this dimension.

See also *NMR Spectroscopy User Guide*

Related

centersw2	Move cursor to center of spectrum in 2nd indirect dimension (M)
cr	Cursor position in directly detected dimension (P)
cr1	Cursor position in 1st indirectly detected dimension (P)
rl2	Set ref. line in 2nd indirectly detected dimension (M)

crcom Create user macro without using text editor (M)

Syntax `crcom(file,actions)`

Description Creates a macro file in the user's macro library (maclib) with the contents given in the actions argument.

Arguments file is the file name of the user macro to be created. If a macro of the same name already exists, the user is asked whether or not to overwrite it.

actions is a string containing the actions making up the user macro. The string cannot include a carriage return. If a single quote is needed within the string, it must be preceded by a backslash (see second example below).

Examples
```
crcom('plot','pl pscale pap page')
crcom('lds','load=\'y\' su load=\'n\'')
```

See also *User Programming*

create Create new parameter in a parameter tree (C)

Syntax `create(parameter<,type<,tree>>)`

Description Creates a parameter in one of the parameter trees. A *parameter tree* is a UNIX file containing the attributes of parameters as formatted text. Refer to the command paramvi for a description of the file contents.

Arguments parameter is the name of the parameter to be created.

type is the type of values in the parameter to be created and can be one of the following values (default is 'real'):

- 'real' is a value with no limits on range and can be positive or negative.

- 'string' is a value composed of characters. Entry of strings can be limited to selected words by enumerating the possible values with the command setenumeral. For example, the enumerated values of intmod are 'off', 'partial', and 'full'. Therefore, intmod can be set only to one of these three string values, such as intmod='full'.
- 'delay' is a value from 0 to 8190, in unit of seconds.
- 'frequency' is a positive real number value.
- 'flag', like 'string', is a value composed of characters. Entry of flags can be limited to selected characters by enumerating the possible values with the command setenumeral. For example, the enumerated values of dmm are 'c', 'f', 'g', 'm', 'p', 'r', 'u', 'w', and 'x'. Therefore, dmm can only be set to a combinations of these nine characters, such as dmm='ccw'. If enumerated values are not set, the 'string' and 'flag' types are identical.
- 'pulse' is a value from 0 to 8190, in units of μs.
- 'integer' is a value composed of integers (0,1,2,3,...).

tree is one of the following types of parameter trees (default is 'current'):

- 'current' contains parameters that are adjusted to set up an experiment. The parameters are from the file curpar in the current experiment.
- 'global' contains user-specific parameters from the file global in the vnmrsys directory of the present UNIX user.
- 'processed' contains parameters with which the data was obtained. These parameters are from the file procpar in the current experiment.
- 'systemglobal' contains instrument-specific parameters from the text file /vnmr/conpar. Most of these parameters are defined using the config program. All users have the same systemglobal tree. Note that conpar is not written out when you exit; the only time conpar is ever modified is by the config program. Thus, any changes you make to conpar using create (or destroy, setvalue, etc.) are not permanent. To permanently create a parameter in conpar, you must use a text editor to change /vnmr/conpar.

Examples
```
create('a')
create('b','string')
create('c','real','global')
```

See also *User Programming*

Related

destroy	Destroy a parameter (C)
display	Display parameters and their attributes (C)
fread	Read parameters from file and load them into a tree (C)
fsave	Save parameters from a tree to a file (C)
paramvi	Edit a parameter and its attributes using vi text editor (M)
prune	Prune extra parameters from current tree (C)

setenumeral	Set values of a string variable in a tree (C)
setgroup	Set group of a parameter in a tree (C)
setprotect	Set protection mode of a parameter (C)

create (P) Parameter used for RF transmitter board temperature compensation

Syntax `create('rftempcomp','string','global')`

Applicability VnmrJ 3.1

Arguments If rftempcomp='n' temperature compensation on the RF transmitter board is turned off.

If rftempcomp='y' temperature compensation on the RF transmitter board is turned on and will make a single compensation.

If rftempcomp='c' temperature compensation on the RF transmitter board is turned on continuously and will continuously update until it is turned off.

createqcomp Create qcomp parameter (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Macro to create the qcomp parameter with the appropriate attributes. qcomp is created as a flag parameter in the global tree.

crf Current time-domain cursor position (P)

Description Contains current time-domain cursor position. To create crf and the other FID display parameters axisf, dotflag, vpf, vpfi, and deltaf (if the parameter set is older and lacks these parameters), enter addpar('fid').

Values Number, in seconds.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
crl1	Clear ref. line in 1st indirectly detected dimension (C)
deltaf	Difference of two time cursors (P)
fidpar	Add parameters for FID display in current experiment (M)

crl — Clear reference line in directly detected dimension (M)

Description: Clears frequency referencing along the directly detected dimension by setting the reference parameters `rfl` and `rfp` to zero. `crl` also resets the referencing parameters `refpos` and `reffrq`.

See also: *NMR Spectroscopy User Guide*

Related:

`crl1`	Clear ref. line in 1st indirectly detected dimension (C)
`crl2`	Clear ref. line in 2nd indirectly detected dimension (C)
`rl`	Set ref. line in directly detected dimension (M)
`reffrq`	Reference frequency of reference line (P)
`refpos`	Position of reference frequency (P)
`rfl`	Ref. peak position in directly detected dimension (P)
`rfp`	Ref. peak frequency in directly detected dimension (P)

crl1 — Clear reference line in 1st indirectly detected dimension (M)

Description: Clears frequency referencing along the first indirectly detected dimension by setting the reference parameters `rfl1` and `rfp1` to zero. `crl1` also resets the referencing parameters `refpos1` and `reffrq1`.

See also: *NMR Spectroscopy User Guide*

Related:

`crl`	Clear ref. line in directly detected dimension (C)
`rl1`	Set ref. line in 1st indirectly detected dimension (M)
`reffrq1`	Ref. frequency of reference line in 1st indirect dimension (P)
`refpos1`	Position of reference frequency in 1st indirect dimension (P)
`rfl1`	Ref. peak position in 1st indirectly detected dimension (P)
`rfp1`	Ref. peak frequency in 1st indirectly detected dimension (P)

crl2 — Clear reference line in 2nd indirectly detected dimension (M)

Description: Clears frequency referencing along the second indirectly detected dimension by setting the reference parameters `rfl2` and `rfp2` to zero. `crl2` also resets the referencing parameters `refpos2` and `reffrq2`.

See also: *NMR Spectroscopy User Guide*

Related:

`crl`	Clear ref. line in directly detected dimension (C)
`rl2`	Set ref. line in 2nd indirectly detected dimension (M)
`reffrq2`	Ref. frequency of reference line in 2nd indirect dimension (P)

refpos2	Position of reference frequency in 2nd indirect dimension (P)
rfl2	Ref. peak position in 2nd indirectly detected dimension (P)
rfp2	Ref. peak frequency in 2nd indirectly detected dimension (P)

crmode — Current state of the cursors in df, ds, or dconi programs (P)

Description Stores the current state (box mode or cursor mode) of cursors in the df, ds, or dconi interactive display programs. crmode is mostly used by programmable menus to determine the status of the cursors. It is stored in the file vnmrsys/global.

Values 'b' signifies the box mode, 'c' signifies the cursor mode.

See also *User Programming*

Related

dconi	Interactive 2D data display (C)
df	Display a single FID (C)
ds	Display a spectrum (C)

crof2 — Recalculate rof2 so that lp = 0 (M)

Syntax crof2<(alfa)>

Description Recalculates a new value for rof2 (receiver gating time following a pulse) based upon the current rof2 and lp (first-order phase) values, so that lp is rendered approximately 0. For crof2 to work properly, a trial spectrum must be obtained and phased to pure absorption. This spectrum provides the current rof2 and lp values for crof2. The value of the alfa delay is left constant, provided rof2 does not become less than 1 µs.

crof2 pertains to processing 2D data. Unless lp is approximately 0, fpmult affects both the dc offset and the curvature of the spectrum.

Arguments alfa specifies a value for the alfa delay before acquisition.

Related

alfa	Set alfa delay before acquisition (P)
cfpmult	Calculate first point multiplier for 2D experiments (P)
fpmult	First point multiplier for np FID data (P)
lp	First-order phase along directly detected dimension (P)
rof2	Receiver gating time following a pulse (P)

cryo_noisetest — Run Cold Probe conditioning experiments (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Runs the probe conditioning experiments and analyzes the noise using the `cnd` macro. Measures the hydrogen-induced noise and provides an efficient remedy.

Values `NOBURN` – waits the operator input period of time between tests.
No arguments – macro will prompt for a time in minutes.

cryoclient Start the CryoBay Monitor program (M, U)

Applicability Systems with Cold Probes and CryoBay Monitor software.

Description Starts the CryoBay Monitor software in a separate window. This program is a CORBA client that requires an active CORBA server running on the CryoBay PC.

See also *Cryogenic Systems Installation and Operation*

csv2cpQ Imports CSV data (M)

Syntax

Applicability VnmrJ 3.1

Description The `csv2cpQ` macro will translate a CSV (Comma Separated Values) file into actions for VnmrJ. The file name must be supplied as the first required argument. The file name may be an absolute path name or relative to userdir/data.

The CSV file used by the `csv2cpQ` macro is an ASCII text file containing, as the name implies, text values separated by commas. The first line of this file defines how the comma separated values in subsequent lines are to be interpreted. The first line can contain VnmrJ parameter names or keywords. There are no required fields. Any field that does not correspond to a VnmrJ parameter name is considered a keyword. Keywords are looked up in a synonym table to see if they should be re-interpreted as a VnmrJ parameter or value. This synonym translation file is in an appdir directory with the name `<appdir>/adm/walkupadm/csv2cpQ_synonym`.

These CSV files may be generated manually, or they may be exported from a spread-sheet. Often, the values available to the spread-sheet do not correspond directly to a VnmrJ parameter. The synonym feature allows VnmrJ to translate the spread-sheet value to something VnmrJ can use. For example, the spread-sheet might define a solvent as MeOH. The synonym table allows `csv2cpQ` to translate that into '`cd3od`'. Any field that is not a vnmr parameter or keyword will be ignored.

Arguments The actions can be submitted to an automation run, to a file for use in a future automation run, or directly to the foreground VnmrJ.

This selection is controlled by the optional argument 'auto', 'enter', or 'acq', respectively. The default is 'auto'. Another optional argument is 'print' or 'noprint'. This controls whether submission information is printed or not. The default is 'noprint'.

Examples "SAMPLE","DAY","NIGHT","solvent","operator","samplename","notebook","page", "Comments"

1,"PROTON",,"CDCl3","John","johnstuff","Johns book","p32","csv2cpQ test location 1"

2,"PROTON gCOSY","gHMBCAD","DMSO","Paul","paulstuff","Pauls book","p42","csv2cpQ test location 2"

3,"PROTON-HSQCAD",,"D2O","George","Georges book","Georges book","p23","csv2cpQ test sample 3"

4,"PROTON gHSQCAD","CARBON","DMSO","Ringo","ringostuff", "Ringos book","p38","Ringos Sample 4"

ct Completed transients (P)

Description Stores a nonuser-enterable informational parameter that changes during the course of an experiment to reflect the number of completed transients. During most experiments, an accurate transient counter is displayed in the acquisition status window, updated every five seconds.

The value of `ct` is displayed in the acquisition parameter group by the `dg` command and is only updated when data processing occurs on the FID. In an experiment that is accumulating and not processed until the acquisition is complete, `ct` always indicates `0` until the end of the acquisition.

See also *NMR Spectroscopy User Guide*

Related `dg` Display parameters of acquisition/processing group (C)

ctext Clear the text of the current experiment (C)

Description Clears the text from the current experiment text file (a block of text that may be used to describe the sample and experiment).

See also *NMR Spectroscopy User Guide*

Related `atext` Append string to the current experiment text (M)
`text` Display text or set new text for current experiment (C)

curexp Current experiment directory (P)

Description Contains the full UNIX path to the currently active experiment. This parameter is useful when accessing text files generated by various commands (e.g., `cat(curexp+'/fp.out')`).

See also *NMR Spectroscopy User Guide*

Related
systemdir	VnmrJ system directory (P)
userdir	VnmrJ user directory (P)

curscan Scan currently in progress (P)

Applicability Systems with LC-NMR accessory.

Description Keeps track of which “scan” is currently in progress. If `curscan` does not exist, the parlc macro can create it.

See also *NMR Spectroscopy User Guide*

Related
parlc	Create LC-NMR parameters (M)

curwin Current window (P)

Description An arrayed global parameter. The first value is the index of the selected window pane in the graphics window. The second value is the number of window pane rows. The third value is the number of columns.

See also *NMR Spectroscopy User Guide*

Related
fontselect	Open FontSelect window (C)
jwin	Activate current window (M)
mapwin	List of experiment numbers (P)
setgrid	Activate selected window (M)
setwin	Activate selected window (C)

cutoff Data truncation limit (P)

Description Defines the distance above and below the current vertical position vp at which spectra and integrals are truncated. By arraying `cutoff` to have two different values, the truncation limits above and below the current vertical position can be controlled independently (e.g., `cutoff=50` truncates data at vp`+50` mm and vp`–50` mm, and `cutoff=50,10` truncates data at vp`+50` mm and vp`-10` mm). `cutoff='n'` disables the action of `cutoff`.

`cutoff` is not active during interactive spectral displays (i.e., for the ds command), but is active during non-interactive spectral displays and plots (for the dss and pl commands).

Values `'n'`, number in mm.

See also *NMR Spectroscopy User Guide*

Related		
	ds	Display a spectrum (C)
	dss	Display stacked spectra (C)
	pl	Plot spectra (C)
	vp	Vertical position of spectrum (P)

cyclenoe — Set up parameters for CYCLENOE pulse sequence (M)

Applicability Systems in which the observe channel is equipped with direct synthesis rf and a linear amplifier.

Description Sets up a difference NOE experiment.

cylbr24 — Set up parameters for cycled BR24 pulse sequence (M)

Applicability Systems with solids module.

Description Sets up a BR24 sequence with quadrature detection and prepulse for solids multiple-pulse line narrowing.

See also *User Guide: Solid-State NMR*

Related br24 — Set up parameters for BR24 pulse sequence (M)

cylmrev — Set up parameters for cycled MREV8 pulse sequence (M)

Applicability Systems with a solids module.

Description Sets up a MREV8 sequence with quadrature detection and prepulse for solids multiple-pulse line narrowing.

See also *User Guide: Solid-State NMR*

Related mrev8 — Set up parameters for MREV8 pulse sequence (M)

cz — Clear integral reset points (C)

Syntax `cz<(frequency1,frequency2,...)>`

Description Removes currently defined integral reset points.

Arguments `frequency1,frequency2,...` are reset points corresponding to specified frequencies to be removed. The default is remove all reset points.

Examples
```
cz
cz(800,600,250,60)
```

See also	*NMR Spectroscopy User Guide*	
Related	`dli`	Display listed integral values (C)
	`dlni`	Display listed normalized integral values (C)
	`nli`	Find normalized integral values (C)
	`z`	Add integral reset point at the cursor position (C)

D

d0	Overhead delay between FIDs (P)
d1	First delay (P)
d2	Incremented delay in 1st indirectly detected dimension (P)
d2pul	Set up parameters for D2PUL pulse sequence (M)
d3	Incremented delay for 2nd indirectly detected dimension (P)
d4	Incremented delay for 3rd indirectly detected dimension (P)
DAC_to_G	Store gradient calibration value in DOSY sequences (P)
da	Display acquisition parameter arrays (C)
daslp	Increment for t1 dependent first-order phase correction (P)
date	Date (P)
daxis	Display horizontal LC axis (M)
Dbppste	Set up parameters for Dbppste pulse sequence (M)
Dbppsteinept	Set up parameters for Dbppsteinept pulse sequence (M)
dbsetup	Set up VnmrJ database (U)
dbupdate	Update the VnmrJ database (U)
dc	Calculate spectral drift correction (C)
dc2d	Apply drift correction to 2D spectra (C)
dcg	Drift correction group (P)
dcon	Display non interactive color intensity map (C)
dconi	Interactive 2D data display (C)
dconi	Control display selection for the dconi program (P)
dconn	Display color intensity map without screen erase (C)
dcrmv	Remove dc offsets from FIDs in special cases (P)
ddf	Display data file in current experiment (C)
ddff	Display FID file in current experiment (C)

ddfp	Display phase file in current experiment (C)
ddif	Synthesize and show DOSY plot (C)
ddrcr	Direct digital receiver coefficient ratio (P)
ddrpm	Set ddr precession mode (P)
ddrtc	Set ddr time constant (P)
dds	Default display (M)
dds_seqfil	Sequence-specific default display (M)
debug	Trace order of macro and command execution (C)
decasynctype	Select the type of decoupler asynchronous mode (P)
decay_gen	Calculates the form of diffusional attenuation expected for the measured gradient and signal maps in non-uniform gradient calibration.
deccwarnings	Control reporting of DECC warnings from PSG (P)
decomp	Decompose a VXR-style directory (M)
def_osfilt	Default value of osfilt parameter (P)
defaultdir	Default directory for Files menu system (P)
delcom	Delete a user macro (M)
delete	Delete a file, parameter directory, or FID directory (C)
delexp	Delete an experiment (M)
delexpdata	Delete data from the current experiment
deletenucleus	Removes nucleus entry to probe file (M)
dels	Delete spectra from T_1 or T_2 analysis (C)
delta	Cursor difference in directly detected dimension (P)
delta1	Cursor difference in 1st indirectly detected dimension (P)
delta2	Cursor difference in 2nd indirectly detected dimension (P)
deltaf	Difference of two time-domain cursors (P)
Dept	Set up parameters for DEPT experiment (M)
deptgl	Set up parameters for DEPTGL pulse sequence (M)
deptproc	Process array of DEPT spectra (M)
destroy	Destroy a variable or a single element of an arrayed variable (C)
destroygroup	Destroy parameters of a group in a tree (C)
df	Display a single FID (C)
df2d	Display FIDs of 2D experiment (C)
dfid	Display a single FID (C)

dfmode	Current state of display of imaginary part of a FID (P)
dfrq2	Transmitter frequency of second decoupler (P)
dfrq3	Transmitter frequency of third decoupler (P)
dfrq4	Transmitter frequency of fourth decoupler (P)
dfs	Display stacked FIDs (C)
dfsa	Display stacked FIDs automatically (C)
dfsan	Display stacked FIDs automatically without screen erase (C)
dfsh	Display stacked FIDs horizontally (C)
dfshn	Display stacked FIDs horizontally without screen erase (C)
dfsn	Display stacked FIDs without screen erase (C)
dfww	Display FIDs in whitewash mode (C)
dg	Display group of acquisition/processing parameters (C)
dg	Control dg parameter group display (P)
dg1	Display group of display parameters (M)
dg1	Control dg1 parameter group display (P)
dg2	Display group of 3rd and 4th rf channel/3D parameters (M)
dg2	Control dg2 parameter group display (P)
dga	Display group of spin simulation parameters (M)
DgcsteSL	Set up parameters for DgcsteSL pulse sequence (M)
Dgcstecosy	Set up parameters for Dgcstecosy pulse sequence (M)
Dgcstehmqc	Set up parameters for Dgcstehmqc pulse sequence (M)
dglc	Display group of LC-NMR parameters (M)
dglc	Control dglc parameter group display (P)
dglp	Control dglp parameter group of linear prediction parameters (P)
dgs	Display group of shims and automation parameters (M)
dgs	Control dgs parameter group display (P)
dhp	Decoupler high-power control with class C amplifier (P)
diagth2d	Exclude diagonal peaks when peak picking
dialog	Display a dialog box from a macro (C)
diffparam	Report differences between parameter sets (U)
diffparams	Report differences between two parameter sets (U)
diffshims	Compare two sets of shims (M,U)

digfilt	Write digitally filtered FIDs to another experiment (M)
dir	List files in directory (C)
display	Display parameters and their attributes (C)
dla	Display spin simulation parameter arrays (M)
dlalong	Long display of spin simulation parameter arrays (C)
dlC	Display LC detector trace(s) in a horizontal format.
dlCNMR	Display all forms of LC-NMR data
dli	Display list of integrals (C)
dlivast	Produce text file and process wells (M)
dll	Display listed line frequencies and intensities (C)
dlni	Display list of normalized integrals (M)
dlp	Decoupler low-power control with class C amplifier (P)
dm	Decoupler mode for first decoupler (P)
dm2	Decoupler mode for second decoupler (P)
dm3	Decoupler mode for third decoupler (P)
dm4	Decoupler mode for fourth decoupler (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmf2	Decoupler modulation frequency for second decoupler (P)
dmf3	Decoupler modulation frequency for third decoupler (P)
dmf4	Decoupler modulation frequency for fourth decoupler (P)
dmfadj	Adjust tip-angle resolution time for first decoupler (M)
dmf2adj	Adjust tip-angle resolution time for second decoupler (M)
dmf3adj	Adjust tip-angle resolution time for third decoupler (M)
dmf4adj	Adjust tip-angle resolution time for fourth decoupler (M)
dmg	Data display mode in directly detected dimension (P)
dmg1	Data display mode in 1st indirectly detected dimension (P)
dmg2	Data display mode in 2nd indirectly detected dimension (P)
dmgf	Absolute-value display of FID data or spectrum in acqi (P)
dmm	Decoupler modulation mode for first decoupler (P)
dmm2	Decoupler modulation mode for second decoupler (P)
dmm3	Decoupler modulation mode for third decoupler (P)
dmm4	Decoupler modulation mode for fourth decoupler (P)

dn	Nucleus for first decoupler (P)
dn2	Nucleus for second decoupler (P)
dn3	Nucleus for third decoupler (P)
dn4	Nucleus for fourth decoupler (P)
dndfid	Retrieve and process fid data from the locator (M)
dndjoin	Join a work space from the locator (M)
dndpar	Retrieve a parameter set from the locator (M)
dndshims	Retrieve a shimset set from the locator (M)
dnode	Display list of valid limNET nodes (M,U)
doautodialog	Start a dialog window using `def` file (M)
dodialog	Start a dialog window with `dialoglib` file (M)
dof	Frequency offset for first decoupler (P)
dof2	Frequency offset for second decoupler (P)
dof3	Frequency offset for third decoupler (P)
dof4	Frequency offset for fourth decoupler (P)
Doneshot	Set up parameters for Doneshot pulse sequence (M)
dopardialog	Start a dialog with `dialoglib/experiment def` file (M)
do_pcss	Calculate proton chemical shifts spectrum (C)
dosy	Process DOSY experiments (M)
dosy2d	Apptype macro for dosy 2D experiments (M)
dosy3Dflag	Used by the dosy macro to determine whether to use 2D or 3D DOSY processing
dosy3Dproc	Used by the dosy macro to determine whether to use 2D or 3D processing
dosybypoints	Determines whether peak picking is used by the dosy macro
dosyfit	Fits 2D or 3D DOSY data to obtain diffusion coefficients, amplitudes and statistics
dosyfrq	Larmor frequency of phase encoded nucleus in DOSY (P)
dosygamma	Gyromagnetic constant of phase encoded nucleus in DOSY (P)
dosyproc	Determines the type of processing performed by the dosy macro
dosytimecubed	Gyromagnetic constant of phase encoded nucleus in DOSY (P)
dot1	Set up a T_1 experiment (M)
dotflag	Display FID as connected dots (P)
downsamp	Downsampling factor applied after digital filtering (P)

dp	Double precision (P)
dpcon	Display plotted contours (C)
dpconn	Display plotted contours without screen erase (C)
dpf	Display peak frequencies over spectrum (C)
dpir	Display integral amplitudes below spectrum (C)
dpirn	Display normalized integral amplitudes below spectrum (M)
dpiv	Display integral amplitudes below spectrum (M)
dpirn	Display normalized integral amplitudes below spectrum (C)
dpl	Default plot (M)
dpl_seqfil	Sequence-specific default plot (M)
dplane	Display a 3D plane (M)
dpr	Default process (M)
dpr_seqfil	Sequence-specific default process (M)
dprofile	Display pulse excitation profile (M)
dproj	Display a 3D plane projection (M)
dps	Display pulse sequence (C)
dpwr	Power level for first decoupler with linear amplifier (P)
dpwr2	Power level for second decoupler with linear amplifier (P)
dpwr3	Power level for third decoupler with linear amplifier (P)
dpwr4	Power level for fourth decoupler amplifier (P)
dpwrf	First decoupler fine power (P)
dpwrf2	Second decoupler fine power (P)
dpwrf3	Third decoupler fine power (P)
dpwrm	First decoupler linear modulator power (P)
dpwrm2	Second decoupler linear modulator power (P)
dpwrm3	Third decoupler linear modulator power (P)
Dqcosy	Convert the parameter to a DQCOSY experiment (M)
draw	Draw line from current location to another location (C)
dres	Measure linewidth and digital resolution (C)
dres	Tip-angle resolution for first decoupler (P)
dres2	Tip-angle resolution for second decoupler (P)
dres3	Tip-angle resolution for third decoupler (P)

dres4	Tip-angle resolution for fourth decoupler (P)
ds	Display a spectrum (C)
ds2d	Display 2D spectra in whitewash mode (C)
ds2dn	Display 2D spectra in whitewash mode without screen erase (C)
dsnarray	Report statistical signal-to-noise for Cold Probes (M)
dscale	Display scale below spectrum or FID (C)
dscoef	Digital filter coefficients for downsampling (P)
dseq	Decoupler sequence for first decoupler (P)
dseq2	Decoupler sequence for second decoupler (P)
dseq3	Decoupler sequence for third decoupler (P)
dseq4	Decoupler sequence for fourth decoupler (P)
dsfb	Digital filter bandwidth for downsampling (P)
dshape	Display pulse shape or modulation pattern (M)
dshapef	Display last generated pulse shape (M)
dshapei	Display pulse shape or modulation pattern interactively (M)
dshim	Display a shim “method” string (M)
dslsfrq	Bandpass filter offset for downsampling (P)
dsn	Measure signal-to-noise (C)
dsnmax	Calculate maximum signal-to-noise (M)
dsplanes	Display a series of 3D planes (M)
dsptype	Type of DSP (P)
dss	Display stacked spectra (C)
dssa	Display stacked spectra automatically (C)
dssan	Display stacked spectra automatically without erasing (C)
dssh	Display stacked spectra horizontally (C)
dsshn	Display stacked spectra horizontally without erasing (C)
dssl	Label a display of stacked spectra (M)
dssn	Display stacked spectra without screen erase (C)
dsvast	Display VAST Data in a stacked 1D-NMR matrix format
dsvast2d	Display VAST data in a pseudo-2D format
dsww	Display spectra in whitewash mode (C)
dtext	Display a text file in graphics window (M)

dtrig	Delay to wait for another trigger or acquire a spectrum (P)
dutyc	Duty cycle for homodecoupling (optional) (P)

d0 Overhead delay between FIDs (P)

Description Defines the extra overhead delay at the start of each FID or array element. Overhead times between increments and transients are deterministic, i.e., both known and constant. However, the time between increments (typically `x`) is longer than the time between transients (`y`, not including times that are actually part of the pulse sequence, such as `d1`). Some experiments may benefit if it is ensured that these two times are not only constant but equal. To ensure that the times are constant and equal, insert the time `d0` at the start of each transient (before the pulse sequence actually starts); the actual delay is then `y+d0`. However, the overhead time may differ with different system configurations. To keep the `d0` delay consistent across systems, set `d0` greater than the overhead delay. The inter-FID delay `x` is then padded so that `y+d0=x+(d0-(x-y))`.

Currently, `d0` only takes into account the extra delay at the start of each array element. It does not take into account the overhead delays at the start and end of each scan. It also does not take into account delays when arraying `status` statements, shims, or spinner speeds.

The `d0` parameter does not exist in any parameter set and must be created by the user. To create `d0`, enter `create('d0','delay')`. If `d0` is nonexistent, do not insert a delay between transients.

Values `'n'`, `'y'`, or `0` to the maximum delay time (in seconds).

If `d0='n'`, the software calculates the overhead time for an array element and then delays that length of time at the beginning of subsequent transients for every array element. The calculated value of `d0` can be viewed by entering `d0='y'` in the input window.

If `d0` is set to a value, that value is the length of delay time at the beginning of subsequent transients for every array element. If the value is greater than the array overhead time, the array overhead time is padded to `d0`.

See also *User Programming*

Related `create` Create new parameter in parameter tree (C)

d1 First delay (P)

Description Length of the first delay in the standard two-pulse sequence and most other pulse sequences. This delay is used to allow recovery of magnetization back to equilibrium, if such a delay is desired.

Values 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related

`alfa`	Set `alfa` delay before acquisition (P)
`d2`	Incremented delay in 1st indirectly detected dimension (P)
`d3`	Incremented delay in 2nd indirectly detected dimension (P)
`d4`	Incremented delay in 3rd indirectly detected dimension (P)
`pad`	Preacquisition delay (P)

d2 Incremented delay in 1st indirectly detected dimension (P)

Description Length of the second delay in the standard two-pulse sequence. The delay is controlled by the parameters `ni` and `sw1` in a 2D experiment.

Values 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related

`d1`	First delay (P)
`ni`	Number of increments in 1st indirectly detected dimension (P)
`sw1`	Spectral width in 1st indirectly detected dimension (P)

d2pul Set up parameters for D2PUL pulse sequence (M)

Description Sets up a standard two-pulse sequence using the decoupler as transmitter.

See also *NMR Spectroscopy User Guide*

Related

`dhp`	Decoupler high power with class C amplifier (P)
`dn`	Nucleus for the first decoupler (P)
`dof`	Frequency offset for first decoupler (P)
`dpwr`	Power level for first decoupler with linear amplifiers (P)
`s2pul`	Set up parameters for standard two-pulse sequence (M)
`tn`	Nucleus for the observe transmitter (P)
`tof`	Frequency offset for observe transmitter (P)
`tpwr`	Power level of observe transmitter with linear amplifiers (P)

d3 Incremented delay for 2nd indirectly detected dimension (P)

Description Length of a delay controlled by the parameters `ni2` and `sw2` in a 3D experiment. The `d2` delay, which is controlled by `ni` and `sw1`, is incremented through its entire implicit array first before `d3` is incremented. To create parameters d3, ni2, phase2, and sw2 to acquire a 3D data set in the current experiment, enter addpar('3d').

Values 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d1	First delay (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
par3d	Create 3D acquisition, processing, display parameters (C)
phase2	Phase selection for 3D acquisition (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

d4 Incremented delay for 3rd indirectly detected dimension (P)

Description Length of a delay controlled by the parameters ni3 and sw3 in a 4D experiment. The d3 delay, which is controlled by ni2 and sw2, is incremented through its entire implicit array first before d4 is incremented. To create parameters d4, ni3, phase3, and sw3 to acquire a 4D data set in the current experiment, enter addpar('4d').

Values 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d1	First delay (P)
ni3	Number of increments in 3rd indirectly detected dimension (P)
par4d	Create 4D acquisition parameters (C)
phase3	Phase selection for 4D acquisition (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

DAC_to_G Store gradient calibration value in DOSY sequences (P)

Description DAG_to_G is automatically set by the setup_dosy macro by retrieving the gradient strength from the probe calibration file if probe<>'' and storing it in DAC_to_G. If probe=' ' (i.e., the probe is not defined), then DAC_to_G is set to the current value of the global parameter gcal

See also *NMR Spectroscopy User Guide.*

Related

dosy	Process DOSY experiments (M)
setup_dosy	Set up gradient levels for DOSY experiments (M)
setgcal	Set the gradient calibration constant (M)

da — Display acquisition parameter arrays (C)

Syntax `da<(par1<,par2><,par3...>)>`

Description Displays arrayed acquisition parameters.

Arguments `par1,par2,par3,...` are names of parameters to be displayed.The default is to display all such parameters.

Examples `da`
`da('d2')`

See also *NMR Spectroscopy User Guide*

Related `dg` Display parameters of acquisition/processing group (C)

daslp — Increment for t1 dependent first-order phase correction (P)

Description Causes "shearing" of f_1 traces of a 2D dataset and is used to rotate the narrow projection of some solids correlations into the f_1 dimension. Several solids experiments for Dynamic Angle Spinning (DAS) and a triple-quantum filtered 2D MAS experiment require the use of `daslp`. (Note that the command `rotate` shears two traces and is inapplicable for these experiments.)

When created, the value of `lp` for each increment of a 2D experiment is incremented by the value of `daslp` after the first Fourier transformation. The incremented phase correction is applied to the interferogram created from the coefficient table by `ft1d`, `ft2d`, `wft1d` and `wft2d`, when coefficients are present. `daslp` is also used with `ft1da`, `ft2da`, `wft1da` and `wft2da`.

Values Real values, typically similar in size to the value of parameter `lp`.

See also *NMR Spectroscopy User Guide*

Related

`ft1d`	Fourier transform along f_2 dimension (C)
`ft1da`	Fourier transform phase-sensitive data (M)
`ft2d`	Fourier transform 2D data (C)
`ft2da`	Fourier transform phase-sensitive data (M)
`lp`	First-order phase in directly detected dimension (P)
`rotate`	Rotate 2D data (C)
`wft1d`	Weight and Fourier transform f2 for 2D data (C)
`wft1da`	Weight and Fourier transform phase-sensitive data (M)
`wft2d`	Weight and Fourier transform 2D data (C)
`wft2da`	Weight and Fourier transform phase-sensitive data (M)

date — Date (P)

Description An informational parameter taken from the UNIX-level calendar (which is set by the UNIX system operator only and cannot be entered by the user). Whenever data are acquired, the date is copied from UNIX and written into the acquisition parameters, thus maintaining a record of the date of acquisition.

See also *NMR Spectroscopy User Guide*

daxis Display horizontal LC axis (M)

Applicability Systems with LC-NMR accessory.

Syntax `daxis(time,major_tic,minor_tic)`

Description Displays a horizontal LC axis. Horizontal axes are assumed to be used with “LC plots” of an entire LC run and are labeled accordingly.

Arguments `time` is the time scale, in minutes (decimal values are fine), of the axis.

`major_tic` is spacing, in minutes (decimal values are fine), of major tics.

`minor_tic` is spacing, in minutes (decimal values are fine), of minor tics.

See also *NMR Spectroscopy User Guide*

Related `paxis` Display horizontal LC axis (M)

Dbppste Set up parameters for Dbppste pulse sequence (M)

Description Converts a parameter set to Dbppste experiment; replaces the macro `bppste`.

See also *NMR Spectroscopy User Guide*

Related
`dosy` Process DOSY experiments (M)
`fiddle` Perform reference deconvolution (M)
`setup_dosy` Set up gradient levels for DOSY experiments (M)

Dbppsteinept Set up parameters for Dbppsteinept pulse sequence (M)

Description Converts a parameter set to Dbppsteinept experiment.

See also *NMR Spectroscopy User Guide*

Related
`dosy` Process DOSY experiments (M)
`fiddle` Perform reference deconvolution (M)
`setup_dosy` Set up gradient levels for DOSY experiments (M)

dbsetup — Set up VnmrJ database (U)

Syntax
dbsetup <vnmr_adm|remove|standard|imaging>
dbsetup vnmr_adm <remove|standard|imaging>
As Root:
dbsetup vnmr_adm VnmrJ_Home_dir <standard|imaging>

Arguments
vnmr_adm is the login ID of the VnmrJ system administrator.
remove only removes the data-database; does not recreate a database.
standard creates the database for standard use.
imaging creates the database for imaging spectroscopy.

Description
The UNIX script dbsetup is used during the installation of VnmrJ software and can only be run by the VnmrJ administrator (vnmr_adm) or the UNIX administrator (root). Normally it is never used again. dbsetup creates and deletes the data-database in /vnmr/pgsql/data and the user information in /vnmr/adm/users.

When run as root at least two arguments must be supplied, the login ID of the VnmrJ administrator and the VnmrJ home directory. When run as root dbsetup will delete and recreate the data-database in /vnmr/pgsql/data for all users in /vnmr/adm/users. If no user list exists yet, the list is created with the VnmrJ administrator as the only user. The mode can be specified with the third argument as 'standard' or 'imaging'; if neither is specified the mode is taken from the global file of the VnmrJ administrator. It defaults to standard. The VnmrJ administrator does not need to supply any of the arguments.

Note that additional users are created using vnmrj adm.

Examples
dbsetup
dbsetup vnmr1

See also
NMR Spectroscopy User Guide
VnmrJ Imaging NMR
VnmrJ Installation and Administration

dbupdate — Update the VnmrJ database (U)

Applicability
Systems with the VnmrJ software.

Syntax
dbsupdate stop|once [slow_ms]|forever [slow_ms]

Arguments
slow_ms is an optional argument used to slow down the database update so as not to use all of the available CPU time. slow_ms=0 is full speed. slow_ms=1000 uses about 2-5% of the CPU.
The dbupdate command is runs under nice so that any other process will be able to take the CPU away from this update anyway. The default slow_ms for forever is 1000. The default slow_ms for once is 0.

Description
A UNIX command to start and stop a program to update the VnmrJ database used by the Locator. This command might be needed at a data station to view newly acquired data. The database at the spectrometer will automatically be updated.

dc Calculate spectral drift correction (C)

Description Turns on a linear baseline correction. The beginning and end of the straight line to be used for baseline correction are determined from the display parameters sp and wp. dc applies this correction to the spectrum and stores the definition of the straight line in the parameters lvl (level) and tlt (tilt). The correction is turned off by the command cdc.

Care must be taken to ensure that a resonance does not appear too close to either end of the spectrum, or dc can produce the opposite effect from that intended; namely, it induces a sloping baseline where none was present!

See also *NMR Spectroscopy User Guide*

Related

bc	1D and 2D baseline correction (C)
cdc	Cancel drift correction (C)
dc	Drift correction group (P)
lvl	Zero-order baseline correction (P)
sp	Start of plot (P)
tlt	First-order baseline correction (P)
wp	Width of plot (P)

dc2d Apply drift correction to 2D spectra (C)

Syntax `dc2d('f1'|'f2')`

Description Computes a drift correction and applies it to each individual trace.

Arguments `'f1'` is a keyword to apply drift correction in the f_1 axis direction.

`'f2'` is a keyword to apply drift correction in the f_2 axis direction.

Examples
```
dc2d('f1')
dc2d('f2')
```

See also *NMR Spectroscopy User Guide*

Related

axis	Axis label for displays and plots (P)
bc	1D and 2D baseline correction (C)

dcg Drift correction group (P)

Description Contains the results of the dc or cdc command. This parameter cannot be set in the usual way but it can be queried by entering dcg? to determine whether drift correction is active.

Values `'dc'` indicates drift correction is active.
`'cdc'` indicates drift correction is inactive.

See also *NMR Spectroscopy User Guide*

Related
cdc Cancel drift correction (C)
dc Calculate spectral drift correction (C)

dcon Display noninteractive color intensity map (C)

Syntax `dcon<(options)>`

Description Produces a “contour plot,” actually a color intensity map, in the graphics window. The parameters sp and wp, sp1 and wp1, and sp2 and wp2 control which portion of the spectrum is displayed. The parameters sf and wf, sf1 and wf1, and sf2 and wf2 control which portion of time-domain data (FIDs and interferograms) is displayed. The parameter trace selects which dimension is displayed along the horizontal axis. The parameters sc, wc, sc2, and wc2 control where on the screen the display occurs. The parameter th is active as a threshold to black out all contours whose intensity is below th. That is, if th=7, the colors 1 to 6 are not used for the display. The parameter vs controls the vertical scale of the spectrum.

dcon displays either absolute-value mode or phase-sensitive 2D data. In av mode, data are shown in 15 different colors (starting with black), with each color representing a factor of two in intensity (a single color is used on monochrome screens). In the ph mode, the normal display of colors ranges from –6 to +6, each representing a factor of two in intensity, with the color black representing intensity 0 in the center.

Arguments options can be any of the following:

- 'linear' is a keyword to use linear instead of logarithmic increments.
- 'phcolor' is a keyword to use a phased color set with positive and negative peaks.
- 'avcolor' is a keyword to use an absolute-value color set with positive peaks. Negative contours only *cannot* be displayed, but if the data can be rephased, 180° added to rp1, and dcon('avcolor') entered again, the same thing is accomplished by inverting the phase of all peaks. Alternatively, dpcon can display negative peaks only.
- 'gray' is a keyword to use a gray scale color set.
- 'noaxis' is a keyword to omit the display outline and any horizontal or vertical axis.
- 'plot' causes the dcon display to be sent to the plotter instead of being drawn on the graphics window.

Examples
```
dcon
dcon('gray')
dcon('linear','phcolor','plot')
```

See also *NMR Spectroscopy User Guide*

Related
dconi Interactive 2D data display (C)
dconi Control display selection for the dconi program (P)

dconn	Display color intensity map without screen erase (C)
dpcon	Display plotted contours (C)
imageprint	Plot noninteractive gray scale image (M)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
sf	Start of FID (P)
sp	Start of plot (P)
sp1	Start of plot in 1st indirectly detected dimension (P)
sp2	Start of plot in 2nd indirectly detected dimension (P)
th	Threshold (P)
trace	Mode for n-dimensional data display (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)
wf	Width of FID (P)
wp	Width of plot (P)
wp1	Width of plot in 1st indirectly detected dimension (P)
wp2	Width of plot in 2nd indirectly detected dimension (P)

dconi Interactive 2D data display (C)

Syntax `dconi<(options)>`

Description Opens a 2D data display that can be interactively adjusted. The `dconi` program can accommodate any data set that can be displayed by `dcon`, `dpcon`, and `ds2d`, including 2D FIDs, interferograms, 2D spectra, planes from 3D data sets, and images. These data sets are generated by the commands `df2d`, `ft1d`, `ft2d`, and `ft3d`.

Arguments `options` can be any of the following (note that the `dconi` parameter is also available to control the `dconi` program display):

- `'dcon'` is a keyword to display a color intensity map; this is the default mode, but `'dcon'` is provided for compatibility with certain macros. If `'dcon'` is the first argument, it can be followed by any of the keywords `'linear'`, `'phcolor'`, `'avcolor'`, `'gray'`, and `'noaxis'`; all of these keywords have the same meaning as when used with `dcon`.
- `'dpcon'` is a keyword to display a true contour plot. If `'dpcon'` is the first argument, it can be followed by any of the keywords `'pos'`, `'neg'`, and `'noaxis'`, and then followed by values for `levels` and `spacing`. All of these options have the same meaning as when used with `dpcon`.

- 'ds2d' is a keyword to display a stacked plot in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind an earlier spectra). If 'ds2d' is the first argument, it can be followed by any of the keywords 'nobase', 'fill', 'fillnb', and 'noaxis'. All of these keywords have the same meaning as used with ds2d.
- 'again' is a keyword to make dconi identify which display mode is currently being used and redraw the screen in that mode.
- 'restart' is a keyword to activate dconi without redrawing the 2D data set. This action causes dconi to make sure that 2D data is already displayed.
- 'toggle' is a keyword to toggle between the cursor and box modes.
- 'trace' is a keyword to draw a trace above the spectrum.
- 'expand' is a keyword to toggle between the expand and full views of the spectrum.
- 'plot' is a keyword to plot a projection or a trace.
- 'hproj_max' is a keyword to do a horizontal projection of the maximum trace.
- 'hproj_sum' is a keyword to do a horizontal projection of the sum of all traces.
- 'vproj_max' is a keyword to do a vertical projection of the maximum trace.
- 'vproj_sum' is a keyword to do a vertical projection of the sum of all traces.

Examples

```
dconi
dconi('dcon','gray','linear')
dconi('dpcon')
```

See also *NMR Spectroscopy User Guide*

Related

boxes	Draw boxes selected by the mark command (C)
crmode	Current state of cursors in dfid, ds, or dconi (P)
dcon	Display noninteractive color intensity map (C)
dconi	Control display selection for the dconi program (P)
dconn	Display color intensity map without screen erase (C)
delta1	Cursor difference in 1st indirectly detected dimension (P)
df2d	Display FIDs of 2D experiment (C)
dpcon	Display plotted contours (C)
ds2d	Display 2D spectra in whitewash mode (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M,U)
imconi	Display 2D data in interactive gray-scale mode (M)
is	Integral scale (P)
ll2d	Automatic and interactive 2D peak picking (C)
proj	Project 2D data (C)
sf	Start of FID (P)

sp	Start of plot (P)
sp1	Start of plot in 1st indirectly detected dimension (P)
th	Threshold (P)
vs2d	Vertical scale for 2D displays (P)
vsadj	Automatic vertical scale adjustment (M)
wf	Width of FID (P)
wp	Width of plot (P)
wp1	Width of plot in 1st indirectly detected dimension (P)

dconi — Control display selection for the dconi program (P)

Description Controls the selection of the 2D display that follows entering the dconi command. Because dconi is implicitly executed by ft2d, the dconi parameter also controls the display that follows the ft2d or wft2d command.

dconi can be a string parameter in the "current" parameter set. Its syntax is similar to an argument string passed to the dconi program. For example, if dconi ='dpcon,pos,12,1.2', the dconi command displays twelve positive contours with dpcon, using a spacing of 1.2. The first component of the dconi string must be the name of the display program, such as dcon, dconn, dpcon, dpconn, ds2d, or ds2dn. Subsequent components of the string are arguments appropriate for that display program. Because the entire dconi parameter is a string, single quotes around words are not necessary and mixing words and numbers is not a problem, as the example above shows.

If the dconi parameter does not exist or is set to the null string (''), the dconi program uses its normal default. If the dconi parameter is set to a string (e.g., dconi='dcon,gray,linear' for image display), and arguments are supplied to the dconi program, (e.g., dconi('dpcon')), the supplied arguments to the command take precedence. In the case of the examples above, a contour map, not an image, is displayed.

If the dconi parameter does not exist in the current experiment, it can be created by the commands create('dconi','string')
setgroup('dconi','display')

Values '' (two single quotes) indicates that this parameter is ignored.

String 'display_program' selects the named program for 2D displays.

String 'display_program,option1,option2' selects the named program for 2D displays with options appropriate to the program.

Examples dconi='dpcon' selects contour drawing rather than default color map
dconi='dcon,gray,linear' selects image display mode.

See also *NMR Spectroscopy User Guide; VnmrJ Imaging NMR*

Related

dcon	Display noninteractive color intensity map (C)
dconi	Interactive 2D data display (C)

dconn	Display color intensity map without screen erase (C)
dpcon	Display plotted contours (C)
dpconn	Display plotted contours without screen erase (C)
ds2d	Display 2D spectra in whitewash mode (C)
ds2dn	Display 2D spectra in whitewash mode without screen erase (C)
ft2d	Fourier transform 2D data (C)
imconi	Display 2D data in interactive gray-scale mode (M)
wft2d	Weight and Fourier transform 2D data (C)

dconn — Display color intensity map without screen erase (C)

Syntax `dconn<(options)>`

Description Produces a “contour plot,” actually a color intensity map, on the screen the same as the dcon command, but without erasing the screen before starting the plot. The options available are the same as the dcon command.

See also *NMR Spectroscopy User Guide*

Related

dcon	Display noninteractive color intensity map (C)
dconi	Control display selection for the dconi program (P)

dcrmv — Remove dc offsets from FIDs in special cases (P)

Description If dcrmv exists and is set to 'y', hardware information is used to remove the dc offset from the FID providing ct=1. This only works on systems with sw less than 100 kHz. If this feature is desired for a particular experiment, create dcrmv in that experiment by entering

```
create('dcrmv','string')
setgroup('dcrmv','processing') dcrmv='y'
```

To create image parameters dcrmv, grayctr and graysl in the current experiment, enter addpar('image').

See also *NMR Spectroscopy User Guide; VnmrJ Imaging NMR*

Related

addpar	Add selected parameters to the current experiment (M)
create	Create new parameter in a parameter tree (C)
ct	Completed transients (P)
dc	Calculate spectral drift correction (C)
setgroup	Set group of a variable in a tree (C)

ddf — Display data file in current experiment (C)

Syntax `ddf<(block_number,trace_number,first_number)>`

Description Displays the file header of the data file in the current experiment. If entered with arguments, it also displays a block header and part of the data file of that block.

Arguments `block_number` is the block number. Default is 1.

`trace_number` is the trace number within the block. Default is 1.

`first_number` is the first data element number within the trace. Default is 1.

See also *User Programming*

Related `ddff` Display FID file in current experiment (C)
`ddfp` Display phase file in current experiment (C)

ddff Display FID file in current experiment (C)

Syntax `ddff<(block_number,trace_number,first_number)>`

Description Displays the file header of the FID file in the current experiment. If entered with arguments, it also displays a block header and part of the FID data of the block.

Arguments `block_number` is the block number. Default is 1.

`trace_number` is the trace number within the block. Default is 1.

`first_number` is the first data element number within the trace. Default is 1.

See also *User Programming*

Related `ddf` Display data file in current experiment (C)
`ddfp` Display phase file in current experiment (C)

ddfp Display phase file in current experiment (C)

Syntax `ddfp<(block_number,trace_number,first_number)>`

Description Displays the file header of the phase file in the current experiment. With arguments, it also display a block header and part of the phase file data of that block.

Arguments `block_number` is the block number. Default is 1.

`trace_number` is the trace number within the block. Default is 1.

`first_number` is the first data element number within the trace. Default is 1.

See also *User Programming*

Related `ddf` Display data file in current experiment (C)
`ddff` Display FID file in current experiment (C)

ddif Synthesize and show DOSY plot (C)

Syntax `ddif(<option>,lowerlimit,upperlimit)`

Description Synthesizes a 2D spectrum from 1D spectra using the information produced by the dosy macro. ddif takes the 1D spectrum and a table of diffusion data stored in the file diffusion_display.inp in the current experiment and synthesizes a 2D DOSY spectrum. It is normally run by dosy, but can be directly run, for example, to recalculate a 2D DOSY spectrum with different digitization.

Arguments option is either 'i' or 'c'.

'i' is for a display in which the 2D peak volume is proportional to 1D peak height.

'c' is for a display in which the 2D peak height equals the 1D.

lowerlimit is the lower diffusion limit (in units of 10^{-10} m^2/s).

upperlimit is the upper diffusion limit (in units of 10^{-10} m^2/s).

If arguments are not supplied, ddif defaults to showing the full range of diffusion coefficients in the file diffusion_display.inp in the current experiment. Make sure that the first increment of the DOSY data set has been transformed with the desired fn2D before using ddif. Digitization of the resultant spectrum is determined by fn2D in the spectral (F2) domain and fn1 in the diffusion (F1) domain. Make sure that the product fn2D*fn1 is not too large, or memory and processing time problems might result. Typical values are fn2D=16384 (max: 64k) and fn1=512. After dosy or ddif, 1D data is overwritten by the 2D (the dosy macro keeps a copy of the 1D data, which can be retrieved with the command undosy). Similarly, after a DOSY spectrum has been calculated, it can be retrieved with the command redosy.

See also *NMR Spectroscopy User Guide*

Related

dosy	Process DOSY experiments (M)
fn2D	Fourier number to build up 2D DOSY display in frequency domain (P)
redosy	Restore the previous 2D DOSY display from the subexperiment (M)
undosy	Restore original 1D NMR data from the subexperiment (M)

ddrcr Direct digital receiver coefficient ratio (P)

Applicability VNMRS systems and 400 - MR systems

Syntax `ddrcr=<value>`

Description Sets the filter sharpness or filter coefficient ratio. The default value of 75 is used if the parameter does not exist.

Examples

```
create('ddrce','integer')
setlimit('ddrcr',1000,2,1)
ddrcr=300
```

Values Integer values between 2 and 1000

See also *NMR Spectroscopy User Guide* and *VnmrJ User Programming*.

Related `sw` Spectral width in directly detected dimension (P)

ddrpm Set ddr precession mode (P)

Applicability VNMRS systems

Syntax `ddrpm=<'mode'>`

Values `mode` can be either of following:

Mode	*Description*
p	Pulse — default if no argument is supplied. The value is calculated as follows if `ddrpm` does not exist or `ddrpm='p'`: `ddrtc = alfa + rof2 + 2 * pw[1] / π`
e	Echo — The value is calculated as follows: `ddrtc = alfa`.

See also *VnmrJ User Programming*

Related `setrc` Set frequency referencing based upon lock signal shift (M)

`ddrtc` Set ddr precession mode (P)

ddrtc Set ddr time constant (P)

Applicability VNMRS systems

Syntax `ddrtc=<'value'>`

Description The value of `ddrtc` is set in the `setrc` macro and is determined by the `ddrpm` parameter.

A value of `ddrtc = alfa` is used by `psg` if the `ddrtc` parameter does not exist.

Values `value 0 to 1000` μsec.

See also *VnmrJ User Programming*

Related `setrc` Set frequency referencing based upon lock signal shift (M)

`setlp0` Set parameters for zero linear phase (M)

`ddrpm` Set ddr precession mode (P)

dds Default display (M)

Description Looks for sequence-specific default display macro (`dds_seqfil`) and executes if one is found. If not, the `dds` macro displays 1D, 2D, or array spectrum as the case may be.

Related
`dds_seqfil`	Sequence-specific default display (M)
`dpl`	Default plot (M)
`dpr`	Default process (M)

dds_seqfil Sequence-specific default display (M)

Description Sequence-specific default display. These macros are called by the `dds` macro.

Examples
```
dds_NOESY1D
dds_TOCSY1D
```

Related
`dds`	Default display (M)
`dpl`	Default plot (M)
`dpr`	Default process (M)

debug Trace order of macro and command execution (C)

Syntax `debug('c'|'C')`

Description Controls VnmrJ command and macro tracing. When turned on, debug displays a list of each command and macro in the shell tool from which VnmrJ was started. If VnmrJ is started when the user logs in, or if it was started from a drop-down menu or the CDE tool, the output goes to a Console window. If no Console window is present, the output goes into a file in the /var/tmp directory. This last option is not recommended. Nesting of the calls is indicated by indentation of the output. This feature is primarily a debugging tool for MAGICAL programming.

To associate the debut('c') output with a particular terminal, enter `tty`. The system responds with `/dev/pts/`*yyy*, where *yyy* is a numerical value. On the VnmrJ command line, enter `jFunc(55, '/dev/pts/`*yyy*`')`, substituting the numerical value for the *yyy*.

Arguments `'c'` is a keyword to turn on command and macro tracing.

`'C'` is a keyword to turn off command and macro tracing.

Examples
```
debug('c')
debug('C')
```

See also *User Programming*

decasynctype Decoupler asynchronous scheme (P)

Applicability VnmrJ 3.1

Description Specifies the decoupler asynchronous scheme. This flag parameter is optional, and can be used to select between different schemes to implement asynchronous decoupling. This parameter will be applicable to decoupling on all of the RF channels. If the decoupling mode (dm), 's' is selected, the decoupling is synchronous and this parameter has no effect.

Values 'p' selects the "progressive offset" scheme, which is the default. This simulates a free running decoupler modulation with respect to the acquisition window.

'b' selects the "bit reversal" scheme. This scheme uses the bit reversal algorithm to implement asynchronous decoupling. It attempts to efficiently sample various phases of the decoupling cycle and hence may be more appropriate when number of transients (nt) is a small number that is a power of two.

'r' selects a random scheme for implementing asynchronous decoupling.

decay_gen Calculates the form of diffusional attenuation expected for the measured gradient and signal maps in non-uniform gradient calibration.

Syntax `decay_gen(D,ngrads)`

Applicability VnmrJ 3.1

Description `decay_gen` takes the measured signal profile and gradient map as a function of position and calculates the predicted signal attenuation as a function of gradient strength.

Arguments `decay_gen` takes two arguments: the diffusion coefficient (D) of the calibrant, and the number of gradient levels (`ngrads`) for which the attenuation is to be calculated. `decay_gen` is normally run only by the `nugcalib` macro.

See also `nugcalib`
`gradfit`
`powerfit`

deccwarnings Control reporting of DECC warnings from PSG (P)

Applicability Systems with DECC (Digital Eddy Current Compensation) boards for gradient compensation.

Description A global parameter that controls whether PSG will warn the user when the ECC corrections are large enough that they could exceed the capabilities of the DECC board. By default, this parameter does not exist, and a warning is printed whenever an experiment is started if the ECC amplitudes are possibly too large. The warning does indicate a definite be a problem, only that not enough ECC drive capability is available to compensate for an instantaneous gradient swing from minus the maximum gradient strength to the maximum positive gradient.

To disable the warnings, create this global string parameter and set it to 'n'.

Values 'n' or 'N' to suppress warnings. If the value starts with any other character, the normal warnings are printed.

decomp Decompose a VXR-style directory (M)

Syntax `decomp<(VXR_file)>`

Description Takes a library, as loaded from a VXR-style system (VXR, XL, or Gemini), and extracts each entry into a separate UNIX file. The file can be obtained from a magnetic tape or over limNET. `decomp` creates a UNIX subdirectory in the current working directory and uses that to write each entry as a UNIX file. The name of the UNIX subdirectory is derived from the library name.

Arguments `VXR_file` is the name of the original file. It must have an extension in the form `.NNN`, where `NNN` is the number of entries in the original library. A limit of 432 entries is imposed.

See also *NMR Spectroscopy User Guide*

Related `convert` Convert data set from a VXR-style system (C,U)

def_osfilt Default value of osfilt parameter (P)

Description A global parameter that establishes the default type of digital filter, Analog*Plus™* or brickwall, when DSP is configured. The *actual* filter used in any experiment is set by the local parameter `osfilt`. Usually, `def_osfilt` is set to the value for normal use, and then `osfilt` is changed within a given experiment if different filter characteristics are desired.

Values `'a'` or `'A'` for the Analog*Plus* digital filter. This filter is flatter in the passband and drops off somewhat more sharply than analog filters.

`'b'` or `'B'` for the brickwall digital filter. This filter is extremely flat across the passband and drops off sharply on the edge; however, the enhanced filtering comes at the expense of somewhat reduced baseline performance.

See also *NMR Spectroscopy User Guide*

Related `dsp` Type of DSP for data acquisition (P)
`osfilt` Oversampling filter for real-time DSP (P)

defaultdir Default directory for Files menu system (P)

Description Stores the name to the default directory for use with the Directory Menu in the Files menu system. Initial value for `defaultdir` is the home or login directory of the user. Selecting the Default button in the Directory Menu sets the current directory to the value of `defaultdir`. The opposite action, setting the value of `defaultdir` to the current directory, occurs when the Set Default button in the Directory Menu is selected. If the entry for a directory is marked and the Set Default button is selected, the directory marked becomes the new value of `defaultdir`.

See also *NMR Spectroscopy User Guide*

delcom Delete a user macro (M)

Syntax `delcom(file)`

Description Deletes a macro file in a user's macro library (`maclib`). Note that `delcom` will not delete a macro in the VnmrJ system macro library.

Arguments `file` is the file name of the user's macro to be deleted.

Examples `delcom('lds')`

See also *User Programming*

Related `crcom` Create user macro without using a text editor (C)
`macrorm` Remove a user macro (C)

delete Delete a file, parameter directory, or FID directory (C)

Syntax `delete(file1<,file2,...>)`

Description Delete files and directories in a somewhat safer manner than the `rm` command. Using `rm` is not recommended in VnmrJ because `rm` allows wildcard characters (* and ?) in the file description and recursive file deletion with the `-r` option. The `delete` command does not allow wildcard characters or the `-r` option, but you can still use the `delete` command to delete a file as well as remove `.fid` and `.par` directories, normally the only directories that need to be removed (experiment directories are deleted with the `delexp` macro).

Arguments `file1, file2, ...` are the names of one or more files or directories to be deleted. When the `delete` command is entered, it first searches for `file1`. If it finds that file and it is not a directory, `file1` is deleted. If `file1` is not found, `.fid` is appended to the file name and `delete` searches for the file in that `.fid` directory. If the file is found, it is removed; otherwise, `.par` is appended to the file name and `delete` searches for the file in that `.par` directory. If the file is found, it is removed; otherwise, the command takes no action and continues to the next file name. The process is repeated for each file name given as an argument.

Examples `delete('/home/vnmr1/memo')`
`delete('/vnmr/fidlib/fid1d')`

See also *NMR Spectroscopy User Guide*

Related		
	`delexp`	Delete an experiment (M)
	`rm`	Delete file (C)
	`rmdir`	Remove directory (C)

`delexp` Delete an experiment (M)

Syntax `delexp(experiment_number)`

Description Deletes an experiment.

Arguments `experiment_number` is the number (from 2 through 9999) of the experiment to be deleted (experiment 1 cannot be deleted). `delexp` also deletes the corresponding `jexp`XXX macro if necessary.

Examples `delexp(321)`

See also *NMR Spectroscopy User Guide*

Related		
	`cexp`	Create an experiment (M)
	`jexp`	Join existing experiment (C)

`delexpdata` Delete data from the current experiment

Syntax `delexpdata`

Applicability VnmrJ 3.1

Description The `delexpdata` command will remove data from the current experiment. It will delete 3D data, if present. This command will not execute if an acquisition is active or queued in the current experiment.

`deletenucleus` Removes nucleus entry from current probe file (M)

Applicability ALL

Description All lines for the specified nucleus are removed from the current probe file. The argument should correspond to an entry in the probe file.

Syntax `deletenucleus('nucleus')`

Arguments `nucleus` — name followed by atomic number, e.g. `c13` not 13C.

Examples `deletenucleus('Si29')`

Related
`addnucleus` Adds nucleus entry to probe file (M)
`addprobe` Create new probe directory and probe file (M)

dels Delete spectra from T_1 or T_2 analysis (C)

Syntax `dels(index1<,index2,...>)`

Description Deletes the spectra selected from the file `fp.out` (the output file of `fp`) used by the `t1` or `t2` analysis. Spectra may be restored by rerunning `fp`.

Arguments `index1,index2,...` are the indexes of the spectra to be deleted.

Examples
`dels(7)`
`dels(2,5)`

See also *NMR Spectroscopy User Guide*

Related
`dll` Display listed line frequencies and intensities (C)
`fp` Find peak heights or phases (C)
`getll` Get frequency and intensity of a line (C)
`t1` T_1 exponential analysis (M)
`t2` T_2 exponential analysis (M)

delta Cursor difference in directly detected dimension (P)

Description Difference between two frequency cursors along the directly detected dimension. The value is changed by moving the right cursor, relative to the left, in the `ds` or `dconi` display.

Values Positive number, in Hz.

See also *NMR Spectroscopy User Guide*

Related
`dconi` Interactive 2D data display (C)
`delta1` Cursor difference in 1st indirectly detected dimension (P)
`delta2` Cursor difference in 2nd indirectly detected dimension (P)
`ds` Display a spectrum (C)
`split` Split difference between two cursors (M)

delta1 Cursor difference in 1st indirectly detected dimension (P)

Description Difference of two frequency cursors along the first indirectly detected dimension. Analogous to the `delta` parameter except that `delta1` applies to the first indirectly detected dimension of a multidimensional data set.

Values Positive number, in Hz.

See also *NMR Spectroscopy User Guide*

Related `delta`	Cursor difference in directly detected dimension (P)

`delta2` Cursor difference in 2nd indirectly detected dimension (P)

Description Difference of two frequency cursors along the second indirectly detected dimension. Analogous to the `delta` parameter except that `delta2` applies to the second indirectly detected dimension of a multidimensional data set.

Values Positive number, in Hz.

See also *NMR Spectroscopy User Guide*

Related `delta`	Cursor difference in directly detected dimension (P)

`deltaf` Difference of two time-domain cursors (P)

Description Difference between the two time-domain cursors of the `df` (or `dfid`) display. To create this parameter and the other FID display parameters `axisf`, `dotflag`, `vpf`, `vpfi`, and `crf` (if the parameter set is older and lacks these parameters), enter `addpar('fid')`.

Values Number, in seconds.

See also *NMR Spectroscopy User Guide*

Related `addpar`	Add selected parameters to the current experiment (M)
`crf`	Current time-domain cursor position (P)
`df`	Display a single FID (C)
`dfid`	Display a single FID (C)

`dept` Set up parameters for DEPT experiment (M)

Description Set up parameters for DEPT experiment

See also *NMR Spectroscopy User Guide*

Related `adept`	Automatic DEPT analysis and spectrum editing (C)
`autodept`	Automated complete analysis of DEPT data (M)
`deptgl`	Set up parameters for DEPTGL pulse sequence (M)
`deptproc`	Process array of DEPT spectra (M)
`padept`	Plot automatic DEPT analysis (C)
`ppcal`	Proton decoupler pulse calibration (M)

deptgl Set up parameters for DEPTGL pulse sequence (M)

Description Macro for the DEPTGL pulse sequence for spectral editing and polarization transfer experiments.

See also *NMR Spectroscopy User Guide*

Related `Dept` Set up parameters for DEPT pulse sequence (M)

deptproc Process array of DEPT spectra (M)

Description Automatically processes arrays of DEPT-type spectra. The FIDs are transformed (using `lb=2.5`), phased, and scaled. In foreground operation, a stacked display is produced. By default, an automatic DEPT analysis (`adept`) is performed.

See also *NMR Spectroscopy User Guide*

Related

`adept`	Automatically edit DEPT spectra (C)
`Dept`	Set up parameters for DEPT experiment
`lb`	Line broadening along the directly detected dimension (P)
`pldept`	Plot DEPT type spectra (M)
`procplot`	Automatically process FIDs (M)

destroy Destroy a variable or single element of an arrayed variable (C)

Syntax

```
destroy(variable<,tree>)
destroy(name[,tree])
destroy(name[,tree]):$ok
```

tree can be:

`current,global,processed,systemglobal, usertree`

`destroy(name,tree,index)`

Description Removes a variable from one of the variable trees. If the destroyed variable was an array, the `array` variable is automatically updated.

If destroy is called for a non-existent parameter, the command will abort with a message. If an optional return value is given, it will indicate success (1) or failure (0) and the command will not abort.

Arguments `variable` is the name of the variable to be destroyed.

`tree` is a keyword for the type of variable tree: `'global'`, `'current'`, `'processed'`, `'usertree'`, or `'systemglobal'`. The default is 'current'. Refer to the create command for more information on types of trees.

The first argument (names) can be a list of space separated names. When using a list of names, the destroy command will abort only if none of the parameters in the list exist. The destroy command will update the array parameter if needed.

An optional third argument specifies the index of an arrayed variable to be destroyed. The tree argument must be supplied if the index is used. An error is given if the specfied index is larger than the number of array elements. If a variable has only one element and the index 1 is given, the entire variable is destroyed. When used with the array index, the destroy command will not update the array parameter.

Examples
```
destroy('a')
destroy('c','global')
destroy('par1 par2 par3')
destroy('acqstatus','current',3)
```

See also *User Programming*

Related

array	variable order and precedence (P)
create	Create new variable in a variable tree (C)
display	Display variables and their attributes (C)
paramvi	Edit a variable and its attributes using vi text editor (C)
prune	Prune extra variables from current tree (C)

destroygroup Destroy parameters of a group in a tree (C)

Syntax `destroygroup(group<,tree>)`

Description Removes parameters of a group from one of the parameters trees.

Arguments `group` is a keyword for the type of parameter group: 'all', 'sample', 'acquisition', 'processing', 'display', or 'spin'.

`tree` is a keyword for the type of parameter tree: 'global', 'current', or 'processed'. The default is 'current'. Refer to the create command for more information on trees.

Examples
```
destroygroup('sample')
destroygroup('all','global')
```

See also *User Programming*

Related

create	Create new parameter in a parameter tree (C)
destroy	Destroy a parameter (C)
display	Display parameters and their attributes (C)
groupcopy	Copy parameters of group from one tree to another (C)
setgroup	Set group of a variable in a tree (C)

df Display a single FID (C)

Syntax
```
df<(index)>
df(options)
df('fidshim')
```

Description
Displays a single FID. Parameter entry after an FID has been displayed causes the display to be updated. The FID is left-shifted by the number of complex data points specified by the parameter `lsfid`. The FID is also phase-rotated (zero-order only) by the number of degrees specified by the parameter `phfid`. Left shifting and phasing can be avoided by setting `lsfid` and `phfid` to `'n'`. `df` is identical in function to the `dfid` command.

Arguments
`index` (used with syntax 1) is the number of a particular FID for arrayed 1D experiments or for 2D experiments. Default is 1.

`options` (used with syntax 2) is any of the following:

- `'toggle'` is a keyword to switch between box and cursor modes.
- `'restart'` is a keyword to redraw the cursor if it has been turned off.
- `'expand'` is a keyword to switch between expanded and full views of the FID.
- `'imaginary'` is a keyword to switch on and off the display of the imaginary FID.
- `'sfwf'` is a keyword to interactively adjust the start and width of the FID display.
- `'phase'` is a keyword to enter an interactive phasing mode.
- `'dscale'` is a keyword to toggle the scale below the FID on and off.
- `df('fidshim')` sets the global parameter `fidarea` to one of two values, depending on whether the parameter `displaymode` contains an `r`, for real mode.

Examples
```
df
df(4)
df('restart')
```

See also *NMR Spectroscopy User Guide*

Related

`crmode`	Current state of cursors in `dfid`, `ds`, or `dconi` (P)
`dfid`	Display a single FID (C)
`df2d`	Display FIDs of 2D experiment (C)
`dfmode`	Current state of display of imaginary part of a FID (P)
`lsfid`	Number of complex points to left-shift the `np` FID (P)
`phfid`	Zero-order phasing constant for the `np` FID (P)

df2d Display FIDs of 2D experiment (C)

Syntax `df2d<(<'nf',><array_index>)>`

Description Produces a color intensity map of the raw 2D FIDs as a function of t_1 and t_2. The display can be modified by subsequent display commands, for example, `df2d dconn` will display the 2D FIDs without clearing the graphics screen.

Arguments `'nf'` is a keyword specifying that the data has been collected in the compressed form using `nf`. In other words, each array element is collected as one 2D FID or image comprised of `nf` FIDs or traces.

`array_index` is the index of the array to be displayed.

Examples `df2d`
`df2d(1)`

See also *NMR Spectroscopy User Guide*

Related `dconi` Interactive 2D data display (C)
`df` Display a single FID (C)

dfid Display a single FID (C)

Syntax (1) `dfid<(index)>`
(2) `dfid<(options)>`

Description Functions the same as the `df` command. See `df` for information.

See also *NMR Spectroscopy User Guide*

Related `df` Display a single FID (C)

dfmode Current state of display of imaginary part of a FID (P)

Description Holds a string variable that reflects the state of display of the imaginary part of a FID. `dfmode` is primarily used by the programmable menu `dfid` to determine the status of the display of the imaginary part of a FID.

Values `'r'` indicates the current display is real only.

`'i'` indicates the current display is imaginary.

`'z'` indicates the display is zero imaginary.

See also *User Programming*

dfrq — Transmitter frequency of first decoupler (P)

Description: Contains the transmitter frequency for the first decoupler. dfrq is automatically set when the parameter dn is changed and should not be necessary for the user to manually set.

Values: Frequency, in MHz. The value is limited by synthesizer used with the channel.

See also: *NMR Spectroscopy User Guide*

Related:

dfrq2	Transmitter frequency of second decoupler (P)
dfrq3	Transmitter frequency of third decoupler (P)
dfrq4	Transmitter frequency of fourth decoupler (P)
dn	Nucleus for first decoupler (P)
dof	Frequency offset for first decoupler (P)
sfrq	Transmitter frequency of observe nucleus (P)
spcfrq	Display frequencies of rf channels (M)

dfrq2 — Transmitter frequency of second decoupler (P)

Applicability: Systems with a second decoupler.

Description: Contains the transmitter frequency for the second decoupler. dfrq2 is automatically set when parameter dn2 is changed and should not be necessary for the user to manually set.

Values: Frequency, in MHz. Value is limited by synthesizer used with the channel. If dn2='' (two single quotes with no space in between) and a second decoupler is present in the console, dfrq2 is internally set to 1 MHz.

See also: *NMR Spectroscopy User Guide*

Related:

dn2	Nucleus for second decoupler (P)
dof2	Frequency offset for second decoupler (P)

dfrq3 — Transmitter frequency of third decoupler (P)

Applicability: Systems with a third decoupler.

Description: Contains the transmitter frequency for the third decoupler. dfrq3 is automatically set when the parameter dn3 is changed and should not be necessary for the user to manually set.

Values: Frequency, in MHz. Value is limited by synthesizer used with the channel. If dn3='' (two single quotes with no space in between) and a third decoupler is present in the console, dfrq3 is internally set to 1 MHz.

See also *NMR Spectroscopy User Guide*

Related	dn3	Nucleus for third decoupler (P)
	dof3	Frequency offset for third decoupler (P)

dfrq4 Transmitter frequency of fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Contains the transmitter frequency for the fourth decoupler. `dfrq4` is automatically set when the parameter `dn4` is changed and should not be necessary for the user to manually set.

Values Frequency, in MHz. Value is limited by a synthesizer used with the channel. If `dn4=''` (two single quotes with no space in between) and a fourth decoupler is present in the console, `dfrq4` is internally set to 1 MHz.

See also *NMR Spectroscopy User Guide*

Related	dn4	Nucleus for fourth decoupler (P)
	dof4	Frequency offset for fourth decoupler (P)
	spcfrq	Display frequencies of rf channels (M)
	rftype	type of rf generation

dfs Display stacked FIDs (C)

Syntax `dfs<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Displays one or more FIDs. The position of the first FIDs is governed by the parameters `wc`, `sc`, and `vpf`. A subsequent FID is positioned relative to the preceding FID by the parameters `vo` and `ho`.

Arguments `start` is the index number of the first FID for multiple FIDs. It can also be the index number of a particular FID for arrayed 1D or 2D data sets.

`finish` is the index number of the last FID for multiple FIDs. To include all FIDs, set `start` to 1 and `finish` to `arraydim` (see example below).

`step` is the increment for the FID index. The default is 1.

`'all'` is a keyword to display all of the FIDs. This is the default.

`'imag'` is a keyword to display only the imaginary FID channel.

`color` is the color of the display: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, or `'white'`.

Examples
```
dfs(1,arraydim,3)
dfs('imag')
```

See also *NMR Spectroscopy User Guide*

Related
arraydim Dimension of experiment (P)
dfsa Display stacked FIDs automatically (C)
dfsan Display stacked FIDs automatically without screen erase (C)
dfsh Display stacked FIDs horizontally (C)
dfshn Display stacked FIDs horizontally without screen erase (C)
dfsn Display stacked FIDs without screen erase (C)
dfww Display FIDs in whitewash mode (C)
ho Horizontal offset (P)
plfid Plot FID (C)
pfww Plot FIDs in whitewash mode (C)
sc Start of chart (P)
vo Vertical offset (P)
vpf Current vertical position of FID (P)
wc Width of chart (P)

dfsa Display stacked FIDs automatically (C)

Syntax `dfsa<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Displays one or more FIDs automatically by adjusting the parameters vo and ho to fill the screen in a lower left to upper right presentation (wc must be set to less than full screen width for this to work). The position of the first FID is governed by parameters wc, sc, and vpf.

Arguments `start` is the index number of the first FID for multiple FIDs. It can also be the index number of a particular FID for arrayed 1D or 2D data sets.

`finish` is the index number of the last FID for multiple FIDs.

`step` is the increment for the FID index. The default is 1.

`'all'` is a keyword to display all of the FIDs. This is the default.

`'imag'` is a keyword to display only the imaginary FID channel.

`color` is the color of the display: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, or `'white'`.

See also *NMR Spectroscopy User Guide*

Related
dfs Display stacked FIDs (C)
dfsan Display stacked FIDs automatically without screen erase (C)

dfsan Display stacked FIDs automatically without screen erase (C)

Syntax `dfsan<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Functions the same as the command dfsa except the graphics window is not erased before starting the display. This allows composite displays of many FIDs to be created. The arguments are the same as dfsa.

See also *NMR Spectroscopy User Guide*

Related dfsa Display stacked FIDs automatically (C)

dfsh Display stacked FIDs horizontally (C)

Syntax `dfsh<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Displays one or more FIDs horizontally by setting vo to zero and adjusting ho, sc, and wc to fill the screen from left to right with the entire array. The position of the first FID is governed by parameters wc, sc, and vpf.

Arguments `start` is the index number of the first FID for multiple FIDs. It can also be the index number of a particular FID for arrayed 1D or 2D data sets.

`finish` is the index number of the last FID for multiple FIDs. To display all FIDs, set `finish` to the parameter arraydim.

`step` is the increment for the FID index. The default is 1.

`'all'` is a keyword to display all of the FIDs. This is the default.

`'imag'` is a keyword to display only the imaginary FID channel.

`color` is the color of the display: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, or `'white'`.

See also *NMR Spectroscopy User Guide*

Related dfs Display stacked FIDs (C)
dfshn Display stacked FIDs horizontally without screen erase (C)

dfshn Display stacked FIDs horizontally without screen erase (C)

Syntax `dfshn<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Functions the same as the command dfsh except the graphics window is not erased before starting the display. This allows composite displays of many FIDs to be created. The arguments are the same as dfsh.

See also *NMR Spectroscopy User Guide*

Related dfsh Display stacked FIDs horizontally (C)

dfsn — Display stacked FIDs without screen erase (C)

Syntax `dfsn<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Functions the same as the command dfs except the graphics window is not erased before starting the display. This allows composite displays of many FIDs to be created. The arguments are the same as dfs.

See also *NMR Spectroscopy User Guide*

Related

dfs	Display stacked FIDs (C)

dfww — Display FIDs in whitewash mode (C)

Syntax `dfww<(<start><,finish><,step><,'all'|'imag'><,color>)>`

Description Displays FIDs in whitewash mode (after the first FID, each FID is blanked out in regions in which it is behind an earlier FID). The position of the first FIDs is governed by parameters wc, sc, and vpf.

Arguments `start` is the index number of the first FID for multiple FIDs. It can also be the index number of a particular FID for arrayed 1D or 2D data sets.

`finish` is the index number of the last FID for multiple FIDs.

`step` is the increment for the FID index. The default is 1.

`'all'` is a keyword to display all of the FIDs. This is the default.

`'imag'` is a keyword to display only the imaginary FID channel.

`color` is the color of the display: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, or `'white'`.

See also *NMR Spectroscopy User Guide*

Related

dfs	Display stacked FIDs (C)
pfww	Plot FIDs in whitewash mode (C)

dg — Display group of acquisition/processing parameters (C)

Syntax `dg('template',<'file_name'>)`

Description Displays the group of acquisition and 1D/2D processing parameters. To display an individual parameter, enter the name of the parameter followed by a question mark (e.g., `sw?`). Parameters do not have to be displayed in order to be entered or changed. The `dg` display is controlled by the string parameter dg.

Arguments `template` is the name of the template parameter. The default is `'dg'`. See the manual *User Programming* for rules on constructing a template. The macros dg dg1, dg2, dglp, and dgs activate `dg` with a `template` argument such as `'dg'`,`'dg1'`, `'dg2'`, `'dglp'`, `'dgs'`, etc. or a user defined template.

`file_name` is the name of the file to which the `dg` command will write the parameters specified by `template`.

Examples `dg`
`dg('dgexp')`
`dg('dg','dgout')`

See also *NMR Spectroscopy User Guide; User Programming*

Related

`?`	Display the value of an individual parameter (C)
`da`	Display acquisition parameter arrays (C)
`dglp`	Display group of linear prediction parameters (C)
`da`	Display acquisition parameter arrays (P)
`dg`	Control `dg` parameter group display (P)
`dglp`	Control `dglp` parameter group of linear prediction parameters (P)
`dg1`	Display group of display parameters (M)
`dg2`	Display group of 3rd and 4th rf channel/3D parameters (M)
`dgs`	Display group of special/automation parameters (M)

dg — Control dg parameter group display (P)

Description Controls the display of the `dg` command for the group of acquisition and 1D/2D processing parameters. `dg`, a string parameter, can be modified with the command `paramvi('dg')`.

See also *NMR Spectroscopy User Guide*

Related

`dg`	Display group of acquisition/processing parameters (C)
`paramvi`	Edit a parameter and its attributes with `vi` text editor (C)

dg1 — Display group of display parameters (M)

Description Displays the group of display parameters. To display an individual parameter, enter the name of the parameter followed by a question mark (e.g., `sp?`). Parameters do not have to be displayed in order to be entered or changed. The `dg1` display is controlled by the string parameter `dg1`.

See also *NMR Spectroscopy User Guide*

Related

`?`	Display individual parameter value (C)
`dg1`	Control `dg1` parameter group display (P)
`dg`	Display group of acquisition/processing parameters (C)

dg1 — Control dg1 parameter group display (P)

Description: Controls the display of the dg1 command for the group of display parameters. dg1, a string parameter, can be modified with paramvi('dg1').

See also: *NMR Spectroscopy User Guide*

Related:

dg1	Display group of display parameters (M)
paramvi	Edit a parameter and its attributes with *vi* text editor (C)

dg2 — Display group of 3rd and 4th rf channel/3D parameters (M)

Description: Displays the group of acquisition parameters associated with a second decoupler channel on a system with a third rf channel. It also displays the group of parameters associated with selective 2D processing of 3D data sets. To display an individual parameter, enter the name of the parameter followed by a question mark (e.g., sw?*)*. Parameters do not have to be displayed in order to be entered or changed. The dg2 display is controlled by the string parameter dg2.

See also: *NMR Spectroscopy User Guide*

Related:

dg	Display group of acquisition/processing parameters (C)
dg2	Control dg2 parameter group display (P)

dg2 — Control dg2 parameter group display (P)

Description: Controls the display of the dg2 command for the group of 3rd and 4th rf channel/3D parameters. dg2, a string parameter, can be modified with the command paramvi('dg2'). To retrieve the dg2 and ap display templates for the current experiment, enter addpar('3rf').

See also: *NMR Spectroscopy User Guide*

Related:

addpar	Add selected parameters to the current experiment (M)
dg2	Display group of 3rd and 4th rf channel/3D parameters (M)
paramvi	Edit a parameter and its attributes with vi text editor (M)

dga — Display group of spin simulation parameters (M)

Description: Displays the file of spin simulation parameters (Group A). There is one such group of parameters in the data system, not one per experiment as with normal NMR parameters.

See also *NMR Spectroscopy User Guide*

Related	dg	Display group of acquisition/processing parameters (C)
	dla	Display spin simulation parameter arrays (C)

dgcsteSL Set up parameters for DgcsteSL pulse sequence (M)

Description Converts a parameter set to DgcsteSL experiment.

See also *NMR Spectroscopy User Guide*

Related	dosy	Process DOSY experiments (M)
	fiddle	Perform reference deconvolution (M)
	setup_dosy	Set up gradient levels for DOSY experiments (M)

dgcstecosy Set up parameters for Dgcstecosy pulse sequence (M)

Description Converts a parameter set to Dgcstecosy experiment

See also *NMR Spectroscopy User Guide*

Related	dosy	Process DOSY experiments (M)
	makeslice	Synthesize 2D projection of a 3D DOSY spectrum (C)
	setup_dosy	Set up gradient levels for DOSY experiments (M)
	showoriginal	Restore first 2D spectrum in 3D DOSY spectrum (M)

dgcstehmqc Set up parameters for Dgcstehmqc pulse sequence (M)

Description Converts a parameter set to Dgcstehmqc experiment

See also *NMR Spectroscopy User Guide*

Related	dosy	Process DOSY experiments (M)
	makeslice	Synthesize 2D projection of 3D DOSY spectrum (C)
	setup_dosy	Set up gradient levels for DOSY experiments (M)
	showoriginal	Restore first 2D spectrum in 3D DOSY spectrum (M)

dglc Display group of LC-NMR parameters (M)

Applicability Systems with LC-NMR accessory.

Description Displays parameters related to LC-NMR on a separate screen. This macro is equivalent to the command `dg('dglc')`.

See also *NMR Spectroscopy User Guide*

Related `dglc` Control LC-NMR parameter display (P)

dglc Control dglc parameter group display (P)

Applicability Systems with LC-NMR accessory.

Description Controls the display of the LC-NMR parameters by the macro `dglc` and the equivalent command `dg('dglc')`. If this parameter does not exist, the `parlc` macro can create it.

See also *NMR Spectroscopy User Guide*

Related
`dglc` Display LC-NMR parameters (M)
`parlc` Create LC-NMR parameters (M)

dglp Display group of linear prediction parameters (C)

Syntax `dglp`

Description Displays the linear prediction parameters group. Parameters do not have to be displayed in order to be entered or changed. The `dglp` display is controlled by the string parameter `dglp`.

Examples `dglp`

See also *NMR Spectroscopy User Guide; User Programming*

Related `dg` Control `dg` parameter group display (P)

dgs Display group of shims and automation parameters (M)

Description Displays the group of shims and automation parameters. To display an individual parameter, enter name of the parameter followed by a question mark (e.g., `sw?`). Parameters do not have to be displayed in order to be entered or changed. The `dgs` display is controlled by the parameter `dgs`.

See also *NMR Spectroscopy User Guide*

Related
`dg` Display group of acquisition/processing parameters (C)
`dgs` Control `dgs` parameter group display (P)

dgs — Control dgs parameter group display (P)

Description: Controls display of the `dgs` command for the group of shims and automation parameters. `dgs`, a string parameter, can be modified by `paramvi('dgs')`.

See also: *NMR Spectroscopy User Guide*

Related:

`dgs`	Display group of special/automation parameters (M)
`paramvi`	Edit a parameter and its attributes with `vi` text editor (C)

dhp — Decoupler high-power control with class C amplifier (P)

Applicability: System with a class C amplifier.

Description: `dhp` selects a decoupler high-power level for systems with class C amplifiers on the decoupler channel. Specific values of `dhp` should be calibrated periodically for any particular instrument and probe combination. As a rough guide, `dhp=75` corresponds to approximately 2 watts at 200 MHz.

CAUTION Decoupler power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate high-power decoupling to avoid exceeding 2 watts of power.

For systems equipped with a linear amplifier on the decoupler channel, `dhp` is nonfunctional and is replaced by the parameter `dpwr`.

Note that `dhp` runs in the opposite direction from `dlp` (i.e., for `dhp` a higher number means more power, for `dlp` a higher number means less power).

Values: 0 to 255 (where 255 is maximum power) in uncalibrated, non-linear units.

`'n'` selects low-power decoupling under the control of the parameter `dlp`.

See also: *NMR Spectroscopy User Guide*

Related:

`dlp`	Decoupler low power with class C amplifier (P)
`dpwr`	Power level for first decoupler with linear amplifier (P)
`tn`	Nucleus for observe transmitter (P)

diagth2d — Exclude diagonal peaks when peak picking

Applicability: VnmrJ 3.1

Description This parameter is used by ll2d to exclude diagonal peaks when peak picking. Peaks within diagth2d Hertz of the diagonal will not be picked by ll2d. Setting diagth2d to 0.0 will cause ll2d to pick all peaks including diagonal peaks.

Related ll2d

dialog Display a dialog box from a macro (C)

Syntax `dialog(definition_file,output_file<,'nowait'>)`

Description Opens a dialog box from a macro. The output is written to a file that can be read by the macro using the lookup command.

Arguments `definition_file` is the name of the file (specified by an absolute path) that defines the layout of the dialog box.

`output_file` is the name of the file (specified by an absolute path) where the results of the dialog box are written.

`'nowait'` is a keyword to return immediately, without waiting for input into the dialog box.

Examples `dialog(userdir+'/dialoglib/array,'/tmp/array')`

See also *User Programming*

Related lookup Look up words and lines from a text file (C)

diffparam Report differences between parameter sets (UNIX)

Syntax `diffparam file1 file2 <parametergroup>`

Applicability VnmrJ 3.1

Description Reports differences between VNMR parameter sets, based on the output of the listparam command.

Arguments file1 and file2 are VNMR parameter files, like

```
$HOME/vnmrsys/exp1/procpar
$HOME/vnmrsys/exp1/curpar
$HOME/vnmrsys/global
/vnmr/conpar
xyz.fid/procpar
```

file1 and file2 can also be directories (xyz.fid or xyz.par, or a local experiment like ~/vnmrsys/exp1); in this case diffpar will look for a subfile procpar in these directories. parametergroup is an optional argument that permits specifying the parameter type. By default, only acquisition parameters are compared. The following options exist (only the first two characters are relevant):

- acquisition - compare acquisition parameters (default)
- processing - compare processing parameters only
- display - compare display parameters only

- spsim - compare spin simulation parameters only
- sample - compare sample parameters only
- all - compare ALL parameters (output indicates group for) for each parameter
- JCAMP - compare acquisition and processing parameters in JCAMP-DX compatible format. Inactive parameters are suppressed.

Examples diffparam abc.fid xyz.fid
diffparam ~/vnmrsys/exp[13] processing
diffparam ~/vnmrsys/exp[12]/curpar

Related `listparam` list parameters in simple format (UNIX)
`vnmr2jcamp` create JCAMP parameters from VNMR parameters (UNIX)

`diffparams` Report differences between two parameter sets (U)

Syntax `diffparams <-list> file1 file2 <macroname>`

Description Reports differences between parameter sets. A macro can optionally be created that will convert `file1` into `file2`.

Arguments `file1` and `file2` are parameter files, like `$HOME/vnmrsys/exp1/procpar $HOME/vnmrsys/exp1/curpar $HOME/vnmrsys/global /vnmr/conpar xyz.fid/procpar file1` and `file2` can also be directories (`xyz.fid` or `xyz.par`, or a local experiment like `~/vnmrsys/exp1`); in this case `diffparams` will look for a subfile `procpar` in these directories. The optional `-list` argument will cause a list of the parameters which are different to be printed. If the `-list` option is used, the macro feature is turned off. If a parameter exists in `file1` but not `file2`, it is not listed. If a parameter exists in `file2` but not `file1`, it is listed. If the parameter exists in both files, it is listed if the values are different. It is not listed if other information associated with the parameter is different. This other information is things like protection bits, maximum values, group, type, etc.

An optional third argument specifies the pathname of a macro to output. This macro will contain the MAGICAL commands necessary to convert `file1` into `file2`.

Examples
```
diffparams abc.fid xyz.fid
diffparams -list abc.fid xyz.fid
diffparams ~/vnmrsys/exp1  ~/vnmrsys/exp3
diffparams ~/vnmrsys/exp1  ~/vnmrsys/exp3
~/vnmrsys/maclib/change1to3
```

diffshims — Compare two sets of shims (M,U)

Syntax `diffshims(shimfile1,shimfile2)`
(From UNIX) `diffshims shimfile1 shimfile2`

Description Compares values for room-temperature shims stored in two separate files.

Arguments `shimfile1` and `shimfile2` are names of separate files containing shim values. Both files must have been written using the `svs` command.

See also *NMR Spectroscopy User Guide*

Related

`svs`	Save shim coil settings (C)

digfilt — Write digitally filtered FIDs to another experiment (M)

Syntax `digfilt(exp_number<,option>)`

Description Saves digitally filtered FIDs to another experiment.

Arguments `exp_number` specifies the number of the experiment, from 1 to 9, for saving the FIDs.

`option` is one of the keywords `'nodc'`, `'zero'`, `'lfs'`, `'zfs'`, or `'t2dc'`. Use a keyword for an option if the same option was used when processing the data with `ft`, `wft`, `ft2d`, or `wft2d`.

See also *NMR Spectroscopy User Guide*

Related

`downsamp`	Sampling factor applied after digital filtering (P)
`ft`	Fourier transform 1D data (C)
`ft2d`	Fourier transform 2D data (C)
`wft`	Weight and Fourier transform 1D data (C)
`wft2d`	Weight and Fourier transform 2D data (C)

dir — List files in directory (C)

Syntax `dir<(string)>`

Description Displays files in a directory on the text window. The `dir` command is identical to the `ls` and `lf` commands.

Arguments `string` is a string argument containing the options and/or directory names used if this were the UNIX `ls` command (e.g., `dir('-l *.fid')` requests a long listing (`-l`) of all files ending with `.fid` (`*.fid`)). If no argument is entered, `dir` lists all files in the current working directory.

Examples

```
dir
dir('data')
dir('-l *.fid')
```

See also *NMR Spectroscopy User Guide*

Related		
	lf	List files in directory (C)
	ls	List files in directory (C)

display Display parameters and their attributes (C)

Syntax `display(parameter|'*'|'**'<,tree>)`

Description Displays one or more parameters and their attributes from a parameter tree.

Arguments Three levels of display are available: `parameter`, `'*'`, and `'**'`.

- `parameter` is the name of a single parameter and the display is of its attributes (e.g., `display('a')` displays the attributes of parameter `a` in the (default) current tree).
- `'*'` is a keyword to display the name and values of all parameters in a tree (e.g., `display('*','global')` displays all parameter names and values in the global tree).
- `'**'` is a keyword to display the attributes of all parameters in a tree (e.g., `display('**','processed')` displays the attributes of all parameters in the processed tree).

`tree` is the type of parameter tree and can be `'global'`, `'current'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the `create` command for more information on types of trees.

Examples
```
display('a')
display('*','global')
display('**','processed')
```

See also *User Programming*

Related		
	create	Create new parameter in a parameter tree (C)
	destroy	Destroy a parameter (C)
	paramvi	Edit a parameter and its attributes with the `vi` text editor (C)
	prune	Prune extra parameters from current tree (C)

dla Display spin simulation parameter arrays (M)

Syntax `dla<('long')>`

Description Displays the parameters containing the line assignments for spin simulation iteration (matching simulated spectra to actual data). A `clindex` value of a calculated transition gives the index of the assigned measured line. The value is zero for unassigned transitions.

Arguments `'long'` is a keyword to display the parameters containing the line assignments for spin simulation iteration (matching simulated spectra to actual data) and put the line assignments into the file `spini.la`. This option is most useful when the `dla` display is too large to display all the calculated transitions in the text window. The dlalong command operates the same as the dla('long') command.

Examples dla
dla('long')

See also *NMR Spectroscopy User Guide*

Related		
	assign	Assign transitions to experimental lines (M)
	clindex	Index of experimental frequency of a transition (P)
	dga	Display parameters of spin simulation group (C)
	dlalong	Long display of spin simulation parameter arrays (C)

dlalong Long display of spin simulation parameter arrays (C)

Syntax dlalong

Description Puts line assignments into the file spini.la in a more complete form, then displays this file in the text window. It is most useful when the dla display is too large to display all the calculated transitions in the text window. The dla('long') command operates the same as dlalong.

See also *NMR Spectroscopy User Guide*

Related dla Display spin simulation parameter arrays (M)

dLC Display LC detector trace(s) in a horizontal format

Applicability VnmrJ 3.1

See also pLC
dLCNMR
pLCNMR

dLCNMR Displays all forms of LC-NMR data

Applicability VnmrJ 3.1

Description This macro is executed with a button on the LC-NMR display pane (labeled spare). Displays on-flow and stopped-flow 1D LC-NMR data. With on-flow data, dconi is used to display the NMR data with the time-aligned LC detector trace(s) along the left side. In the stopped-flow mode, dLC displays the 1D NMR data for each stop code at a position that it is time-aligned with the relevant LC peak. If arguments are supplied, dLCNMR passes the supplied arguments to dconi and forces a contour plot display. With no arguments, or when activated by the "Display LC & NMR" button, the dconi display uses the dconi parameter to determine the default display mode. The "Contour" check-box can be used to select the contour map (dpcon) display mode instead of the default color intensity map (dconi) display.

Examples dLCNMR(<number of contours>,<contour spacing>)

See also dLC
pLC
dLCNMR
pLCNMR

dli Display list of integrals (C)

Description Displays a list of integrals at the integral reset points. The frequency units of the displayed list of integrals is controlled by the parameter axis. The reset points may be defined with the z command and these frequencies are stored in lifrq. The calculated amplitudes of the integral region are stored in liamp. The reset points are stored as hertz and are not referenced to rfl and rfp. The amplitudes are stored as the actual value; they are not scaled by ins or by insref. When the integral blanking mode is used (i.e., intmod='partial'), only the integrals corresponding to the displayed integral regions are listed.

The displayed integral value can be scaled with the setint macro. The integral is scaled by the parameters ins and insref.

See also *NMR Spectroscopy User Guide*

Related

axis	Axis label for displays and plots (P)
cz	Clear integral reset points (C)
dlni	Display list of normalized integrals (M)
ins	Integral normalization scale (P)
insref	Fourier number scaled value of an integral (P)
liamp	Amplitudes of integral reset points (P)
lifrq	Frequencies of integral reset points (P)
nli	Find integral values (C)
rfl	Reference peak position in directly detected dimension (P)
rfp	Reference peak frequency in directly detected dimension (P)
setint	Set value of an integral (M)
z	Add integral reset point at cursor position (C)

dlivast Produce text file and process wells (M)

Applicability VAST accessory.

Syntax dlivast<(last)>

Description	Produces a text file containing the integral of the partial regions and processes the wells.
Arguments	`last` is the number of the last well. The default is 96.
See also	*NMR Spectroscopy User Guide*

Related
- `combiplate` View a color map for visual analysis of VAST microtiter plate (U)
- `combishow` Display regions as red, green, and blue in CombiPlate window (M)

`dll` Display listed line frequencies and intensities (C)

Syntax `dll<('pos'<,noise_mult>)><:number_lines,scale>`

Description Displays a list of line frequencies and amplitudes that are above a threshold defined by `th`. Frequency units are defined by the parameter `axis`. The results of this calculation are stored in `llfrq` and `llamp`. The frequencies are stored as Hz and are not referenced to `rfl` and `rfp`. Amplitudes are stored as the actual data point value; they are not scaled by `vs`.

Arguments `'pos'` is a keyword to list only positive lines.

`noise_mult` is a numerical value that determines the number of noise peaks listed for broad, noisy peaks. The default value is 3. A smaller value results in more peaks, a larger value results in fewer peaks, and a value of 0.0 results in a line listing containing all peaks above the threshold `th`. Negative values of `noise_mult` are changed to 3.

`number_lines` is a return argument with the number of lines above the threshold.

`scale` is a return argument with a scaling factor for line amplitudes. This scaling factor accounts for `vs` and whether the lines are listed in absolute intensity mode or normalized mode.

Examples

```
dll
dll('pos')
dll(2.5)
dll:r1,sc
```

See also *NMR Spectroscopy User Guide*

Related

`axis`	Axis label for displays and plots (P)
`dels`	Delete spectra from T_1 or T_2 analysis (C)
`fp`	Find peak heights (C)
`getll`	Get frequency and intensity of a line (C)
`llamp`	List of line amplitudes (P)
`llfrq`	List of line frequencies (P)
`nl`	Position the cursor at the nearest line (C)
`nll`	Find line frequencies and intensities (C)
`rfl`	Reference peak position in directly detected dimension (P)
`rfp`	Reference peak frequency in directly detected dimension (P)

`th`	Threshold (P)
`vs`	Vertical scale (P)

dlni — Display list of normalized integrals (M)

Description: Displays integrals in a normalized format. The parameter `ins` represents the value of the sum of all the integrals. When the integral blanking mode is used (i.e., `intmod='partial'`), only the integrals corresponding to the displayed integral regions are listed and are used in the summation.

See also: *NMR Spectroscopy User Guide*

`cz`	Clear integral reset points (C)
`dli`	Display list of integrals (C)
`ins`	Integral normalization scale (P)
`nli`	Find integral values (C)
`z`	Add integral reset point at cursor position (C)

dlp — Decoupler low-power control with class C amplifier (P)

Applicability: Systems with a class C amplifier.

Description: `dlp` controls the decoupler power level for systems with a class C decoupler amplifier in the low-power mode, generally used for homonuclear decoupling. `dlp` specifies dB of attenuation of the decoupler, below a nominal 1 watt value. `dlp` is active only if `dhp='n'`.

On systems with a decoupler linear amplifier, `dlp` is nonfunctional and `dpwr` controls decoupler power.

Values: 0 to 39 (in dB of attenuation, 0 is maximum power).

See also: *NMR Spectroscopy User Guide*

Related:

`dhp`	Decoupler high-power control with class C amplifier (P)
`dm`	Decoupler mode for first decoupler (P)
`dmf`	Decoupler modulation frequency for first decoupler (P)
`dpwr`	Power level for first decoupler with linear amplifier (P)

dm — Decoupler mode for first decoupler (P)

Applicability: VNMRS systems

Description: Determines the state of first decoupler during different status periods within a pulse sequence (refer to the manual *User Programming* for a discussion of status periods). Pulse sequences may require one, two, three, or more different decoupler states. The number of letters that make up the `dm` parameter vary appropriately, with each letter representing a status period (e.g., `dm='yny'` or `dm='ns'`). If the decoupler status is constant for the entire pulse sequence, it can be entered as a single letter (e.g., `dm='n'`).

Values `'n'`, `'y'`, `'a'`, or `'s'` (or a combination of these values), where:

`'n'` specifies no decoupler rf.

`'y'` specifies the asynchronous mode. In this mode, the decoupler rf is gated on and modulation is started at a random places in the modulation sequence.

On the VNMRS system, the default asynchronous decoupling uses a "progressive offset" scheme. Other asynchronous schemes are also implemented on the VNMRS. They can be selected using an optional flag parameter "`decasynctype`". Create "`decasynctype`" as a flag parameter in the current tree and set the following:

`decasynctype` = 'p' selects the "progressive offset" scheme (default)
= 'b' selects the "bit reversed" scheme, and
= 'r' selects the random scheme.

`'a'` specifies the asynchronous mode, the same as `'y'`.

`'s'` specifies the synchronous mode in which the decoupler rf is gated on and modulation is started at the beginning of the modulation sequence.

See also *NMR Spectroscopy User Guide*

Related

`dm2`	Decoupler mode for second decoupler (P)
`dm3`	Decoupler mode for third decoupler (P)
`dm4`	Decoupler mode for fourth decoupler (P)
`dmf`	Decoupler modulation frequency for first decoupler (P)
`dmm`	Decoupler modulation mode for first decoupler (P)
`dn`	Nucleus for first decoupler (P)
`decasynctype`	Decoupler asynchronous mode (P)

dm2 Decoupler mode for second decoupler (P)

Applicability Systems with a second decoupler.

Description Determines the state of second decoupler during different status periods within a pulse sequence. It functions analogously to `dm`.

Values Same as `dm`, except that if `dn2=''` (two single quotes with no space in between) and a second decoupler is present in the console, `dm2` assumes a default value of `'n'` when `go` is executed.

See also *NMR Spectroscopy User Guide*

Related

`dm`	Decoupler mode of first decoupler (P)
`dmf2`	Decoupler modulation frequency for second decoupler (P)
`dmm2`	Decoupler modulation mode for second decoupler (P)
`dn2`	Nucleus for second decoupler (P)

dm3 Decoupler mode for third decoupler (P)

Applicability Systems with a third decoupler.

Description Determines the state of third decoupler during different status periods within a pulse sequence. It functions analogously to `dm`.

Values Same as `dm`, except that if `dn3=''` (two single quotes with no space in between) and a third decoupler is present in the console, `dm3` assumes a default value of `'n'` when `go` is executed.

See also *NMR Spectroscopy User Guide*

Related

`dm`	Decoupler mode of first decoupler (P)
`dmf3`	Decoupler modulation frequency for third decoupler (P)
`dmm3`	Decoupler modulation mode for third decoupler (P)
`dn3`	Nucleus for third decoupler (P)
`decasynctype`	Select the type of decoupler asynchronous mode (P)

dm4 Decoupler mode for fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Determines the state of fourth decoupler during different status periods within a pulse sequence. It functions analogously to `dm`.

Values Same as `dm`, except that if `dn4=''` (two single quotes with no space in between) and a fourth decoupler is present in the console, `dm4` assumes a default value of `'n'` when `go` is executed.

See also *NMR Spectroscopy User Guide*

Related

`dm`	Decoupler mode of first decoupler (P)
`dmf4`	Decoupler modulation frequency for fourth decoupler (P)
`dmm4`	Decoupler modulation mode for fourth decoupler (P)
`dn4`	Nucleus for fourth decoupler (P)
`decasynctype`	Select the type of decoupler asynchronous mode (P)

dmf Decoupler modulation frequency for first decoupler (P)

Description Controls modulation frequency of the first decoupler. It specifies `1/pw90` at the particular power level used. After calibrating the decoupler field strength γH_2 (expressed in units of Hz), `dmf` should be set equal to $4^*\gamma H_2$ for WALTZ, MLEV16, GARP, and XY32 (when available).

`dmf` is inactive for CW mode decoupling (`dmm='c'`).

`dmf` is also active for square wave mode decoupling (`dmm='r'`) and fm-fm mode (`dmm='f'`) decoupling. For `dmm='f'`, the modulation frequency is swept back and forth between about 0.5% and 5% of the dmf frequency (e.g., if dmf is 100 kHz, the modulation is swept between approximately 500 Hz and 5 kHz). A reasonable optimum value for dmf when dmm='f' is the decoupler frequency divided by 4000.

Values 5 Hz to 2 MHz in steps of 5 Hz (steps are actually approximately 4.768 Hz).

For GARP modulation, the dmf value is internally multiplied by 45, making the limit of possible dmf values to 5 Hz to 44.4 kHz when dmm='g'.

See also *NMR Spectroscopy User Guide*

Related

dmf2	Decoupler modulation frequency for second decoupler (P)
dmf3	Decoupler modulation frequency for third decoupler (P)
dmf4	Decoupler modulation frequency for fourth decoupler (P)
dmm	Decoupler modulation mode for first decoupler (P)
pw90	90° pulse width (P)

dmf2 Decoupler modulation frequency for second decoupler (P)

Applicability Systems with a second decoupler.

Description Controls the modulation frequency of the second decoupler. It functions analogously to the parameter dmf.

Values Same as dmf except that if dn2='' (two single quotes with no space in between) and a second decoupler is present in the console (numrfch greater than 2), dmf2 assumes a default value of 1000 Hz when go is executed.

See also *NMR Spectroscopy User Guide*

Related

dm2	Decoupler mode for second channel (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmm2	Decoupler modulation mode for second decoupler (P)
dn2	Nucleus for second decoupler (P)
numrfch	Number of rf channels (P)

dmf3 Decoupler modulation frequency for third decoupler (P)

Applicability Systems with a third decoupler.

Description Controls the modulation frequency of the third decoupler. It functions analogously to the parameter dmf.

Values Same as dmf except that if dn3='' (two single quotes with no space in between) and a third decoupler is present in the console (numrfch equals 4), dmf3 assumes a default value of 1000 Hz when go is executed.

See also *NMR Spectroscopy User Guide*

Related
dm3	Decoupler mode for third channel (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmm3	Decoupler modulation mode for third decoupler (P)
dn3	Nucleus for third decoupler (P)
numrfch	Number of rf channels (P)

dmf4 Decoupler modulation frequency for fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Controls the modulation frequency of the fourth decoupler. It functions analogously to the parameter `dmf`.

Values Same as `dmf` except that if `dn4=''` (two single quotes with no space in between) and a fourth decoupler is present in the console (`numrfch` equals 5), `dmf4` assumes a default value of 1000 Hz when `go` is executed.

See also *NMR Spectroscopy User Guide*

Related
dm4	Decoupler mode for fourth channel (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmm4	Decoupler modulation mode for fourth decoupler (P)
dn4	Nucleus for fourth decoupler (P)
numrfch	Number of rf channels (P)

dmfadj Adjusts the parameter 'dmf'

Syntax `dmfadj(<tipangle_resoln>)`

Applicability VnmrJ 3.1

Description `` `dmfadj` `` adjusts the parameter '`dmf`' so that the time associated with the tip-angle resolution is an integral multiple of 100 ns. This insures that there is no truncation error in time in the execution of the programmable decoupling or spin-locking sequence by the waveform generator. The optional argument 'tipangle_resoln' specifies the necessary tip-angle resolution for the programmable decoupling or spin-locking sequence which is to be executed. For example, the tip-angle resolution for an MLEV-16 decoupling sequence should be 90.0 degrees since every pulse in that sequence can be represented as an integral multiple of 90.0 degrees; the tip-angle resolution for a GARP decoupling sequence, however, should be 1.0 degrees.

Arguments If the argument 'tipangle_resoln' is not specified when the macro `` `dmfadj` `` is called, the default value therefore is taken from the parameter '`dres`'.

Related
dmf2adj	adjusts the parameter 'dmf2
pwsadj	adjusts 'pulse_parameter'

dmf2adj — Adjust tip-angle resolution time for second decoupler (M)

Applicability	Systems with a second decoupler.
Syntax	`dmf2adj<(tipangle_resolution)>`
Description	Adjusts the parameter `dmf2` to make time associated with the second decoupler tip-angle resolution an integral multiple of 50 ns. `dmf2adj` functions analogously to the macro `dmfadj`.
Arguments	`tipangle_resolution` specifies the necessary tip-angle resolution for the programmable decoupling or spin-locking sequence to be executed. The default value is the current value of the parameter `dres2`.
Examples	`dmf2adj` `dmf2adj(90.0)`
See also	*NMR Spectroscopy User Guide*

Related

`dmf2`	Decoupler modulation frequency for second decoupler (P)
`dmfadj`	Adjust decoupler tip-angle resolution time (M)
`dres2`	Tip angle resolution for second decoupler (P)

dmf3adj — Adjust tip-angle resolution time for third decoupler (M)

Applicability	Systems with a third decoupler.
Syntax	`dmf3adj<(tipangle_resolution)>`
Description	Adjusts the parameter `dmf3` to make time associated with the third decoupler tip-angle resolution an integral multiple of 50 ns. `dmf3adj` functions analogously to the macro `dmfadj`.
Arguments	`tipangle_resolution` specifies the necessary tip-angle resolution for the programmable decoupling or spin-locking sequence to be executed. The default value is the current value of the parameter `dres3`.
Examples	`dmf3adj` `dmf3adj(90.0)`
See also	*NMR Spectroscopy User Guide*

Related

`dmf3`	Decoupler modulation frequency for third decoupler (P)
`dres3`	Tip-angle resolution for third decoupler (P)

dmf4adj — Adjust tip-angle resolution time for fourth decoupler (M)

Applicability	Systems with a deuterium decoupler as the fourth decoupler.
Syntax	`dmf4adj<(tipangle_resolution)>`

Description Adjusts the parameter dmf4 to make time associated with the fourth decoupler tip-angle resolution an integral multiple of 50 ns. `dmf4adj` functions analogously to the macro dmfadj.

Arguments `tipangle_resolution` specifies the necessary tip-angle resolution for the programmable decoupling or spin-locking sequence to be executed. The default value is the current value of the parameter dres4.

Examples `dmf4adj`

See also *NMR Spectroscopy User Guide*

Related		
	dmf4	Decoupler modulation frequency for fourth decoupler (P)
	dres4	Tip-angle resolution for fourth decoupler (P)

dmg Data display mode in directly detected dimension (P)

Description Controls the mode of data display along the directly detected dimension. `dmg` is in the display group and can be set manually or by executing the commands ph, av, pwr, or pa for the values 'ph', 'av', 'pwr', or 'pa', respectively.

Values `'ph'` sets the *phased mode* in which each real point in the displayed spectrum is calculated from a linear combination of real and imaginary points comprising each respective complex data point.

`'av'` sets the *absolute-value mode* in which each real point in the displayed spectrum is calculated as the square root of the sum of squares of the real and imaginary points comprising each respective complex data point.

`'pwr'` sets the *power mode* in which each real point in the displayed spectrum is calculated as the sum of squares of the real and imaginary points comprising each respective complex data point.

`'pa'` sets the *phase angle* mode in which each real point in the displayed spectrum is calculated as the phase angle from the arc tangent of the real and imaginary points comprising each respective complex data point.

See also *NMR Spectroscopy User Guide*

Related		
	aig	Absolute intensity group (P)
	av	Set absolute-value mode in directly detected dimension (C)
	dcg	Drift correction group (P)
	dmg1	Data display mode in 1st indirectly detected dimension (P)
	dmg2	Data display mode in 2nd indirectly detected dimension (P)
	ft	Fourier transform 1D data (C)
	ft1d	Fourier transform along f_2 dimension (C)
	ft2d	Fourier transform 2D data (C)
	pa	Set phase angle mode in directly detected dimension (C)
	ph	Set phased mode in directly detected dimension (C)
	pmode	Processing mode for 2D data (P)

pwr	Set power mode in directly detected dimension (C)
wft	Weigh and Fourier transform 1D data (C)
wft1d	Weigh and Fourier transform of 2D data (C)
wft2d	Weigh and Fourier transform 2D data (C)

dmg1 Data display mode in 1st indirectly detected dimension (P)

Description Controls the mode of data display along the first indirectly detected dimension of a multidimensional data set. `dmg1` is in the display group and can be set manually or by executing the commands `ph1`, `av1`, `pwr1`, or `pa1` for the values `'ph1'`, `'av1'`, `'pwr1'`, or `'pa1'`, respectively. If `dmg1` does not exist or if it is set to the empty string (`dmg1=''`), VnmrJ uses the value of `dmg` to decide the display mode along the first indirectly detected dimension.

Values `'ph1'` sets phased mode.

`'av1'` sets absolute-value mode.

`'pwr1'` sets power mode.

`'pa1'` sets phase angle mode.

See also *NMR Spectroscopy User Guide*

Related

av1	Set absolute-value mode in 1st indirectly det. dim. (C)
dmg	Data display mode in directly detected dimension (P)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
ph1	Set phased mode in 1st indirectly detected dimension (C)
pwr1	Set power mode in 1st indirectly detected dimension (C)

dmg2 Data display mode in 2nd indirectly detected dimension (P)

Description Controls the mode of data display along the second indirectly detected dimension of a multidimensional data set. `dmg2` is in the display group and can be set manually or by executing the commands `ph2`, `av2`, or `pwr2` for the values `'ph2'`, `'av2'`, or `'pwr2'`, respectively. If `dmg2` does not exist or if it is set to the empty string (`dmg2=''`), VnmrJ uses the value of the parameter `dmg` instead of `dmg2` to decide the display mode along the second indirectly detected dimension.

Values `'ph2'` sets phased mode.

`'av2'` sets absolute-value mode.

`'pwr2'` sets power mode.

See also *NMR Spectroscopy User Guide*

Related		
	av2	Set absolute-value mode in 2nd indirectly det. dim. (C)
	dmg	Data display mode in directly detected dimension (P)
	ph2	Set phased mode in 2nd indirectly det. dim. (C)
	pwr2	Set power mode in 2nd indirectly det. dim. (C)

dmgf Absolute-value display of FID data or spectrum in acqi (P)

Description If the parameter `dmgf` exists and is set to `'av'`, the FID display in the `acqi` program is set to the absolute-value mode, which displays the square root of the sum of the squares of the real and imaginary channels. `dmgf` has no function outside of the `acqi` program. This display mode may cause the displayed FID to exceed the displayed ADC limits in `acqi` by as much as a factor of the square root of 2.

See also *NMR Spectroscopy User Guide*

Related;		
	acqi	Interactive acquisition display process (C)
	av	Set absolute-value mode in directly detected dimension (C)
	gf	Prepare parameters for FID/spectrum display in `acqi` (M)

dmm Decoupler modulation mode for first decoupler (P)

Description Sets the modulation modes for the first decoupler. In the standard two-pulse sequence, `dmm` typically has a single state because the decoupler modulation is normally not changed during the pulse sequence, but this is not fixed. For example, `dmm='ccw'` gives single-frequency CW decoupling during the first part of the sequence and WALTZ-16 decoupling during acquisition.

In pulse sequences using the decoupler for pulsing (INEPT, DEPT, HETCOR, etc.), decoupler modulation must be set to `'c'` during periods of the pulse sequence when the decoupler is to be pulsed.

Values `'c'`, `'f'`, `'g'`, `'m'`, `'p'`, `'r'`, `'u'`, `'w'`, and `'x'` are available;:

- `'c'` sets continuous wave (CW) modulation.
- `'f'` sets fm-fm modulation (swept-square wave).
- `'g'` sets GARP modulation.
- `'m'` sets MLEV-16 modulation.
- `'n'` sets noise modulation.
- `'p'` sets programmable pulse modulation using the `dseq` parameter to specify the decoupling sequence.
- `'r'` sets square-wave modulation.
- `'u'` sets user-supplied modulation using external hardware.

- 'w' sets WALTZ-16 modulation.
- 'x' sets XY32 modulation.

See also *NMR Spectroscopy User Guide*

Related

dm	Decoupler mode for first decoupler (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmm2	Decoupler modulation mode for second decoupler (P)
dmm3	Decoupler modulation mode for third decoupler (P)
dmm4	Decoupler modulation mode for fourth decoupler (P)
dseq	Decoupler sequence for the first decoupler (P)

dmm2 Decoupler modulation mode for second decoupler (P)

Applicability Systems with a second decoupler.

Description Sets the type of decoupler modulation for the second decoupler during different status periods within a pulse sequence. It functions analogously to dmm.

Values 'c', 'f', 'g', 'm', 'p', 'r', 'u', 'w', and 'x' are available. Refer to dmm for the definition of these values (note that if the mode 'p' is selected, dseq2 specifies the decoupling sequence). If dn2='' (two single quotes) and a second decoupler is present in the console (numrfch greater than 2), dmm2 is internally set to 'c' when go is executed.

See also *NMR Spectroscopy User Guide*

Related

dm2	Decoupler modulation for the second decoupler (P)
dmf2	Decoupler modulation frequency for the second decoupler (P)
dmm	Decoupler modulation mode for first decoupler (P)
dn2	Nucleus for the second decoupler (P)
dseq2	Decoupler sequence for the second decoupler (P)
numrfch	Number of rf channels (P)

dmm3 Decoupler modulation mode for third decoupler (P)

Applicability Systems with a third decoupler.

Description Sets type of decoupler modulation for the third decoupler during different status periods within a pulse sequence. It functions analogously to dmm.

Values 'c', 'f', 'g', 'm', 'p', 'r', 'u', 'w', and 'x' are available. Refer to dmm for the definition of these values (note that if the mode 'p' is selected, dseq3 specifies the decoupling sequence). If dn3='' (two single quotes) and a third decoupler is present in the console (numrfch equal to 4), dmm3 is internally set to 'c' when go is executed.

See also	*NMR Spectroscopy User Guide*	
Related	dm3	Decoupler modulation for third decoupler (P)
	dmf3	Decoupler modulation frequency for third decoupler (P)
	dmm	Decoupler modulation mode for first decoupler (P)
	dn3	Nucleus for the third decoupler (P)
	dseq3	Decoupler sequence for the third decoupler (P)
	numrfch	Number of rf channels (P)

dmm4 Decoupler modulation mode for fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Sets type of decoupler modulation for the fourth decoupler during different status periods within a pulse sequence. It functions analogously to dmm.

Values 'c', 'f', 'g', 'm', 'r', 'u', 'w', and 'x' are available. Refer to dmm for the definition of these values. If dn4='' (two single quotes) and a fourth decoupler is present in the console (numrfch greater than 4), dmm4 is internally set to 'c' when go is executed.

See also *NMR Spectroscopy User Guide*

Related	dm4	Decoupler modulation for the fourth decoupler (P)
	dmf4	Decoupler modulation frequency for the fourth decoupler (P)
	dmm	Decoupler modulation mode for first decoupler (P)
	dn4	Nucleus for the fourth decoupler (P)
	dseq4	Decoupler sequence for the fourth decoupler (P)
	numrfch	Number of rf channels (P)

dn Nucleus for first decoupler (P)

Description Changing the value of dn causes a macro (named _dn) to be executed that extracts values for dfrq and dof from lookup tables. The tables, stored in the directory /vnmr/nuctables, are coded by atomic weights.

Values In the lookup tables, typically 'H1', 'c13', 'P31', etc.

See also *NMR Spectroscopy User Guide*

Related	dfrq	Transmitter frequency of first decoupler (P)
	dn2	Nucleus for second decoupler (P)
	dn3	Nucleus for third decoupler (P)
	dn4	Nucleus for fourth decoupler (P)
	dof	Frequency offset for first decoupler (C)
	tn	Nucleus for observe transmitter (P)

dn2 Nucleus for second decoupler (P)

Applicability Systems with a second decoupler.

Description Changing the value of dn2 causes a macro (named _dn2) to be executed that extracts values for dfrq2 and dof2 from lookup tables. Otherwise, dn2 functions analogously to the parameters tn and dn. If an experiment does not use the second decoupler channel, the channel can be disabled by setting dn2='' (two single quotes with no space in between). This sets dm2='n', dmm2='c', dmf2=1000 (in Hz), dfrq2=1 (in MHz), dof2=0, dpwr2=0, dseq2='', and dres2=1.

See also *NMR Spectroscopy User Guide*

Related

dfrq2	Transmitter frequency of second decoupler (P)
dn	Nucleus for first decoupler (P)
dof2	Frequency offset for second decoupler (C)
numrfch	Number of rf channels (P)
tn	Nucleus for observe transmitter (P)

dn3 Nucleus for third decoupler (P)

Applicability Systems with a third decoupler.

Description Changing the value of dn3 causes a macro (named _dn3) to be executed that extracts values for dfrq3 and dof3 from lookup tables. Otherwise, dn3 functions analogously to the parameters tn and dn. If an experiment does not use the third decoupler channel, the channel can be disabled by setting dn3='' (two single quotes with no space in between). This sets dm3='n', dmm3='c', dmf3=1000 (in Hz), dfrq3=1 (in MHz), dof3=0, dpwr3=0, dseq3='', and dres3=1.

See also *NMR Spectroscopy User Guide*

Related

dn	Nucleus for first decoupler (P)
dfrq3	Transmitter frequency of third decoupler (P)
dof3	Frequency offset for third decoupler (C)
numrfch	Number of rf channels (P)
tn	Nucleus for observe transmitter (P)

dn4 Nucleus for fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Changing the value of dn4 causes a macro (named _dn4) to be executed that extracts values for dfrq4 and dof4 from lookup tables. Otherwise, dn4 functions analogously to the parameters tn and dn except that the only valid value for dn4 is 'H2'. If an experiment does not use the fourth decoupler channel, the channel can be disabled by setting dn4='' (two single quotes with no space in between). This sets dm4='n', dmm4='c', dmf4=1000 (in Hz), dfrq4=1 (in MHz), dof4=0, dpwr4=0, dseq4='', and dres4=1.

See also *NMR Spectroscopy User Guide*

Related

dfrq4	Transmitter frequency of fourth decoupler (P)
dn	Nucleus for first decoupler (P)
dof4	Frequency offset for fourth decoupler (C)
numrfch	Number of rf channels (P)
tn	Nucleus for observe transmitter (P)

dndfid Retrieve and process fid data from the locator (M)

Applicability Liquids, Imaging, Solids

Description Retrieve fid data from an item selected in the locator. Data is also processed if Process data on drag-and-drop from locator is selected in the System settings dialog in the Utilities menu.

Related

dndjoin	Join a work space from the locator (M)
dndpar	Retrieve a parameter set from the locator (M)
dndshims	Retrieve a shimset set from the locator (M)
locaction	Locator action (M)
locprotoexec	Execute a protocol from the locator (M)
xmmakenode	Make a new study queue node (M)

dndjoin Join a work space from the locator (M)

Description Join the work space selected by the locator.

Related

dndfid	Retrieve and process fid data from the locator (M)
dndpar	Retrieve a parameter set from the locator (M)
dndshims	Retrieve a shimset set from the locator (M)
locaction	Locator action (M)
locprotoexec	Execute a protocol from the locator (M)
xmmakenode	Make a new study queue node (M)

dndpar Retrieve a parameter set from the locator (M)

Description Retrieve a parameter set selected by the locator.

Related

dndfid	Retrieve and process fid data from the locator (M)
dndjoin	Join a work space from the locator (M)
dndshims	Retrieve a shimset set from the locator (M)
locaction	Locator action (M)

locprotoexec	Execute a protocol from the locator (M)
xmmakenode	Make a new study queue node (M)

dndshims Retrieve a shimset set from the locator (M)

Description Retrieve a shimset set selected by the locator.

Related

dndfid	Retrieve and process fid data from the locator (M)
dndjoin	Join a work space from the locator (M)
dndpar	Retrieve a parameter set from the locator (M)
locaction	Locator action (M)
locprotoexec	Execute a protocol from the locator (M)
xmmakenode	Make a new study queue node (M)

dnode Display list of valid limNET nodes (M,U)

Applicability Systems with limNET.

Description Displays the contents of the user's limNET node database (i.e., all remote nodes available to limNET). Each node is listed by name, Ethernet address (6 hexadecimal bytes), and burst size

See also *NMR Spectroscopy User Guide*

Related eaddr Display Ethernet address (M,U)

doautodialog Start a dialog window using `def` file (M)

Applicability Systems with automation.

Syntax `doautodialog`

Description Internal macro used by enter to start a dialog window using the `def` file for an experiment in the `dialoglib` directory.

Related enter Enter sample information for automation run (M,U)

dodialog Start a dialog window with `dialoglib` file (M)

Syntax `dodialog`

Description Internal macro that starts a dialog window using a dialog file in the `dialoglib` directory.

dof — Frequency offset for first decoupler (P)

Description: Controls the frequency offset of the first decoupler. Higher numbers move the decoupler to higher frequency (toward the left side of the spectrum). The frequency accuracy of the decoupler offset is generally0.1 Hz. The value is specified in the `config` program.

Values: –100000 to 100000 Hz (approximate, depends on frequency), in steps of 0.1 Hz.

See also: *NMR Spectroscopy User Guide*

Related:

`config`	Display current configuration and possible change it (M)
`dof2`	Frequency offset for second decoupler (P)
`dof3`	Frequency offset for third decoupler (P)
`dof4`	Frequency offset for fourth decoupler (P)
`tof`	Frequency offset for observe transmitter (P)

dof2 — Frequency offset for second decoupler (P)

Applicability: Systems with a second decoupler.

Description: Controls the frequency offset for the second decoupler. `dof2` functions analogously to the parameters `tof` and `dof`.

Values: –100000 to 100000 Hz (approximate, depends on frequency), in steps of 0.1 Hz. If `dn2=''` (two single quotes with no space in between) and a second decoupler channel is present in the console, `dof2` assumes a default value of `0` when `go` is executed.

See also: *NMR Spectroscopy User Guide*

Related:

`dn2`	Nucleus for second decoupler (P)
`dof`	Frequency offset for first decoupler (P)
`tof`	Frequency offset for observe transmitter (P)

dof3 — Frequency offset for third decoupler (P)

Applicability: Systems with a third decoupler.

Description: Controls the frequency offset for the third decoupler. `dof3` functions analogously to the parameters `tof` and `dof`.

Values: –100000 to 100000 Hz (approximate, depends on frequency), in steps of 0.1 Hz. If `dn3=''` (two single quotes with no space in between) and a third decoupler channel is present in the console, `dof3` assumes a default value of `0` when `go` is executed.

See also: *NMR Spectroscopy User Guide*

Related:

`dn3`	Nucleus for third decoupler (P)

dof	Frequency offset for first decoupler (P)
tof	Frequency offset for observe transmitter (P)

dof4 Frequency offset for fourth decoupler (P)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Controls the frequency offset for the fourth decoupler. dof4 functions analogously to the parameters tof and dof.

Values –100000 to 100000 Hz (approximate, depends on frequency), in steps of 2.384 Hz. If dn4='' (two single quotes with no space in between) and a fourth decoupler channel is present in the console, dof4 assumes a default value of 0 when go is executed.

See also *NMR Spectroscopy User Guide*

Related

dn4	Nucleus for fourth decoupler (P)
dof	Frequency offset for first decoupler (P)
tof	Frequency offset for observe transmitter (P)

doneshot Set up parameters for Doneshot pulse sequence (M)

Description Converts a parameter set to Doneshot experiment.

See also *NMR Spectroscopy User Guide*

Related

dosy	Process DOSY experiments (M)
fiddle	Perform reference deconvolution (M)
setup_dosy	Set up gradient levels for DOSY experiments (M)

dopardialog Start a dialog with dialoglib/experiment def file (M)

Description Internal macro that starts a dialog window using a def file in the directory dialoglib/experiment.

do_pcss Calculate proton chemical shifts spectrum (C)

Syntax do_pcss<(<threshold><,max_cc><,max_width)>

Description Strips a high-resolution proton spectrum down to a list of chemical shifts. The list is saved in the file pcss.outpar. If no argument is given, do_pcss automatically calculates the threshold and uses default values for the maximum allowable coupling constant and the maximum width of a spin multiplet.

Arguments `threshold` sets the level whether a point belongs to a peak or is noise.

`max_cc` is the maximum allowable coupling constant in the spectrum. Default is 20 Hz.

`max_width` is the maximum width of a spin multiplet in the spectrum. Default is 60 Hz.

Examples
```
do_pcss
do_pcss(10)
do_pcss(9,20,80)
```

See also *NMR Spectroscopy User Guide*

Related

pcss	Calculate and show proton chemical shifts spectrum (M)

dosy Process DOSY experiments (M)

Syntax `dosy(<'prune'>,<lowerlimit,upperlimit>)`

Description Performs a DOSY (diffusion ordered spectroscopy) analysis of the data in an array of spectra.

`dosy` uses the commands `dll` and `fp` to determine the heights of all signals above the threshold defined by the parameter `th` and then fits the decay curve for each signal to a Gaussian using the program `dosyfit`. It stores a summary of all diffusion coefficients and their estimated standard errors and various other results as follows:

- In the directory `$HOME/vnmrsys/Dosy`: diffusion_display.inp, general_dosy_stats, calibrated_gradients, fit_errors, and diffusion_spectrum
- In the current experiment: a second copy of diffusion_display.inp.

The command `showdosy` has been incorporated into `dosy`.

Arguments `prune` starts a dialog to allow one or more spectra to be omitted from the analysis.
`lowerlimit` is the lower diffusion limit (in units of 10^{-10} m^2/s) to be displayed.
`upperlimit` is the upper diffusion limit (in units of 10^{-10} m^2/s) to be displayed.

Without arguments, `dosy` uses all the experimental spectra and covers the whole diffusion range seen in the experimental peaks.

See also *NMR Spectroscopy User Guide*

Related

ddif	Synthesize and display DOSY plot (C)
fiddle	Perform reference deconvolution (M)
setup_dosy	Set up gradient levels for DOSY experiments (M)

dosy2d — Apptype macro for dosy 2D experiments (M)

Applicability: Liquids

Description: Performs the actions for 2D dosy protocols to set up, process, and plot experiments. It is only available if the Dosy software is installed.

Related:

apptype	Application type (PM)
execpars	Set up the exec parameters (M)

dosy3Dflag — Used by the dosy macro to determine whether to use 2D or 3D DOSY processing

Syntax: `dosy3Dflag`

Applicability: VnmrJ 3.1

Description: `dosy3Dflag` is a parameter used by the dosy macro to determine whether to use 2D or 3D processing. It is normally set automatically, but can also be set manually, e.g. to force 2D processing of one increment of a 3D dataset.

Arguments: dosy3Dflag='y'

dosy3Dflag='n'

See also: dosy

dosy3Dproc — Used by the dosy macro to determine whether to use 2D or 3D processing

Syntax: `dosy3Dproc`

Applicability: VnmrJ 3.1

Description: `dosy3Dproc` is a parameter used by the dosy macro to determine whether to use 2D or 3D processing, and what type of the latter. It is normally set automatically, but can also be set manually, e.g. to force 2D processing of an increment extracted from a 3D dataset.

Arguments: dosy3Dproc='n'

dosy3Dproc='ntype'

dosy3Dproc='ptype'

dosy3Dproc='y'

See also: dosy

dosybypoints Determines whether peak picking is used by the dosy macro

Syntax `dosybypoints`

Applicability VnmrJ 3.1

Description Determines whether dosy produces a 2D display based on whole peaks (the default) or point by point (much slower) in the spectral dimension.

Arguments 'n' divides the spectrum into individual peaks, creating one cross-peak for each individual peak found in the 1D spectrum. 'y' performs a diffusion fit for every point in the displayed region of the spectrum that lies above the threshold th.

See also ddif

dosy

dosyfit fits 2D or 3D DOSY data to obtain diffusion coefficients, amplitudes and statistics

Syntax `dosyfit`

dosyfit('version')

dosyfit('3D')

dosyfit('3D', avgnoise)

Applicability VnmrJ 3.1

Description dosyfit performs monoexponential least squares fitting on signal intensities from 2D and 3D datasets, summarising the results in various files.

Arguments dosyfit takes 0, 1, or 2 arguments: 'version' returns the version number of the software, '3D' invokes processing of cross-peak volumes stored in the files peaks.bin.<n> rather than peak heights stored in the file dosy_in. In the case of 3D processing, the parameter avgnoise allows correction for the average baseplane noise in absolute value data

See also ddif

dosy

dosyfrq Larmor frequency of phase encoded nucleus in DOSY (P)

Description Stores the NMR frequency of the phase encoded nucleus in DOSY experiments. It is directly set by the DOSY sequences.

See also *NMR Spectroscopy User Guide*

Related dosy Process DOSY experiments (M)

dosygamma — Gyromagnetic constant of phase encoded nucleus in DOSY (P)

Description — Stores the gyromagnetic constant of the phase encoded nucleus in DOSY experiments. It is automatically set by the DOSY sequences and used by the `dosy` macro.

See also — *NMR Spectroscopy User Guide*

Related — `dosy` Process DOSY experiments (M)

dosypeaks — Determines whether peak picking is used by the dosy macro

Syntax — `dosypeaks`

Applicability — VnmrJ 3.1

Description — Determines whether `dosy` produces a 2D display based on whole peaks (the default) or point by point (much slower) in the spectral dimension.

Arguments — 'y' divides the spectrum into individual peaks, creating one cross-peak for each individual peak found in the 1D spectrum. 'n' performs a diffusion fit for every point in the displayed region of the spectrum that lies above the threshold `th`.

See also — `ddif`

`dosy`

dosyproc — Determines the type of processing performed by the dosy macro

Syntax — `dosyproc`

Applicability — VnmrJ 3.1

Description — Determines whether `dosy` produces a discrete or a continuous diffusion spectrum.

Arguments — 'discrete' invokes monoexponential fitting with `dosyfit` if ncomp=1, and multiexponential fitting with the external programme SPLMOD if ncomp>1. 'continuous' invokes processing with the external programme CONTIN and gives a continuous distribution in the diffusion domain.

See also — `dosy`

For information about the programmes SPLMOD and CONTIN please visit http://s-provencher.com/index.shtml.

dosytimecubed Gyromagnetic constant of phase encoded nucleus in DOSY (P)

Description	Time cubed factor in the expression for diffusional attenuation. It is automatically set by the DOSY sequences and used by the dosy macro.
See also	*NMR Spectroscopy User Guide*
Related	dosy — Process DOSY experiments (M)

dot1 Set up a T_1 experiment (M)

Syntax `dot1<(min_T1_estimate,max_T1_estimate,time)>`

Description Sets up all parameters to perform a T_1 experiment, including d1, pw, p1, nt, and an array of d2 values, based on information entered you enter. Make sure that the parameter pw90 is set properly and contains the correctly calibrated 90° pulse width because dot1 uses this information. If you have not done a pulse width calibration recently, you may wish to do so now.

Minimum and maximum T_1 for the peaks of interest are estimates. Do the best you can. Your estimates are used to select optimum values of d2. If the T_1 does not fall between your two guesses, your experiment may not be optimum, but it should still be usable unless your estimates are extremely far off. When you are satisfied with the parameters, enter ga or au to acquire the data.

Arguments `min_T1_estimate` is the estimated minimum expected T_1. The default is the system prompts the user for the value.

`max_T1_estimate` is the estimated maximum expected T_1. The default is the system prompts the user for the value.

`time` is the total time in hours that the experiment should take. The default is the system prompts the user for the value.

Examples
```
dot1
dot1(1,2,.5)
```

See also *NMR Spectroscopy User Guide*

Related

d1	First delay (P)
d2	Incremented delay in 1st indirectly detected dimension (P)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
nt	Number of transients (P)
p1	First pulse width (P)
pw	Pulse width (P)
pw90	90° pulse width (P)

dotflag Display FID as connected dots (P)

Description When sparse FID data points are displayed, they are displayed as unconnected dots. If `dotflag` exists and is set to `'n'`, the FID dots will be connected. To create `dotflag`, enter `create('dotflag','flag')`. To create `dotflag` and the FID display parameters `axisf`, `vpf`, `vpfi`, `crf`, and `deltaf` (if the parameter set is older and lacks these parameters), enter `addpar('fid')`.

Values `'n'` sets connecting the dots. `'y'` sets not connecting the dots.

See also *NMR Spectroscopy User Guide*

Related

`addpar`	Add selected parameters to the current experiment (M)
`create`	Create new parameter in a parameter tree (C)
`df`	Display a single FID (C)

dousermacro Mechanism to provide customization to VnmrJ operations

Syntax `dousermacro('rootName' <,args>)`

Applicability VnmrJ 3.1

Description Certain VnmrJ operations have software hooks to allow for easy user customization. For example, the svf operation will call a macro named `usersvf`, if it exists. That `usersvf` macro could copy additional files into the .fid directory, write a log file, or email a message. It is up to the user to decide how they may want to customize the operation.

The mechanism we use to provide this customization is `dousermacro`. This macro is often called with the syntax `dousermacro($0)` where $0 is the name of the macro being executed (svf in the example above.) The `dousermacro` prepends the string 'user' to the first passed argument and then checks if that macro exists. If it does, it is executed. If any additional arguments are passed to `dousermacro`, these are passed along to the `'user'+rootName` macro.

Some of the operations that have these `dousermacro` hooks include:

- bootup
- calibrate
- operatorlogin
- operatorlogout
- plot
- process
- rt
- rtp
- savefid
- svf
- updateprobe

Creating a local macro named, for example, userplot will allow customization any time the plot macro is called. There are several other macros that call dousermacro. They generally require a fairly good understanding of how these other macros are used in order to effectively use the dousermacro tool. You can find all the macros that call dousermacro by executing: grep dousermacro /vnmr/maclib/* from a shell tool.

downsamp Downsampling factor applied after digital filtering (P)

Description Specifies the downsampling factor applied after digital filtering. The spectral width of the data set after digital filtering and downsampling is sw divided by downsamp, where sw is the acquired spectral width. If downsamp does not exist in the current experiment, enter addpar('downsamp') to add it. addpar('downsamp') creates the digital filtering and downsampling parameters downsamp, dscoef, dsfb, dslsfrq, and filtfile.

Values Number for the downsampling factor. 1 sets digital filtering with a filter bandwidth specified by dsfb without downsampling.

'n' sets normal data processing without digital filtering.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to current experiment (M)
digfilt	Write digitally filtered FID to another experiment (M)
dscoef	Digital filter coefficients for downsampling (P)
dsfb	Digital filter bandwidth for downsampling (P)
dslsfrq	Bandpass filter offset for downsampling (P)
filtfile	File of FIR digital filter coefficients (P)
pards	Create additional parameters used by downsampling (M)
sw	Spectral width in directly detected dimension (P)

dp Double precision (P)

Description Sets whether data are acquired in a 16-bit or 32-bit integer format.

Values 'n' sets 16-bit format, 'y' sets 32-bit format. If the 200-kHz receiver option is installed (Max. Narrowband Width set to 200 kHz in the Spectrometer Configuration window), dp is forced to 'n' if 120000<sw<=200000. If sw>200000, dp is forced to 'y'. On wideline systems, dp='y' is required when sw>100000.

See also *NMR Spectroscopy User Guide*

Related

sw	Spectral width in directly detected dimension (P)

dpcon Display plotted contours (C)

Syntax dpcon(<options,><levels,spacing>)

Description Produces a true contour plot display.

Arguments options must precede levels and spacing in the argument list and can be one or more of the following:

- 'pos' is a keyword to limit the display to positive peaks only in phased spectra. The default is both positive and negative peaks.
- 'neg' is a keyword to limit the display to negative peaks only in phased spectra.
- 'noaxis' is a keyword to omit outlining the display and drawing the horizontal or vertical axis.

levels is the maximum number of contours to be shown. The default is 4.

spacing is the spacing by relative intensity of successive contour levels. The default is 2.

Examples
```
dpcon
dpcon('pos',6)
dpcon(15,1.4)
```

See also *NMR Spectroscopy User Guide*

Related

dcon	Display noninteractive color intensity map (C)
dconi	Control display selection for the dconi program (P)
dpconn	Display plotted contours without screen erase (C)
pcon	Plot contours on plotter (C)

dpconn Display plotted contours without screen erase (C)

Syntax dpconn(<options,><levels,spacing>)

Description Produces a true contour plot display exactly the same as the dpcon command, but without erasing the screen before drawing. The arguments are entered the same as dpcon.

See also *NMR Spectroscopy User Guide*

Related

dpcon	Display plotted contours (C)

dpf Display peak frequencies over spectrum (C)

Syntax (1) dpf<(<'noll'><,'pos'><,noise_mult><,'top'>)>
(2) dpf<(<'noll'><,'pos'><,noise_mult><,'leader'>
<,length>)>

Description Displays peak frequencies in the graphics window, with units specified by the axis parameter. Only those peaks greater than th high are selected. If the interactive command ds is active, dpf deactivates it.

Two basic modes of label positioning are available: labels placed at the top, with *long leaders* extending down to the tops of the lines (syntax 1 using 'top' keyword) or labels positioned just above each peak, with *short leaders* (syntax 2 using 'leader' keyword). The default is short leaders.

Arguments 'noll' is a keyword to display frequencies using last previous line listing.

'pos' (or 'noneg') is a keyword to display positive peaks only.

noise_mult is a numerical value that determines the number of noise peaks displayed for broad, noisy peaks. The default is 3. A smaller value results in more peaks, a larger value results in fewer peaks, and a value of 0.0 results in a line listing containing all peaks above the threshold th. Negative values of noise_mult are changed to a value of 3. The noise_mult argument is inactive when the 'noll' keyword is specified.

'top' is a keyword to display peak labels at the top with long leaders. In this mode, the height of labels is varied by changing the parameter wc2.

'leader' is a keyword to display labels positioned just above each peak.

length specifies the leader length, in mm, if labels are positioned just above each peak. The default is 20.

Examples

```
dpf('pos')
dpf('leader',30)
dpf('top','noll')
dpf('pos',0.0,'leader',30)
```

See also *NMR Spectroscopy User Guide*

Related

axis	Axis label for displays and plots (P)
dpir	Display integral amplitudes below spectrum (C)
dpirn	Display normalized integral amplitudes below spectrum (M)
pir	Plot integral amplitudes below spectrum (C)
pirn	Plot normalized integral amplitudes below spectrum (M)
ppf	Plot peak frequencies over spectrum (M)
th	Threshold (P)
vp	Vertical position of spectrum (P)
wc2	Width of chart in second direction (P)

dpir Display integral amplitudes below spectrum (C)

Description Displays integral amplitudes below the appropriate spectral regions.

See also *NMR Spectroscopy User Guide*

Related		
	dpf	Display peak frequencies over spectrum (C)
	dpirn	Display normalized integral amplitudes below spectrum (M)
	pir	Plot integral amplitudes below spectrum (C)
	pirn	Plot normalized integral amplitudes below spectrum (M)
	ppf	Plot peak frequencies over spectrum (M)

dpirn Display normalized integral amplitudes below spectrum (M)

Description Equivalent to the command dpir except that the sum of the integrals is normalized to the value of the parameter ins.

See also *NMR Spectroscopy User Guide*

Related		
	dpir	Display integral amplitudes below spectrum (C)
	ins	Integral normalization scale (P)
	pirn	Plot normalized integral amplitudes below spectrum (M)

dpiv Display integral values below spectrum (M)

Syntax `dpiv<(vertical_position)>`

Description Labels integrals with a bracket below the spectrum and a vertical number indicating the integral value.

- vertical labels for narrower regions
- avoids label overlap by label shifting
- more flexible vertical positioning

The vertical position defaults to a location just underneath the scale labels, assuming there is enough room below the scale. If the vertical position is too low, the vertical position is allowed to approach the position of the spectrum up to 1 mm. If the spectral position is so low that the integral labels would overlap with the spectrum, an error message is produced (indicating the minimum vp), and the command aborts. No error message is produced in case of overlap with the scale. The minimum for vp depends on the plotter and the character size, and in the case of dpiv also on the size of the graphics window.

Use an optional argument to force the vertical position to any value; no checking is done, and no error message is produced in case of overlap. piv(vp-2) produces integral labels with the brackets ending 2 mm below the position of the spectrum.

`dpiv` follows this convention: the output is controlled by `ins` and `insref` and not by `is`. Restore the `is` integration mode by creating a (local or global) parameter `oldint` and set `oldint=` 'y':

```
create('oldint','flag','global')
oldint='y'
```

`oldint='n'` (or destroy the parameter) switches back to the default integration mode.

Examples `vp=25 dpiv`

`vp=50 pl pscale piv(0)`

Related

dpir	Display integral amplitudes below spectrum (C)
dpirn	Display normalized integral amplitudes below spectrum (C)
dpivn	Display normalized integral amplitudes below spectrum (M)
pirn	Plot normalized integral amplitudes below spectrum (C)
pir	Plot integral amplitudes below spectrum (C)
piv	Plot integral amplitudes below spectrum (M)
pivn	Plot normalized integral amplitudes below spectrum (M)

dpivn Display normalized integral values below spectrum (M)

Syntax `dpivn<(vertical_position)>`

Description Labels integrals with a bracket below the spectrum and a vertical number indicating the integral value.

See dpiv for description and use.

Related

dpir	Display integral amplitudes below spectrum (C)
dpirn	Display normalized integral amplitudes below spectrum (C)
dpiv	Display integral amplitudes below spectrum (M)
pirn	Plot normalized integral amplitudes below spectrum (C)
pir	Plot integral amplitudes below spectrum (C)
piv	Plot integral amplitudes below spectrum (M)
pivn	Plot normalized integral amplitudes below spectrum (M)

dpl Default plot (M)

Description Looks for sequence-specific default plot macro (`dpl_`*`seqfil`*) and executes if one is found.

Related

dpl_seqfil	Sequence-specific default plot (M)
dpr	Default process (M)
dds	Default display (M)

dpl_*seqfil* Sequence-specific default plot (M)

Description Sequence-specific default plot. These macros are called by the `dpl` macro.

Examples `dpl_NOESY1D`
`dpl_TOCSY1D`

Related		
	dpl	Default plot (M)
	dpr	Default process (M)
	dds	Default display (M)

dplane Display a 3D plane (M)

Syntax `dplane(<plane_type,>plane_number)`

Description Displays the 2D color map of a particular data plane from a 3D spectral data set. The 3D parameters are loaded into VnmrJ each time `dplane` is executed. The parameter `path3d` specifies the absolute path to the directory (without the `.extr` file extension) where the 2D planes extracted from the 3D spectral data set reside.

Arguments `plane_type` is one of the keywords `'f1f3'`, `'f2f3'`, and `'f1f2'` for the f_1f_3, f_2,f_3, and f_1f_2 planes, respectively. If `plane_type` is specified, the parameter `plane` is updated with that new value. `plane` is then used to determine the type of 3D plane to be displayed.

`plane_number` specifies which plane of a particular type is to be displayed:

- For plane f_1f_3, the range of plane_number is `1` to `fn2/2`
- For plane f_2f_3, the range of plane_number is `1` to `fn1/2`
- For plane f_1f_2, the range of plane_number is `1` to `fn/2`

Examples `dplane(3)`
`dplane('f1f2',2)`

See also *NMR Spectroscopy User Guide*

Related		
	dsplanes	Display a series of 3D planes (M)
	dproj	Display a 3D plane projection (M)
	getplane	Extract planes from a 3D spectral data set (M)
	nextpl	Display the next 3D plane (M)
	path3d	Path to currently displayed 2D planes from a 3D data set (P)
	plane	Currently displayed 3D plane type (P)
	prevpl	Display the previous 3D plane (M)
	plplanes	Plot a series of 3D planes (M)

dpr Default process (M)

Description Looks for sequence-specific default plot macro (`dpr_seqfil`) and executes if one is found.

Related
`dpr_seqfil` Sequence-specific default process (M)
`dpl` Default plot (M)
`dds` Default display (M)

dpr_*seqfil* Sequence-specific default process (M)

Description Sequence-specific default plot. These macros are called by the `dpr` macro.

Examples
```
dpr_NOESY1D
dpr_TOCSY1D
```

Related
`dpr` Default process (M)
`dpl` Default plot (M)
`dds` Default display (M)

dprofile Display pulse excitation profile (M)

Syntax `dprofile<(axisflag<,profile<,shapefile>>)>`

Description Displays the X, Y and Z excitation (inversion) profile for a pulse shape generated by the Pbox software. If `shapefile` is not provided, the last simulation data stored in the `shapelib/pbox.sim` file are displayed.

Arguments The `axisflag` and `profile` arguments can be given in any order.

`axisflag` is `'y'` to display the full spectrum and a frequency scale, or `'n'` to suppress the scale and spectrum. The default is `'n'`.

`profile` is a character string identifying the desired profile. `'xyz'` selects X, Y, and Z (inversion) profiles; `'xy'` selects only the excitation (transverse) profiles; `'x'` selects only the X transverse excitation profile; and `'z'` selects only the inversion profile. The default is `'xyz'`.

`shapefile` is the name of a *.RF or *.DEC file, including the extension.

Examples
```
dprofile
dprofile('y','xy')
dprofile('xy','n','softpls.RF')
```

See also *NMR Spectroscopy User Guide*

Related
`pprofile` Plot pulse excitation profile (M)
`Pbox` Pulse shaping software (U)

dproj Display a 3D plane projection (M)

Syntax `dproj<(plane_type)>`

Description Displays 2D color map of the 2D projection plane from a 3D spectral data set. The projection is a skyline projection. The 3D parameters are loaded into VnmrJ each time `dproj` is executed. For this macro, the parameter `path3d` specifies the directory (without the `.extr` extension) where the 2D projection resides that has been created from the 3D spectral data set.

Arguments `plane_type` is one of the keywords `'f1f3'`, `'f2f3'`, and `'f1f2'` for the f_1f_3, f_2,f_3, and f_1f_2 planes, respectively. If `plane_type` is specified, the parameter `plane` is updated with that value. `plane` is then used to determine the type of 2D projection to be displayed.

Examples
```
dproj
dproj('f1f2')
```

See also *NMR Spectroscopy User Guide*

Related

`dplane`	Display a 3D plane (M)
`dsplanes`	Display a series of 3D planes (M)
`getplane`	Extract planes from a 3D spectral data set (M)
`nextpl`	Display the next 3D plane (M)
`path3d`	Path to currently displayed 2D planes from a 3D data set (P)
`plane`	Currently displayed 3D plane type (P)
`plplanes`	Plot a series of 3D planes (M)
`prevpl`	Display the previous 3D plane (M)

dps Display pulse sequence (C)

Syntax `dps<(file),x,y,width,height>`

Description Displays a picture of pulse sequences consisting of three to five parts. The top part is the transmitter pulse sequence (Tx). The second part is the decoupler pulse sequence (Dec). The third part might be the second or third decoupler (Dec2 or Dec3) pulse sequence or gradients (X, Y, or Z), depending on the program. The lowest part is the status.

The pulse parameters are displayed if there is enough space an if the length of the parameter name is less than thirty letters. The value of each pulse is also displayed. If the value delay or width is less than zero, a question mark (?) is displayed. The time units are displayed in color (on a color monitor). The height of pulses is scaled according to their power level.

`dps` also displays spin lock, transmitter gating, observe transmitter power, and other information.

Arguments `file` specifies the name of the file containing the pulse sequences. The default is the file `seqfil`.

x,y specifies the start of the position with respect to the lower-left corner of the window.

width,height are in proportion to wcmax and wc2max.

See also *NMR Spectroscopy User Guide*

Related

pps	Plot pulse sequence (C)
seqfil	Pulse sequence name (P)
wc	Width of chart (P)
wcmax	Maximum width of chart (P)
wc2max	Maximum width of chart in second direction (P)

dpwr Power level for first decoupler with linear amplifier (P)

Applicability Systems with a linear amplifier.

Description On systems equipped with a linear amplifier, a 63-dB or 79-dB attenuator between the decoupler transmitter and the amplifier controls the power level.

The system value for the attenuator upper safety limit is set fin the Spectrometer Configuration window (opened by config). The Upper Limit entry sets this value. For broadband decoupling of ^{1}H nuclei, typical values range from 36 to 49 dB. For homonuclear decoupling, typical values range from 5 to 15 dB.

Values 79 dB, -16 to +63, in steps of 1 dB.

Decoupler power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate decoupling to avoid exceeding 2 watts. The maximum value for dpwr on a 200-, 300-, or 400-MHz system with a linear amplifier on the decoupler channel has been set to 49, corresponding to about 2 watts of power. Before using dpwr=49 for continuous decoupling, ensure safe operation by measuring the output power. This should be done during system installation and checked periodically by the user.

See also *VnmrJ Installation and Administration*

Related

cattn	Coarse attenuator (P)
config	Display current configuration and possible change it (M)
dpwrf	First decoupler fine power (P)
dpwr2	Power level for second decoupler (P)
dpwr3	Power level for third decoupler (P)
dpwr4	Power level for fourth decoupler (P)
fattn	Fine attenuator (P)
tpwr	Power level of observe transmitter with linear amplifiers (P)
tpwrf	Observe transmitter fine power (P)

dpwr2 Power level for second decoupler with linear amplifier (P)

Applicability Systems with a linear amplifier as the second decoupler.

Description Controls the coarse attenuator (63 dB or 79 dB) that resides between the transmitter board and the linear amplifier associated with the second decoupler. The system value for the attenuator upper safety limit is set in the Spectrometer Configuration window (opened by `config`).

Values 79 dB, -16 to +63, in steps of 1 dB.

If `dn2=''` (two single quotes) and a second decoupler channel is present in the console, `dpwr2` assumes a default value of `0` when `go` is executed.

CAUTION Decoupler power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate decoupling to avoid exceeding 2 watts. The maximum value for dpwr2 on a 200-, 300-, or 400-MHz system with a linear amplifier on the decoupler channel has been set to 49, corresponding to about 2 watts of power. Before using dpwr2=49 for continuous decoupling, ensure safe operation by measuring the output power. This should be done during system installation and checked periodically by the user.

See also *NMR Spectroscopy User Guide*

Related

Related	`cattn`	Coarse attenuator type (P)
	`config`	Display current configuration and possible change it (M)
	`dn2`	Nucleus for second decoupler (P)

dpwr3 Power level for third decoupler with linear amplifier (P)

Applicability Systems with a linear amplifier as the third decoupler.

Description Controls the coarse attenuator (63 dB or 79 dB) that resides between the transmitter board and the linear amplifier associated with the third decoupler. The system value for the attenuator upper safety limit is set in the Spectrometer Configuration window (opened by `config`).

Values If 63-dB attenuator installed: 0 to 63 (63 is max. power), in units of dB. If 79-dB attenuator installed: –16 to 63 (63 is max. power), in units of dB. If `dn3=''` (two single quotes) and a third decoupler channel is present in the console, `dpwr3` assumes a default value of `0` when `go` is executed.

CAUTION Decoupler power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate decoupling to avoid exceeding 2 watts. The maximum value for dpwr3 on a 200-, 300-, or 400-MHz system with a linear amplifier on the decoupler channel has been set to 49, corresponding to about 2 watts of power. Before using dpwr3=49 for continuous decoupling, ensure safe operation by measuring the output power. This should be done during system installation and checked periodically by the user.

See also *NMR Spectroscopy User Guide*

Related	`cattn`	Coarse attenuator type (P)
	`config`	Display current configuration and possible change it (M)
	`dn3`	Nucleus for third decoupler (P)

`dpwr4` Power level for fourth decoupler amplifier (P)

Applicability Systems with deuterium decoupler channel as the fourth decoupler.

Description Controls the coarse attenuator (45 dB range) that resides on the Lock Transceiver board and the amplifier associated with the fourth decoupler. The system value for the attenuator upper safety limit is set in the Spectrometer Configuration window (opened by `config`).

Values 48-dB attenuator: 15 to 63 (63 is max. power), in units of dB. If `dn4=''` (two single quotes) and a third decoupler channel is present in the console, `dpwr4` assumes a default value of `0` when `go` is executed.

CAUTION Decoupling power greater than 5 watts applied to a triple-resonance probe will damage the probe. The maximum value for dpwr4 is 63, corresponding to about 35 watts to the probe. A value of dpwr4 equal to 52 corresponds to about 5 watts and will produce approximately a 1 kHz decoupling field. Always carefully calibrate decoupling power to avoid exceeding 5 watts. Before using dpwr4=52 continuous decoupling, ensure safe operation by measuring the output power. Measurement should be taken during system installation and checked periodically by the user.

See also *NMR Spectroscopy User Guide*

Related	`cattn`	Coarse attenuator type (P)
	`config`	Display current configuration and possible change it (M)
	`dn3`	Nucleus for third decoupler (P)

dpwrf First decoupler fine power (P)

Applicability Systems with an optional fine attenuator on the decoupler channel.

Description Controls the first decouple fine attenuator. Systems with this attenuator are designated within the Spectrometer Configuration window (opened by `config`) by the status of the Fine Attenuator entry. The fine attenuator is linear and spans 6 dB.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrf` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming, User Guide: Solids; CP/MAS Installation,*

Related

`config`	Display current configuration and possibly change it (M)
`dpwr`	Power level for first decoupler with linear amplifiers (P)
`dpwrf2`	Second decoupler fine power (P)
`dpwrf3`	Third decoupler fine power (P)
`dpwrm`	First decoupler linear modulator power (P)
`fattn`	Fine attenuator (P)
`tpwr`	Power level of observe transmitter with linear amplifiers (P)
`tpwrf`	Transmitter fine power (P)

dpwrf2 Second decoupler fine power (P)

Applicability Systems with an optional fine attenuator on the second decoupler channel.

Description Controls the second decoupler fine attenuator, functioning analogously to `dpwrf`.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrf2` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming*

Related `dpwrf` First decoupler fine power (P)

dpwrf3 Third decoupler fine power (P)

Applicability Systems with an optional fine attenuator on the third decoupler channel.

Description Controls the third decoupler fine attenuator, functioning analogously to `dpwrf`.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrf3` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming*

Related `dpwrf` First decoupler fine power (P)

dpwrm First decoupler linear modulator power (P)

Applicability Systems with a first decoupler linear modulator.
The fine power control is linear and spans 0 to `dpwr`.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrm` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming; User Guide: Solids; CP/MAS Installation*

Related
`dpwrm2` Second decoupler linear modulator power (P)
`dpwrm3` Third decoupler linear modulator power (P)
`tpwrm` Observe transmitter linear modulator power (P)

dpwrm2 Second decoupler linear modulator power (P)

Applicability Systems with a second decoupler linear modulator.

Description Controls the second decoupler linear modulator systems.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrm2` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming*

Related `dpwrm` First decoupler linear modulator power (P)

dpwrm3 Third decoupler linear modulator power (P)

Applicability Systems with a third decoupler linear modulator.

Description Controls the third decoupler linear modulator systems.

Values 0 to 4095 (where 4095 is maximum power). If `dpwrm3` does not exist in the parameter table, a value of 4095 is assumed.

See also *User Programming*

Related `dpwrm` First decoupler linear modulator power (P)

Dqcosy Convert the parameter to a DQCOSY experiment (M)

Description Convert the parameter to a double-quantum filtered (DQCOSY) experiment

See also *NMR Spectroscopy User Guide*

Related
`cosyps` Set up parameters for phase-sensitive COSY (M)
`Cosy` Set up parameters for COSY pulse sequence (M)
`relayh` Set up parameters for COSY pulse sequence (M)

draw — Draw line from current location to another location (C)

Syntax `draw(<'keywords'>x,y)`

Description Draws a line from the current location to the absolute location with coordinates given by the arguments.

Arguments `'keywords'` identifies the output device (`'graphics'|'plotter'`), drawing mode (`'xor'|'normal'`), and drawing capability (`'newovly'|'ovly'|'ovlyC'`).

- `'graphics'|'plotter'` is a keyword for the output device. The default is `'plotter'`. The output selected is passed to subsequent `pen`, `move`, or `draw` commands and remains active until a different output is specified.
- `'xor'`,`'normal'` is a keyword for the drawing mode when using the `'graphics'` output device. The default is `'normal'`. In the `'xor'` mode, if a line is drawn such that one or more points of the line are in common with a previous `'xor'` line, the common points are erased. In the `normal` mode, the common points remain. The mode selected is passed to subsequent `draw`, `pen`, and `move` commands and remains active until a different mode is specified.
- `'newovly'`, `'ovly'`, and `'ovlyC'` are keywords that specify an interactive drawing capability that is slightly slower than the `'xor'` mode but more consistent in color. `'newovly'` clears any previous draws, boxes, and writes made with the `'ovly'` modes and draws the figure. `'ovly'` draws without clearing so that multisegment figures can be created. `'ovlyC'` clears without drawing.

`x,y` are the absolute coordinates, in mm, of the endpoint of the line to be drawn. The range of `x` is `0` at the left edge of the chart and `wcmax` at the right edge. The range of `y` is `-20` at the bottom of the chart and `wc2max` at the top.

Examples
```
draw('graphics','xor'.wcmax-sc,vp+th)
draw(wcmax-sc-wc*(cr-delta-sp)/wp,wc2max)
```

See also *NMR Spectroscopy User Guide*

Related

`gin`	Return current mouse position and button values (C)
`move`	Move to an absolute location (C)
`pen`	Select a pen or color for drawing (C)
`wcmax`	Maximum width of chart (P)
`wc2max`	Maximum width of chart in second direction (P))

dres — Measure linewidth and digital resolution (C)

Syntax
```
dres<(<freq<,fractional_height>>)>
   :linewidth,digital_resolution
```

Description Analyzes the line defined by the current cursor position for its linewidth (width at half-height) and digital resolution.

Arguments freq is the frequency of the line. The default is the parameter cr. This overrides using the current cursor position as the frequency.

fractional_height is the linewidth is measured at this height.

linewidth is the value returned for the linewidth of the line.

digital_resolution is the value returned for the digital resolution of the line.

Examples
```
dres:$width,$res
dres(cr,0.55)
```

See also *NMR Spectroscopy User Guide; User Programming*

Related

cr	Current cursor position (P)
dsn	Measure signal-to-noise (C)

dres Tip-angle resolution for first decoupler (P)

Applicability Systems with waveform generators.

Description Controls the tip-angle resolution to be used within a waveform generator decoupling sequence on the first decoupler. The optimum value is a function of the decoupling sequence to be used: for WALTZ-16, dres=90.0; for MLEV16-240, dres=30.0; and for GARP1, dres=1.0.

Values 1.0 to 90.0, in units of degrees. In reality, dres can assume values as small of 0.7 (but no smaller) and can be specified in units of 0.1°. To use this capability, change the limits of dres by using destroy('dres') create('dres','real') setlimit('dres',360,0.7,0.1). Making corresponding changes within the fixpar macro ensures that dres is created in the desired way with each new parameter set.

See also *NMR Spectroscopy User Guide*

Related

dmfadj	Adjust decoupler tip-angle resolution time (M)
dres2	Tip angle resolution for second decoupler (P)
dres3	Tip angle resolution for third decoupler (P)
fixpar	Correct parameter characteristics in experiment (M)

dres2 Tip-angle resolution for second decoupler (P)

Applicability Systems with waveform generators.

Description Controls the tip-angle resolution to be used within a waveform generator decoupling sequence on the second decoupler. The optimum value is a function of the decoupling sequence to be used: for WALTZ-16, dres2=90.0; for MLEV16-240, dres2=30.0; and for GARP1, dres2=1.0.

Values 1.0 to 90.0, in units of degrees.

See also *NMR Spectroscopy User Guide*

Related
dmf2adj Adjust second decoupler tip-angle resolution time (M)
dres Tip-angle resolution for first decoupler (P)

dres3 Tip-angle resolution for third decoupler (P)

Applicability Systems with waveform generators.

Description Controls the tip-angle resolution to be used within a waveform generator decoupling sequence on the third decoupler. The optimum value is a function of the decoupling sequence to be used: for WALTZ-16, `dres3=90.0`; for MLEV16-240, `dres3=30.0`; and for GARP1, `dres3=1.0`.

Values 1.0 to 90.0, in units of degrees.

See also *NMR Spectroscopy User Guide*

Related
dmf3adj Adjust third decoupler tip-angle resolution time (M)
dres Tip-angle resolution for first decoupler (P)

dres4 Tip-angle resolution for fourth decoupler (P)

Applicability Systems with deuterium decoupler channel as the fourth decoupler.

Description Controls the tip-angle resolution to be used for the decoupling sequence on the fourth decoupler. The optimum value is a function of the decoupling sequence to be used: for WALTZ-16, `dres4=90.0`; for MLEV16-240, `dres4=30.0`; and for GARP1, `dres4=1.0`.

Values 1.0 to 90.0, in units of degrees.

See also *NMR Spectroscopy User Guide*

Related
dmf4adj Adjust fourth decoupler tip-angle resolution time (M)
dres Tip-angle resolution for first decoupler (P)

ds Display a spectrum (C)

Syntax (1) `ds<(index)>`
(2) `ds<(options)>`

Description Displays a single spectrum. Parameter `intmod` controls integral display:

- `intmod='off'` turns off the integral display
- `intmod='full'` displays the entire integral
- `intmod='partial'` displays every other integral region

Parameter entry after a spectrum has been displayed with the ds command causes the spectrum to be updated.

Two additional parameters control the behavior of the ds command:

- The parameter phasing (in the "global" parameter set) controls the percentage of the spectrum updated during interactive phasing. This parameter can be set in the range of 10 to 100. A value of 100 causes the entire spectrum to be updated. A value of 20 causes the area between the two horizontal cursors to be updated.
- The parameter lvltlt (in the "current" parameter set) controls the sensitivity of the interactive lvl and tlt adjustments. lvltlt can be set to any positive real number. It is basically a multiplier for the sensitivity. The default value is 1.0. Larger values make the adjustments larger. Smaller values make the adjustments smaller.

For arrayed 1D spectra or for 2D spectra, a particular trace can be viewed by supplying the index number as an argument. For 2D data sets, spectra can be displayed from either the f_1 or f_2 domain by setting the parameter trace equal to 'f1' or 'f2', respectively. After entering ft1d, interferograms can be viewed by setting trace='f1' and then typing ds.

Spectra are scaled according to the number of completed transients ct. If nt is arrayed (nt=1,2,4,8), each spectrum is scaled by its own ct.

Arguments index (used with syntax 1) is the index number of a particular trace to be displayed in arrayed 1D spectra or in 2D spectra (syntax 1).

options (used with syntax 2) is any of the following keywords:

- 'toggle' switches between the box and the cursor modes.
- 'restart' redraws the cursor if it has been turned off.
- 'expand' toggles between expanded and full view of the spectrum.
- 'spwp' interactively adjusts start and width of the spectrum display.
- 'phase' enters an interactive phasing mode.
- 'thresh' interactively adjusts the threshold.
- 'z' interactively sets integral resets.
- 'dscale' toggles the scale below the spectrum on and off.
- 'lvltlt' interactively adjusts the lvl and tlt parameters.
- 'scwc' interactively adjusts the start and width of chart.
- 'noclear' start or restart the ds display without clearing the graphics screen
- 'exists' exit the ds display, leaving a non-interactive dss display.

Examples
```
ds
ds(7)
ds('restart')
```

See also *NMR Spectroscopy User Guide*

Related		
	crmode	Current state of cursors in dfid, ds, or dconi (P)
	ct	Completed transients (P)
	exists	

ft1d	Fourier transform along f_2 dimension (C)
intmod	Integral display mode (P)
lp	First-order phase in directly detected dimension (P)
lvl	Zero-order baseline correction (P)
lvltlt	Control sensitivity of lvl and tlt adjustments (P)
nt	Number of transients (P)
phasing	Control update region during ds phasing (P)
rp	Zero-order phase in directly detected dimension (P)
select	Select a spectrum without displaying It (C)
tlt	First-order baseline correction (P)
trace	Mode for n-dimensional data display (P)
wft1d	Weight and Fourier transform f_2 for 2D data (C)

ds2d — Display 2D spectra in whitewash mode (C)

Syntax `ds2d<(options)>`

Description Displays a stacked plot of 2D spectra in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind an earlier spectra). Color does not represent intensity (unlike dcon), because intensity can be seen visually, but instead successive traces are displayed in different colors so that color represents frequency.

Arguments `options` can be any of the following keywords:

- `'nobase'` is a keyword to activate the th parameter to suppress all intensity below the th level.
- `'fill'` is a keyword to fill in the peaks. When using `'fill'`, th operates linearly and not logarithmically (factors of 2) as it does in the contour or color intensity displays.
- `'fillnb'` is a keyword to combine base suppression and peak filling. When using `'fillnb'`, th operates linearly and not logarithmically (factors of 2) as it does in the contour or color intensity displays.
- `'noaxis'` is a keyword to omit outlining the display and drawing the horizontal and vertical axis.

Examples
```
ds2d
ds2d('fillnb')
```

See also *NMR Spectroscopy User Guide*

Related

dcon	Display noninteractive color intensity map (C)
dconi	Control display selection for the dconi program (P)
ds2dn	Display 2D spectra in whitewash mode without screen erase (C)
pl2d	Plot 2D spectra in whitewash mode (C)
th	Threshold (P)

ds2dn — Display 2D spectra in whitewash mode without screen erase (C)

Syntax ds2dn<(options)>

Description Displays a stacked plot of 2D spectra in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind an earlier spectra) the same as ds2d but without erasing the screen before drawing. The arguments are the same as ds2d.

Examples ds2dn
ds2dn('fillnb')

See also *NMR Spectroscopy User Guide*

Related ds2d — Display 2D spectra in whitewash mode (C)

dsnarray — Report statistical signal-to-noise for Cold Probes (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Report the statistical S/N of a series of repeated gNhsqc data sets acquired with a labeled protein sample.

dscale — Display scale below spectrum or FID (C)

Syntax dscale<(<rev><,axis><,label><,vp0><,sp0><,color><,pen>)>

Description Displays a scale under a spectrum or FID.

Arguments rev – reverses the direction of the scale. That is, the smaller numbers will be at the left side of the scale. If used, 'rev' must be the first argument.

axis – If the letter p, h, k, etc. is supplied, it will be used instead of the current value of the parameter axis. For an FID scale, if the letter s, m, or u is supplied, it will be used instead of the current value of the parameter axisf.

label – If a string of 2 or more characters is supplied, it will be used as the axis label.

vp0 – This is supplied as the first real number. It defines the vertical position where the scale is drawn. The default is 5 mm below the current value of the parameter vp.

sp0 – This is supplied as the second real number. It is a modified start of plot. If, for example, the display is from 347 to 447 hz, but the scale is desired to read 0 to 100 hz., sp0 would be input as 0.

wp0 – This is supplied as the third real number. It is a modified width of plot. If, for example, the display is from 347 to 447 hz, but the scale is desired to read 0 to 550 Units. sp0 would be input as 0, wp0 would be 550, and the label would be 'Units'.

An optional color or pen number can be supplied to dscale or pscale. The available colors and pens are: 'red', 'green', 'blue', 'cyan', 'magenta', 'yellow', 'black', 'white'
'pen1', 'pen2', 'pen3',..., 'pen8'

Examples
```
dscale
dscale('rev')
dscale('h',0,'green')
dscale('h',vp-10,0)
```

See also *NMR Spectroscopy User Guide*

Related

axis	Axis label for displays and plots (P)
axisf	Axis label for FID displays and plots (P)
pscale	Plot scale below spectrum or FID (C)
vp	Vertical position of spectrum (P)

dscoef — Digital filter coefficients for downsampling (P)

Description: Specifies the number of coefficients used in the digital filter. This parameter does not need to be changed as the parameter downsamp is changed, because dscoef is automatically adjusted by VnmrJ to give filter cutoffs that are the same, regardless of the value of downsamp. This is done by using dscoef*downsamp/2 coefficients in the digital filter. VnmrJ always rounds dscoef*downsamp/2 to an odd number. If dscoef does not exist in the current experiment, enter addpar('downsamp') to add it. Entering addpar('downsamp') creates the digital filtering and downsampling parameters downsamp, dscoef, dsfb, dslsfrq, and filtfile.

Values: Number of digital filter coefficients. The default is 61. A larger number of coefficients gives a filter with sharper cutoffs; a smaller number gives a filter with more gradual cutoffs.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to current experiment (M)
downsamp	Downsampling factor applied after digital filtering (P)
dsfb	Digital filter bandwidth for downsampling (P)
dslsfrq	Bandpass filter offset for downsampling (P)
filtfile	File of FIR digital filter coefficients (P)
pards	Create additional parameters used for downsampling (M)

dseq — Decoupler sequence for first decoupler (P)

Applicability: Systems with waveform generators.

Description: Specifies the decoupling sequence (without the .DEC file extension) to be used during any period of programmable decoupling on the first decoupler under status control (i.e., dmm='p'). The decoupling sequence must be located in the user’s shapelib directory or in the VnmrJ system’s shapelib directory.

See also *NMR Spectroscopy User Guide*

Related
dmm	Decoupler modulation mode for first decoupler (P)
dseq2	Decoupler sequence for second decoupler (P)
dseq3	Decoupler sequence for third decoupler (P)

dseq2 Decoupler sequence for second decoupler (P)

Applicability Systems with waveform generators.

Description Specifies the decoupling sequence (without the .DEC file extension) to be used during any period of programmable decoupling on the second decoupler under status control (i.e., dmm2='p'). The decoupling sequence must be located in the user’s shapelib directory or in the VnmrJ system shapelib directory.

See also *NMR Spectroscopy User Guide*

Related
dmm2	Decoupler modulation mode for second decoupler (P)
dseq	Decoupler sequence for first decoupler (P)

dseq3 Decoupler sequence for third decoupler (P)

Applicability Systems with waveform generators.

Description Specifies the decoupling sequence (without the .DEC file extension) to be used during any period of programmable decoupling on the third decoupler under status control (i.e., dmm3='p'). The decoupling sequence must be located in the user's shapelib directory or in the shapelib directory.

See also *NMR Spectroscopy User Guide*

Related
dmm3	Decoupler modulation mode for third decoupler (P)
dseq	Decoupler sequence for first decoupler (P)

dseq4 Decoupler sequence for fourth decoupler (P)

Applicability Systems with waveform generators.

Description Specifies the decoupling sequence (without the .DEC file extension) to be used during any period of programmable decoupling on the third decoupler under status control (i.e., dmm4='p'). The decoupling sequence must be located in the user's shapelib directory or in the system's shapelib directory.

See also *NMR Spectroscopy User Guide*

Related
dmm4	Decoupler modulation mode for third decoupler (P)
dseq	Decoupler sequence for first decoupler (P)

dsfb Digital filter bandwidth for downsampling (P)

Description Specifies the bandwidth of the digital filter used for downsampling. If `dsfb` does not exist in the current experiment, enter `addpar('downsamp')` to add it. `addpar('downsamp')` creates the digital filtering and downsampling parameters `downsamp`, `dscoef`, `dsfb`, `dslsfrq`, and `filtfile`.

Values Number, in Hz. A smaller value rejects frequencies at the spectrum edges; a larger value aliases noise and signals at frequencies outside of ±`sw/2`.

`'n'` makes `dsfb` default to the final `sw/2`.

See also *NMR Spectroscopy User Guide*

Related
addpar	Add selected parameters to current experiment (M)
downsamp	Downsampling factor applied after digital filtering (P)
dscoef	Digital filter coefficients for downsampling (P)
dslsfrq	Bandpass filter offset for downsampling (P)
filtfile	File of FIR digital filter coefficients (P)
pards	Create additional parameters used for downsampling (M)
sw	Spectral width in directly detected dimension (P)

dshape Display pulse shape or modulation pattern (M)

Syntax `dshape<(pattern.ext)>`

Description Displays the real (X) and imaginary (Y) components of a shaped pulse. Any type of waveform (.RF, .DEC or .GRD) can be displayed.

Arguments `pattern` is the name of a shape or pattern file specified by an absolute file name, relative file name, or a simple pattern file name. `ext` is a file name extension that specifies the file type. In the case of a simple file name, `dshape` searches for the file in the local directory, then in the user's `shapelib`, and finally in the directory `/vnmr/shapelib`. If `pattern.ext` is not given, `dshape` displays the last created waveform stored in the `pbox.fid` file.

Examples
```
dshape
dshape('Pbox.RF')
```

See also *NMR Spectroscopy User Guide*

Related
Pbox	Pulse shaping software (U)
pshape	Plot pulse shape or modulation pattern (M)

dshapef Display last generated pulse shape (M)

Description Displays the real (X) and imaginary (Y) components of last generated shaped pulse, stored in `pbox.fid` file.

See also *NMR Spectroscopy User Guide*

Related `Pbox` Pulse shaping software (U)
`pshapef` Plot last generated pulse shape (M)

dshapei Display pulse shape or modulation pattern interactively (M)

Syntax `dshapei<(pattern.ext)>`

Description Displays the real (X) and imaginary (Y) components of a pulse shape, modulation pattern or gradient shape interactively. `dshapei` overwrites the existing data (FID) after the permission is granted by the user. It also asks for the duration of the waveform and displays the timescale.

Arguments `pattern` is the name of a shape or pattern file specified by an absolute file name, relative file name, or a simple pattern file name. `ext` is a file name extension that specifies the file type. In the case of a simple file name, `dshapei` searches for the file in the local directory, then in the user's `shapelib`, and finally in the directory `/vnmr/shapelib`. If no file name is given, `dshapei` displays the last created waveform stored in the `pbox.fid` file.

Examples `dshapei`
`dshapei('myfile.DEC')`

See also *NMR Spectroscopy User Guide*

Related `Pbox` Pulse shaping software (U)

dshim Display a shim “method” string (M)

Syntax (1) `dshim<(file)>`
(2) `dshim('method'|'help')`

Description Looks in the user's `shimmethods` directory and then in the system `shimmethods` directory for a file and displays the file (syntax 1) or displays information about `method` strings (syntax 2).

Arguments `file` is the name of a file to be searched for in the `shimmethods` directories. The default is to display the contents of the `shimmethods` directories.

`'method'` is a keyword to explain the structure of `method` strings.

`'help'` is a keyword to describe the `method` strings in the system’s `shimmethods` directory.

Examples	`dshim`
	`dshim('method')`
	`dshim('help')`	

See also	*NMR Spectroscopy User Guide*	
Related	method	Autoshim method (P)
	newshm	Interactively create a shim “method” with options (M)
	shim	Submit an Autoshim experiment to acquisition (C)
	stdshm	Interactively create a shim “method” (M)

dslsfrq Bandpass filter offset for downsampling (P)

Description	For downsampling, selects a bandpass filter that is not centered about the transmitter frequency. In this way, `dslsfrq` works much like lsfrq. If `dslsfrq` does not exist in the current experiment, add it by entering `addpar('downsamp')`. The command `addpar('downsamp')` creates the digital filtering and downsampling parameters downsamp, dscoef, dsfb, `dslsfrq`, and filtfile.	
Values	A number, in Hz. A positive value selects a region upfield from the transmitter frequency; a negative value selects a downfield region.	
See also	*NMR Spectroscopy User Guide*	
Related	addpar	Add selected parameters to current experiment (M)
	downsamp	Downsampling factor applied after digital filtering (P)
	dscoef	Digital filter coefficients for downsampling (P)
	dsfb	Digital filter bandwidth for downsampling (P)
	filtfile	File of FIR digital filter coefficients (P)
	lsfrq	Frequency shift of the `fn` spectrum in Hz (P)
	movedssw	Set parameters for digital filtering and downsampling (M)
	pards	Create additional parameters used by downsampling (M)

dsn Measure signal-to-noise (C)

Syntax	`dsn<(low_field,high_field)>:signal_to_noise,noise`
Description	Measures the signal-to-noise ratio of the spectrum by first measuring the intensity of the largest peak in the spectral range defined by sp and wp, and then measuring the noise in the spectral region defined by the position of the two cursors. The noise value returned from `dsn` is not scaled by vs. The interrelations between the signal-to-noise ratio, the noise, and peak intensities can be illustrated by comparing `dsn:$sn,$noise` and peak`:$signal`. In this case, `$sn` is equal to `($signal /$noise)/vs`.

Calculate noise by first doing a drift correction on the noise region. Noise is defined as:

$$noise = 2x\left(\left(\sum_{i=1}^{np} Y_i^2\right) / np\right)^{\frac{1}{2}}$$

Y^2_i values are the square of the drift-corrected amplitude and np is the number of points in the noise region.

Arguments low_field and high_field are the upper and lower frequencies of the noise region to be measured. The default is the position of the two cursors.

signal_to_noise is the calculated value of signal-to-noise ratio.

noise is the noise value measured within the defined spectral region.

Examples
```
dsn:$ston
dsn(sp+sp,sp+wp-100)
dsn(10000,8000):r1
```

See also *User Programming*

Related

dres	Measure linewidth and digital resolution (C)
peak	Find tallest peak in specified region (C)
sp	Start of plot (P)
vs	Vertical scale (P)
wp	Width of plot (P)

dsnmax Calculate maximum signal-to-noise (M)

Syntax dsnmax<(noise_region)>

Description Finds the best signal-to-noise in a specified region.

Arguments noise_region is the size, in Hz, of the region. The default is the region between the cursors as defined by the parameter delta.

Examples
```
dsnmax
dsnmax(400)
```

See also *User Programming*

Related

delta	Cursor difference in directly detected dimension (P)

dsp Display calculated spectrum (C)

Syntax dsp<(file<,'nods'>)>

Description Using the current table of transitions and intensities, dsp recalculates the simulated spectrum (using the current value for the linewidth slw) and displays the spectrum. dsp can only be used after the spins program has been run. If only the linewidth slw or vertical scale svs have been changed, dsp can be used to redisplay the spectrum. If a chemical shift or coupling constant has been changed, however, `dsp` will not display a spectrum reflecting the changes in the parameter; `spins` must be run again to recalculate the new spectrum.

The number of points in the calculated spectrum is `fn/2`. To increase the number of points, change `fn` and rerun `dsp` without doing a transform.

To display a synthetic spectrum, prepare a file in the following format:

```
Freq1, Intens1, LineWidth1, GaussFrac1
Freq2, Intens2, LineWidth2, GaussFrac2
...
FreqN, IntensN, LineWidthN, GaussFracN
```

The units for frequency and line width are Hz. The Gaussian fraction, which is the percentage of the line shape that is Gaussian (the rest is Lorentzian) should be between 0 and 1 (i.e., 0 is pure Lorentzian, 1 is pure Gaussian). Units for intensity are not particularly important. Given numbers in a file `myshape`, it is only necessary to enter `dsp('myshape')` to display the synthetic spectrum. This approach is often preferred over deconvolution for quantifying small shoulders on large peaks.

Arguments `file` is the name of a file containing spectral information that displays the result of a spectrum deconvolution. Any file in the proper format can be used to generate a display. The default is the file `spins.outdata` in the experiment directory. This file contains information about frequencies, intensities, line widths, and Gaussian/Lorentzian fractions.

`'nods'` is a keyword for `dsp` to recalculate the simulated spectrum but not to display the spectrum. The spectrum can be displayed with the `ds` or `dss` command.

Examples

```
dsp
dsp('fitspec.outpar')
```

See also *NMR Spectroscopy User Guide*

Related

`ds`	Display a spectrum (C)
`dss`	Display stacked spectra (C)
`fn`	Fourier number in directly detected dimension (P)
`slw`	Spin simulation linewidth (P)
`spins`	Perform spin simulation calculation (C)
`svs`	Spin simulation vertical scale (P)

dsp — Type of DSP for data acquisition (P)

Description Selects the type of DSP (digital signal processing) for data acquisition:

- *Inline DSP* performs digital filtering and downsampling on the workstation immediately after each oversampled FID is transferred from the console. sw and at should be set to the values desired for the final spectrum. Only the digital filtered and downsampled data is written to the disk. Selective detection of a region of a spectrum is available using the moveossw macro.
- *Real-time DSP* uses optional hardware (not available on all systems) to filter the data prior to summing to memory. Real-time DSP is not compatible with pulse sequences that use explicit acquisition to acquire less than the full number of data points (np) in a single acquire statement (e.g., solids sequences such as BR24 and FLIPFLOP).

If either type is active, the filter bandwidth parameter fb is not active. The actual analog filter *is* active and is automatically set by the software to a value that matches (sw/2)*oversamp as closely as possible.

Another type of DSP is available that allows post-processing of data. See the description of the pards macro for details.

Values 'i' selects inline DSP and calls addpar('oversamp') to create the DSP parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp. A value of oversamp greater than 1 causes the next experiment run to be oversampled, digitally filtered, and downsampled back to the selected sw prior to saving it to disk.

'r' selects real-time DSP and calls the macro addpar('oversamp') to create the DSP parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp (although only oversamp and osfilt are user adjustable for real-time DSP). Use dsp='r' only if the optional DSP hardware is present in the system. Set fsq='y' to use frequency-shifted quadrature detection.

'n' (or parameter dsp is not present) disables both types of DSP. Set dsp='n' if you wish to turn off DSP on a permanent or semi-permanent basis. To turn off DSP within just a single experiment, set oversamp='n'.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to current experiment (M)
at	Acquisition time (P)
def_osfilt	Default value of osfilt (P)
fb	Filter bandwidth (P)
filtfile	File of FIR digital filter coefficients (P)
fsq	Frequency-shifted quadrature detection (P)
il	Interleave arrayed and 2D experiments (P)
moveossw	Set oversampling parameters for selected spectral region (M)
np	Number of data points (P)
oscoef	Digital filter coefficients for oversampling (P)
osfb	Digital filter bandwidth for oversampling (P)
osfilt	Oversampling filter for real-time DSP (P)
oslsfrq	Bandpass filter offset for oversampling (P)

oversamp	Oversampling factor for acquisition (P)
pards	Create additional parameters used by downsampling (M)
paros	Create additional parameters used by oversampling (M)
ra	Resume acquisition stopped with *sa* command (C)
sa	Stop acquisition (C)
sw	Spectral width in the directly detected dimension (P)

dsplanes Display a series of 3D planes (M)

Syntax `dsplanes(start_plane,stop_plane)`

Description Produces a graphical 2D color or contour map for a subset of 3D planes. The dconi program is used to display the planes.

Arguments `start_plane` specifies the number of the 3D plane with which display is to begin. It must be greater than 0.

`stop_plane` specifies the number of the 3D plane with which the display is to end. If `start_plane` is greater than `stop_plane`, only the first plane, whose number is `start_plane`, is plotted. The range of `stop_plane` depends on the value of the parameter plane as follows:

- If plane='f1f3', range of stop_plane is between 0 and fn2/2
- If plane='f2f3', range of stop_plane is between 0 and fn1/2
- If plane='f1f2', range of stop_plane is between 0 and fn/2

Examples `dsplanes(1,3)`

See also *NMR Spectroscopy User Guide*

Related

dconi	Interactive 2D data display (C)
dplane	Display a 3D plane (M)
dproj	Display a 3D plane projection (M)
getplane	Extract planes from 3D spectral data set (M)
nextpl	Display the next 3D plane (M)
plane	Currently displayed 3D plane type (P)
plplanes	Plot a series of 3D planes (M)
prevpl	Display the previous 3D plane (M)

dsptype Type of DSP (P)

Description Indicates the existence of digital signal processing (DSP).

Values 0 indicates no digital signal processing. 1 indicates DSP exists.

Examples `dsptype?=0 dsptype?=1`

See also *NMR Spectroscopy User Guide*

Related

dsp	Type of DSP for data acquisition (P)

dss Display stacked spectra (C)

Syntax dss<(<start,finish<,step>><,options>)>

Description Displays one or more spectra on the screen.

The display is not interactive like the command ds. Integral display is controlled by the parameter intmod when a single spectrum is displayed (see 'int' option below). The following values are accepted for intmod:

- intmod='off' turns off the integral display.
- intmod='full' displays the entire integral.
- intmod='partial' displays every other integral region.

An individual trace is displayed from and arrayed 1D spectra or 2D spectra by supplying the index number as an argument. Spectra from 2D data set are displayed from either the f_1 or f_2 domain by setting the parameter trace equal to 'f1' or 'f2', respectively. Enter ft1d, trace='f1', and dss to view the interferogram. Multiple spectra are displayed by supplying indexes of the first and last spectra.

The position of the first spectrum is governed by the parameters wc, sc, and vp. For 1D data, subsequent spectra are positioned relative to the preceding spectrum by the parameters vo (vertical offset) and ho (horizontal offset). For 2D data, ho defines the total horizontal offset between the first and last spectrum. Also for 2D data, vo is inactive while the parameter wc2 defines the total vertical offset between the first and last spectrum.

The parameter cutoff, if it exists and is active, defines the distance above and below the current vertical position vp at which peaks are truncated. By arraying cutoff to have two different values, the truncation limits above and below the current vertical position can be controlled independently. For example, cutoff=50 truncates peaks at vp+50 mm and vp–50 mm. cutoff=50,10 truncates peaks at vp+50 mm and vp–10 mm.

Arguments start is the index of the first spectra when displaying multiple spectra. It is also the index number of a particular trace to be viewed when displaying arrayed 1D spectra or 2D spectra.

finish is the index of the last spectra when displaying multiple spectra. Since the parameter arraydim is automatically set to the total number of spectra, it can be used to set finish to include all spectra (e.g., dss(1,arraydim,3)).

step is the increment for the spectral index when displaying multiple spectra. The default is 1.

options can be any of the following:

- 'all' is a keyword to display all of the spectra.
- 'int' is a keyword to display only the integral, independently of the value of the parameter intmod

- `'top'` or `'side'` are keywords that cause the spectrum to be displayed either above or at the left edge, respectively, of a contour plot. This assumes that the parameters `sc`, `wc`, `sc2`, and `wc2` are those used to position the contour plot.
- `'dodc'` is a keyword for all spectra to be drift corrected independently.
- `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, and `'white'` are keywords that select a color.
- `'pen1'`, `'pen2'`, `'pen2'` ... are keywords that pens.
- `'nopars'` — prevents the display commands from drawing the parameters at the bottom of the graphics screen.
- `'custom'` — uses the parameters `shownumx` (x position) and `shownumy` (y position), counting from bottom left of every spectrum.
- `'reverse'` — rotate the text by 90° - useful if the arrayed parameter values are long with respect to the width of the individual sub-spectra.
- `'value'` —The values of up to two simultaneous arrays are displayed. Diagonal arrays are allowed. The second parameter is shown in different color). The name of the arrayed parameter(s) is also shown. If used on a one-dimensional array representation of a 2D spectrum, `ni` and `phase` (in case of phase sensitive 2Ds) parameters are shown.

Examples
```
dss(1,3)
dss(1,12,3,'green')
```

See also *NMR Spectroscopy User Guide*

Related

`cutoff`	Data truncation limit (P)
`dssa`	Display stacked spectra automatically (C
`dssan`	Display stacked spectra automatically without erasing (C)
`dssh`	Display stacked spectra horizontally (C)
`dsshn`	Display stacked spectra horizontally without erasing (C)
`dssn`	Display stacked spectra without screen erase (C)
`dsww`	Display spectra in whitewash mode (C)
`ft1d`	Fourier transform along f_2 dimension (C)
`ho`	Horizontal offset (P)
`intmod`	Integral display mode (P)
`pl`	Plot spectra (C)
`plww`	Plot spectra in whitewash mode (C)
`sc`	Start of chart (P)
`sc2`	Start of chart in second direction (P)
`shownumx`	x position counting from bottom left of every spectrum (P)
`shownumy`	y position counting from bottom left of every spectrum (P)
`trace`	Mode for 2D data display (P)
`vo`	Vertical offset (P)
`vp`	Vertical position of spectrum (P)

wc	Width of chart (P)
wc2	Width of chart in second direction (P)

dssa Display stacked spectra automatically (C)

Syntax dssa<(<start,finish<,step>><,options>)>

Description Displays one or more spectra automatically.

Integral display is controlled by the parameter intmod when a single spectrum is displayed (see 'int' option below). The following values are accepted for intmod:

- intmod='off' turns off the integral display.
- intmod='full' displays the entire integral.
- intmod='partial' displays every other integral region.

An individual trace is displayed from and arrayed 1D spectra or 2D spectra by supplying the index number as an argument. Spectra from 2D data set are displayed from either the f_1 or f_2 domain by setting the parameter trace equal to 'f1' or 'f2', respectively. Enter ft1d, trace='f1', and dss to view the interferogram. Multiple spectra are displayed by supplying indexes of the first and last spectra.

The position of the first spectrum is governed by the parameters wc, sc, and vp. For 1D data, subsequent spectra are positioned relative to the preceding spectrum by the parameters vo (vertical offset) and ho (horizontal offset). For 2D data, ho defines the total horizontal offset between the first and last spectrum.

Also for 2D data, vo is inactive while the parameter wc2 defines the total vertical offset between the first and last spectrum. To display spectra “automatically,” the command dssa adjusts the parameters vo and ho to fill the screen in a lower left to upper right presentation (wc must be set to less than full screen width for this to work)

The parameter cutoff, if it exists and is active, defines the distance above and below the current vertical position vp at which peaks are truncated. By arraying cutoff to have two different values, the truncation limits above and below the current vertical position can be controlled independently. For example, cutoff=50 truncates peaks at vp+50 mm and vp–50 mm. cutoff=50,10 truncates peaks at vp+50 mm and vp–10 mm.

Arguments start is the index of the first spectra when displaying multiple spectra. It is also the index number of a particular trace to be viewed when displaying arrayed 1D spectra or 2D spectra.

finish is the index of the last spectra when displaying multiple spectra.

step is the increment for the spectral index when displaying multiple spectra. The default is 1.

options can be any of the following:

- 'all' is a keyword to display all of the spectra.

- 'int' is a keyword to only display the integral, independently of the value of the parameter intmod
- 'dodc' is a keyword for all spectra to be drift corrected independently.
- 'red', 'green', 'blue', 'cyan', 'magenta', 'yellow', 'black', and 'white' are keywords that select a color.
- 'pen1', 'pen2', 'pen2' ... are keywords that pens.
- 'nopars' — prevents the display commands from drawing the parameters at the bottom of the graphics screen.

Examples `dssa(1,3)`

See also *NMR Spectroscopy User Guide*

Related

cutoff	Data truncation limit (P)
dss	Display stacked spectra (C)
dssan	Display stacked spectra automatically without erasing (C)
dssh	Display stacked spectra horizontally (C)
dsshn	Display stacked spectra horizontally without erasing (C)
dssn	Display stacked spectra without screen erase (C)
dsww	Display spectra in whitewash mode (C)
ft1d	Fourier transform along f_2 dimension (C)
ho	Horizontal offset (P)
intmod	Integral display mode (P)
pl	Plot spectra (C)
plww	Plot spectra in whitewash mode (C)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
shownumx	x position counting from bottom left of every spectrum (P)
shownumy	y position counting from bottom left of every spectrum (P)
trace	Mode for 2D data display (P)
vo	Vertical offset (P)
vp	Vertical position of spectrum (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

dssan Display stacked spectra automatically without erasing (C)

Syntax `dssan<(<start,finish<,step>><,options>)>`

Description Functions the same as the command dssa except the graphics window is not erased before starting the display. This allows composite displays of many spectra to be created. The arguments are the same as dssa.

Examples `dssan(1,3)`

See also *NMR Spectroscopy User Guide*

Related dssa Display stacked spectra automatically (C)

dssh Display stacked spectra horizontally (C)

Syntax `dssh<(<start,finish<,step>><,options>)>`

Description Displays one or more spectra horizontally.

Integral display is controlled by the parameter intmod when a single spectrum is displayed (see 'int' option below). The following values are accepted for intmod:

- intmod='off' turns off the integral display.
- intmod='full' displays the entire integral.
- intmod='partial' displays every other integral region.

An individual trace is displayed from and arrayed 1D spectra or 2D spectra by supplying the index number as an argument. Spectra from 2D data set are displayed from either the f_1 or f_2 domain by setting the parameter trace equal to 'f1' or 'f2', respectively. Enter ft1d, trace='f1', and dss to view the interferogram. Multiple spectra are displayed by supplying indexes of the first and last spectra.

The position of the first spectrum is governed by the parameters wc, sc, and vp. For 1D data, subsequent spectra are positioned relative to the preceding spectrum by the parameters vo (vertical offset) and ho (horizontal offset). For 2D data, ho defines the total horizontal offset between the first and last spectrum. Also for 2D data, vo is inactive while the parameter wc2 defines the total vertical offset between the first and last spectrum. To display spectra horizontally, the command dssh causes vo to be set to zero and for ho, sc, and wc to be adjusted to fill the screen from left to right with the entire array.

The parameter cutoff, if it exists and is active, defines the distance above and below the current vertical position vp at which peaks are truncated. By arraying cutoff to have two different values, the truncation limits above and below the current vertical position may be controlled independently. For example, cutoff=50 truncates peaks at vp+50 mm and vp–50 mm, and cutoff=50,10 truncates peaks at vp+50 mm and vp–10 mm.

Arguments start is the index of the first spectra when displaying multiple spectra. It is also the index number of a particular trace to be viewed when displaying arrayed 1D spectra or 2D spectra.

finish is the index of the last spectra when displaying multiple spectra.

step is the increment for the spectral index when displaying multiple spectra. The default is 1.

options can be any of the following:

- 'all' is a keyword to display all of the spectra.

- `'int'` is a keyword to only display the integral, independently of the value of the parameter `intmod`
- `'dodc'` is a keyword that causes all spectra to be drift corrected independently.
- `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, and `'white'` are keywords that select a color.
- `'pen1'`, `'pen2'`, `'pen2'` ... are keywords that pens.
- `'nopars'` — prevents the display commands from drawing the parameters at the bottom of the graphics screen.

Examples `dssh(1,3)`

See also *NMR Spectroscopy User Guide*

Related

`cutoff`	Data truncation limit (P)
`dss`	Display stacked spectra (C)
`dssa`	Display stacked spectra automatically (C)
`dssan`	Display stacked spectra automatically without erasing (C)
`dsshn`	Display stacked spectra horizontally without erasing (C)
`dssn`	Display stacked spectra without screen erase (C)
`dsww`	Display spectra in whitewash mode (C)
`ft1d`	Fourier transform along f_2 dimension (C)
`ho`	Horizontal offset (P)
`intmod`	Integral display mode (P)
`pl`	Plot spectra (C)
`plww`	Plot spectra in whitewash mode (C)
`sc`	Start of chart (P)
`sc2`	Start of chart in second direction (P)
`shownumx`	x position counting from bottom left of every spectrum (P)
`shownumy`	y position counting from bottom left of every spectrum (P)
`trace`	Mode for 2D data display (P)
`vo`	Vertical offset (P)
`vp`	Vertical position of spectrum (P)
`wc`	Width of chart (P)
`wc2`	Width of chart in second direction (P)

dsshn Display stacked spectra horizontally without erasing (C)

Syntax `dsshn<(<start,finish<,step>><,options>)>`

Description Functions the same as the command `dssh` except the graphics window is not erased before starting the display. This allows composite displays of many spectra to be created. The arguments are the same as `dssh`.

Examples `dssh(1,3)`

See also *NMR Spectroscopy User Guide*

Related	dssh	Display stacked spectra horizontally (C)

dssl Label a display of stacked spectra (M)

Syntax `dssl(<options>)`

Description Displays a label for each element in a set of stacked spectra. The label is an integer value from 1 up to the number of spectra in the display or the values of parameters up to 2 dimensions.

Labels can appear at incorrect positions if `wysiwyg='n'`. The positions are empirically determined for a large screen display and are not guaranteed to be correct for all displays.

Arguments `options` control the display (more than one option can be entered as long as the options do not conflict with each other):

- `'center'`, `'left'`, `'right'`, `'top'`, `'bottom'`, `'above'`, and `'below'` are keywords setting the position of the displayed index relative to each spectrum.
- `'custom'` — uses the parameters `shownumx` (x position) and `shownumy` (y position), counting from bottom left of every spectrum.
- `'list=xxx'` produces a display of the values contained in the arrayed parameter `xxx`.
- `'format=yyy'` uses the format `yyy` to control the display of each label. See the `write` command for information about formats.
- `'reverse'` — rotate the text by 90° - useful if the arrayed parameter values are long with respect to the width of the individual sub-spectra.
- `'value'` —The values of up to two simultaneous arrays are displayed. Diagonal arrays are allowed. The second parameter is shown in different color). The name of the arrayed parameter(s) is also shown. If used on a one-dimensional array representation of a 2D spectrum, `ni` and `phase` (in case of phase sensitive 2Ds) parameters are shown.

Examples
```
dssl
dssl('top','left')
dssl('value','format=%3.1f') pssl
```

See also *NMR Spectroscopy User Guide*

Related	dss	Display stacked spectra (C)
	shownumx	x position counting from bottom left of every spectrum (P)
	shownumy	y position counting from bottom left of every spectrum (P)
	write	Write formatted text to a device (C)

dssn Display stacked spectra without screen erase (C)

Syntax `dssn<(<start,finish<,step>><,options>)>`

Description Functions the same as the command dss except the graphics window is not erased before starting the display. This allows composite displays of many spectra to be created. The arguments are the same as dss.

Examples `dssn(1,3)`

See also *NMR Spectroscopy User Guide*

Related dss Display stacked spectra (C)

dsvast Display VAST Data in a stacked 1D-NMR matrix format

Applicability VnmrJ 3.1

Description If an array of 1D spectra have been acquired (in particular if a block of 96 spectra have been acquired using VAST automation, especially in a microtiter-plate format), and if these spectra have been glued into a reconstructed 2D dataset (see vastglue), this macro will arrange and display them (on the screen) in a convenient 8 x 12 sample format (as a matrix of 1D spectra).
Uses a file (template) created by plate_glue to display a matrix of data. The number of spectra displayed, and their order, are controlled by the template file. Each "little spectrum" is labeled with its respective alphanumeric coordinates. The modulo number controls how many spectra appear per row.

Examples `dsvast(<display order>, <modulo>)`

See also dsvast
dsvast2d
plvast
plvast2d
intvast
pintvast
plateglue
vastglue
vastget

dsvast2d Display VAST Data in a pseudo-2D format

Applicability VnmrJ 3.1

Description If an array of 1D spectra have been acquired (in particular if a block of 96 spectra has been acquired using VAST automation, especially in a microtiter-plate format), and if these spectra have been glued into a reconstructed 2D dataset (see vastglue), this macro will arrange and display them (on the screen) in a convenient pseudo-2D format (almost like an LC-NMR chromatogram).

The default is to plot all the spectra (from 1 through arraydim). An optional argument (plvast(##)) allows one to specify that only spectra from 1 through ## should be plotted.

See also dsvast
dsvast2d
plvast
plvast2d
pintvast

dsww Display spectra in whitewash mode (C)

Syntax `dsww<(<start,finish<,step>><,'int'>)>`

Description Displays one or more spectra in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind a prior spectra).

Arguments `start` is the index of the first spectra when displaying multiple spectra. It is also the index number of a particular trace to be viewed when displaying arrayed 1D spectra or 2D spectra; default is to display all spectra.

`finish` is the index of the last spectra when displaying multiple spectra.

`step` is the increment for the spectral index when displaying multiple spectra. The default is 1.

`'int'` is a keyword to display only the integral, independently of the value of the parameter intmod

Examples `dsww(1,3)`

Related

dss	Display stacked spectra (C)
dssa	Display stacked spectra automatically (C)
dssan	Display stacked spectra automatically without erasing (C)
dssh	Display stacked spectra horizontally (C)
dsshn	Display stacked spectra horizontally without erasing (C)
dssn	Display stacked spectra without screen erase (C)
pl	Plot spectra (C)
plww	Plot spectra in whitewash mode (C)

dtext Display a text file in graphics window (M)

Syntax `dtext<(file,x,y)><:$x_next,$y_next,$increment>`

Description Displays a text file in the graphics window.

Arguments `file` is the name of a text file. The default is the current experiment text file.

`x` and `y` are coordinates of the first line of text. This positions the location of the output. The default is the upper left-hand corner of the screen.

`$x_next` and `$y_next` are the coordinates where the start of the next line would have been displayed. This is useful for subsequent character display.

`$increment` is the increment between lines.

Examples
```
dtext
dtext(userdir+'/exp3/text')
dtext(100,100)
dtext:$x,$y,$dy
```

Related

pltext	Plot a text file (M)
ptext	Print out a text file (M)
text	Display text or set new text for current experiment (C)
write	Write formatted text to a device (C)

dtrig — Delay to wait for another trigger or acquire a spectrum (P)

Applicability: Systems with LC-NMR accessory.

Description: If `ntrig` is greater than 0 after a trigger is detected, a pulse sequence waits for `dtrig` seconds before either waiting for another trigger or acquiring a spectrum. Typically, after the LC has positioned the sample in the NMR probe and stopped the pump, there is a small time (30 seconds) during which conditions (pressure, etc.) in the NMR probe are still settling; better NMR performance is obtained if an appropriate delay is inserted using `dtrig`. If `dtrig` does not exist, a value of 0 is assumed. If `dtrig` does not exist, the `parlc` macro can create it.

Related

ntrig	Number of trigger signals to wait before acquisition (P)
parlc	Create LC-NMR parameters (M)

dutyc — Duty cycle for homodecoupling (optional) (P)

Applicability: VNMRS systems, 400 MR

Syntax: `dutyc=<value>`

Description: Sets the rf duty cycle fraction (0.0-0.4) for rf on part of homonuclear decoupling. The duty cycle default is 0.1 (or 10% rf on) if the `dutyc` does not exist. Homonuclear decoupling delay before and after the rf on period. `homorof1`, `homorof2`, and `homorof3`, are equivalent to `rof1`, `rof2` and `rof3` and all default to 2 µsec.

Values: 0.0 to 0.4 — default is 0.1

Examples: `dutyc=0.2` sets a 20% duty cycle

Related	homo	Homodecoupling control for observe channel (P)
	hdof	Frequency offset for homodecoupling (P)
	hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
	hdmf	modulation frequency for the band selective homonuclear decoupling (P)
	hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
	hdres	Sets the tip angle resolution (P)
	hdseq	Sets the decoupler waveform filename (P)
	homorof1	Delay before turning on homo decoupling rf (P)
	homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
	homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
	tn	Nucleus for observe transmitter (P)

E

e	Eject sample (M)
eaddr	Display Ethernet address (M,U)
ecc_on	Turns on eddy current compensation for Cold Probes (M)
ecc_off	Turns off eddy current compensation for Cold Probes (M)
echo	Simple echo command similar to unix echo
edit	Edit a file with user-selectable editor (M)
editht	Create and edit a Hadamard frequency list
editLog	Customize the log details.
editparlib	This macro has been superseded by the Clone utilities. (M)
eject	Eject sample (M)
elist	Display directory on remote VXR-style system (M,U)
email	Email address (P)
enter	Enter sample information for automation run (M,U)
enterdialog	Start a dialog window using enterexp file (M)
epage	Emails Output
eplot	Emails PostScript
eread	Transfer file from remote source (M,U)
ernst	Calculate the Ernst angle pulse (C)
errlog	Display recent error messages (C)
errloglen	Number of lines in error message display (P)
ewrite	Transfer file to remote destination (M,U)
exec	Execute a command (C)
execpars	Set up the exec parameters (M)
execplot	Execute plotting macro (P)
execprep	Execute prepare macro (P)

execprescan	Execute prescan macro (P)
execproc	Execute processing macro (P)
execprocess	Execute processing macro (P)
execsetup	Execute setup macro (P)
exists	Checks if parameter, file, or macro exists and file type (C)
exit	Call the vnmrexit command (M)
exp	Find exponential value of a number (C)
expl	Display data on the screen
expladd	Add another diffusion analysis to current display (M)
explib	Display experiment library (M)
explist	Display current experiment chain and approx. time for each (M)

e — Eject sample (macro)

Syntax `e`

Applicability VnmrJ 3.1

Description Turns on the eject and slow drop air to eject the sample from the probe.

Arguments This command is valid on Mercury and GEMINI 2000 only if the optional spin control hardware is installed.

eaddr — Display Ethernet address (M,U)

Description Displays the name of the local host and its hardware Ethernet address. The 48-bit address is presented in octal, decimal, and hexadecimal formats.

See also *NMR Spectroscopy User Guide*

Related dnode Display list of valid limNET nodes (M,U)

ecc_on — Turns on eddy current compensation for Cold Probes (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Turns on eddy current compensation

Related `ecc_off` Turns off eddy current compensation for Cold Probes (M)

`ecc_off` Turns off eddy current compensation for Cold Probes (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Turns off eddy current compensation.

Related `ecc_on` Turns on eddy current compensation for Cold Probes (M)

`echo` Simple echo command similar to unix echo

Syntax `echo[([-n,]arg1, arg2, ....)]`

Applicability VnmrJ 3.1

Description This command will display strings and variable values on the output window. The `echo` command automatically advances to the next line after displaying (it sends a newline character).

Arguments args can be strings surrounded by single quotes and variables. The -n option prevents the `echo` command from sending a newline character.

Examples `echo`: Advance to next line (send newline)
`echo('hello')`: Display string.
`echo('variable a=',a)`: Display string and variable
`echo('-n','Please enter a number:')`: Display string without a newline.

`edit` Edit file or a macro with user-selectable editor

Syntax `edit('myfile') - edit a file with user-selectable editor`
`macroedit('mycmd') - edit a macro with user-selectable editor`

Applicability VnmrJ 3.1

Description The `edit` command will edit a file, letting you select the editor program to be used. Set the environmental parameter "`vnmreditor`" to be the desired editor program. The default is "`vi`".

You must provide a `vnmr_<editor>` script in the bin subdirectory of the VNMR system directory. For example, if "emacs" is to be used, a script named "`vnmr_emacs`" would need to be present. The major task for this script is determining if a GUI is in use and making required adjustments. The scripts "vnmr_vi" and "vnmr_textedit" provide a mode for non-window and window-based editor interface respectively.

The command macroedit will edit a Magical macro in your personal macro library. System macros cannot be directly edited with this command; they must first be copied to your personal library first.

editht Create and edit a Hadamard frequency list.

Syntax

Applicability VnmrJ 3.1

Description The editht macro opens the Edit HT Freq dialog, for interactively creating and editing a Hadamard frequency line list.

To set up a Hadamard experiment starting from a 1D experiment, do the following:

1. First run a Proton, Carbon, or other 1D experiment, depending on the type of Hadamard experiment you wish to run (homonuclear or heteronuclear).

2. When the acquisition is finished, process and phase the spectrum.

3. Run the editht macro to open the Edit HT Freq dialog. Create a Hadamard frequency list for the nucleus of interest. Save the frequency list.

4. For a heteronuclear Hadamard experiment, run a Proton experiment, and adjust spectral width and decoupling as desired.

5. Load the desired Hadamard experiment. Check the Hadamard frequency list and other parameters.

6. Start the acquisition of the Hadamard experiment.

7. When acquisition is complete, process with proc1='ht' wft2da.

How to use the Edit HT Freq dialog.

To make a Hadamard frequency list from a 1D spectrum (step 3 above), use the buttons in the Edit HT Freq dialog.

Create Line List: Processes the current spectrum as follows:

- Fourier transform with wft. Multiplet structures can be smoothed out using line broadening.
- Create a line list using nll, greater than the current threshhold.
- Keep only frequencies that are the minimum line width apart in the "Min line width" entry box (e.g. 20 Hz).

Nearest line: Place the cursor on the nearest line.

Select: Adds the current cursor position to the line list. (The cursor must be more than the minimum line width from an existing frequency in the line list.)

Remove: Removes the line nearest the cursor position from the line list.

Display: Display the frequency list. If a 1D spectrum is displayed, show the frequencies using dpf in units set by the axis parameter.

CLEAR: Clear all frequencies from the frequency list.

Save HT Frequencies: Saves the current frequency list as a Hadamard line list for the current nucleus (tn). It saves the frequency list, band width, current nucleus, spectral width, and frequency offset in a persistence file. The frequencies and other parameters are loaded from the persistence file when loading a Hadamard experiment (step 5 above).

Line List: The line list is displayed in the text entry window on the right hand side of the page. You may edit the line list directly from this window. Click the 'Set list into parameters' button to set the line list changes into the parameters. The first column of numbers is the Hadamard frequency list, e.g. htfrq1. If there is a second column of numbers, it specifies the bandwidth for each frequency in Hz.

Hz/ppm menu: Select Hz or ppm to display the line list in Hz or ppm. If Hz is selected, the line list is displayed in Hz from the center of the spectrum.

Move HT pars to exp: Move the Hadamard parameters from the current workspace to a new workspace. The workspace number is specified in the entry box.

Set list into parameters: Sets the changes from the line list text entry window into the parameters.

Import list curexp / htfrq1.ll: Copies a line list file from curexp into the current line list, and sets the line list into the parameters. The line list file to be copied is named after the frequency parameter, e.g.

```
/export/home/vnmr1/vnmrsys/exp2/htfrq1.ll
```

The format of the file is the same as the line list display.

Arguments

htfrq1 - Hadamard frequency list in indirect dimension, in Hz from center of spectrum, or ppm.

htbw1 - Hadamard band width in indirect dimension, in Hz. It may be a single value or a list of values for each element in the htfrq1 list.

tn - nucleus used for frequency list.

Examples

Example #1:

```
freq [Hz from center]
```

1172.37

327.69

-346.37

-1292.10

In Example #1, the Hadamard frequencies are in Hz from the center of the spectrum.

Example #2:

freq [ppm]	bw [Hz]
7.930	20
5.819	16
4.134	20

1.770 20

In Example #2, the Hadamard frequencies are in ppm, referenced to the current spectrum. The bandwidth for each frequency is also specified as 16 Hz for the second frequency, and 20 Hz for the rest. If `htbw1` is arrayed to two or more values in the parameter set, the values are written to the line list file. If the size of the `htbw1` array is smaller than the size of the htfrq1 array, the last value of `htbw1` is applied to the remaining frequencies.

In a 2D display, the Edit HT Freq dialog may be used to view the Hadamard frequency list in F1. Interactive frequency selection and display from the graphics window may be done. You may also edit frequencies from the Line List window. In a 2D display, frequencies in ppm are referenced to F1.

See also
`ht`
`HsqcHT`
`tocsyHT`
`getht`
`mht`
`sethtfrq1`
`htfrqdisp`
`dll`

editLog — Customize the log details

Applicability: VnmrJ 3.1, VnmrJ 3.2

Description: The `sqLog` macro records specific events from a study queue. The messages and details of the logging are customizable with the `editLog` utility.
The `sqLog` macro is very generic. It gets all of its details from a file written by the `editLog` utility. This file has the same name as the macro and is in the <appdir>/templates/vnmrj/loginfo directory.
`sqLog` saves logging information only for automation runs. The log editor can handle menus of choices. Files in templates/vnmrj/loginfo with the same name as the keyword will be used to make menus of choices to select from within the `editLog` editor.
Files prefixed with the name of the logging macro, for example `sqLog` will make a File menu specific for editLog('sqLog'). The logging macro, along with the editLog editor are very general and can be used to log other events. As an example, suppose one wants to monitor access to the VnmrJ program.

Description: The sqLog facility will record the following events: `SampleStart, SampleEnd, ExpStart, ExpEnd, ExpError`. Each event recorded in the logfile may may be preceded by header information. This may include things like the date, time, user, etc. This header information is customizable.

Examples: The sqLog macro is very generic. It gets all of its details from a file written be the `editLog` utility. This file has the same name as the macro and is in **<appdir>/templates/vnmrj/loginfo** directory. For example, the current sqLog file is:

```
# Formatting statements for automation log files.
#
1Header Date: %MOC% %DAY% %YR% at %HR%:%MIN%:%SEC%,
User: $operator$, Sample: $samplename$,
1SampleStart Start new sample at location $loc$.
1SampleEnd Finish sample at location $loc$\\####
1ExpStart Experiment $pslabel$ started.
1ExpEnd Experiment $pslabel$ complete.
1ExpError Experiment error: $$2$
1ExpPrescan Prescan:
1File $autodir$/logfile
1Ifcondition (auto='y')
```

Lines starting with a hash mark (#) are comments. The first character of each non-comment line is a `1` or `0`, indicating enabled or disabled. The rest of the first word, following the `1` or `0`, is a keyword that is passed to the `sqLog` macro. The remainder of a line is the template for writing the log file. The template is passed to the `chkname` command for translation.

The `File` keyword defines where the log file will be saved. If this keyword is disabled, all of the `sqLog` event logging will be disabled. Disabling other keywords only disables that specific event or feature.

The `Ifcondition` keyword allows the logging mechanism to make decisions as to whether to log the event. For example, in the case of `sqLog`, we only log events during an automation run. Logging will occur only if the `Ifcondition` is true.
A special keyword of "`None`" for the `Ifcondition` specifies no special conditions. That is, events are always logged.

The `sqLog` macro is called from appropriate places in the software. It is called with the keyword as the first argument. If the template uses passed arguments, they can be passed to the `sqLog` macro. For example, the `ExpError` template includes the second argument in its templates, which contains the actual error. This would be called as:

```
geterror:$err
  sqLog('ExpError',$err)
```

During an automation run, messages written to 'line3', which puts them into the "`acqlog`". If `sqLog` is called with no arguments but one return value, the pathname of the log file, defined by the `File` keyword, is returned.

If `sqLog` is passed an event keyword, with optional additional arguments, and requests a return value, the message will not be written into the log file (nor on line3 for automation runs), but will be returned to the calling macro. An example would be:

```
sqLog('SampleStart'):$res
```

As defined above, sqLog saves logging information only for automation runs. By changing the File attribute to your **userdir** directory, and setting the Ifcondition to None, all study queue activities will be logged, both automation and foreground.

The log editor can handle menus of choices. Files in **templates/vnmrj/loginfo** with the same name as the keyword will be used to make menus of choices to select from within the editLog editor. Files prefixed with the name of the logging macro, for example sqLog will make a File menu specific for editLog('sqLog')

The logging macro, along with the editLog editor are very general and can be used to log other events. As an example, suppose one wants to monitor access to the VnmrJ program. A "loginLog" could be made as follows.
Make a copy of the sqLog macro called loginLog.
Add a loginLog file describing the events to logged to the **<appdir>/templates/vnmrj/loginfo**. An example of such a file may be:

```
# Formatting statements for login log files.
#
1Header Date: %MOC% %DAY% %YR% at %HR%:%MIN%:%SEC%,
User: $operator$
1Login Login
1Logout Logout
1File $systemdir$/acqqueue/loginLog
1Ifcondition ((auto='n') and (jviewport=1))
```

The only remaining task is to place calls to the loginLog macro in various other macros. In this case, one might call loginLog('Login'):$res fromthe bootup macro and loginLog('Logout'):$res
from the exit macro. If one wanted to monitor "operator" logins, one could and additional keywords such as Operatorlogin and Operatorlogout to the above file and then call loginLog('Operatorlogin'):$res from the operatorlogin macro and call loginLog('Operatorlogout'):$res from the operatorlogout macro.

The following are more examples.

sqLog(event<,args>) - log automation events

sqLog(event<,args>):$res - return automation events to calling macro

sqLog:$path - return log file path

editLog - Customize the log details.

See also sqLog

eject — Eject sample (M)

Syntax eject

Description Ejects the sample from the probe by turning on the eject air and the slow drop air. The e macro functions the same as the e macro.

See also *NMR Spectroscopy User Guide*

Related

e	Eject sample (M)
i	Insert sample (M)
insert	Insert sample (M)

elist — Display directory on remote VXR-style system (M,U)

Syntax elist(remote_node,remote_directory)
(From UNIX) elist remote_node remote_directory

Description Lists directory contents on a remote VXR-style (Gemini, VXR-4000, or XL) system.

Arguments remote_node is the name of the remote VXR-style system.

remote_directory is the name of the directory on the remote system.

Examples elist('gemini','fidlib')
(From UNIX) elist gemini fidlib

See also *NMR Spectroscopy User Guide*

Related

dnode	Display list of valid limNET nodes (M,U)

email — Tool to Send Email

Description Called on a filename, this utility prompts for email addresses and sends the specified file.

Syntax email(filename)

See also email('textfile')

enter — Enter sample information for automation run (M,U)

Applicability Systems with an automatic sample changer.

Syntax enter<(file<,configuration_file>)>
(From UNIX) enter <file> <configuration_file>

Description Enables entry of sample information for automation runs, including the sample location, user information, solvent used, experiment or experiments to run, and arbitrary text information. enter('abc') creates a directory named abc. In this directory is a file named abc, which contains experiment information.

Arguments
`file` is the name of the file to be edited. The default is that enter prompts for this information. If the file already exists, new entries are appended to it.

`configuration_file` is the name of a user-supplied file that customizes enter for local use. Several configuration files are provided:

- `enter.conf` is used when defining an experiment when an automation run is not currently active.
- `auto.conf` is used when defining an experiment for a current automation run. The walkup macro is provided for this style of entering samples.
- `gilson.conf` is used with the VAST accessory.

Examples
(From VnmrJ or UNIX) `enter`
(From VnmrJ) `enter('mysamples')`
(From UNIX) `enter MySamples`
(From VnmrJ) `enter('mysamples','auto.conf')`

See also
NMR Spectroscopy User Guide; User Programming, VnmrJ Walkup

Related

auto	Set up an automation directory (C)
autogo	Start an automation run (C)
autoname	Prefix for automation data file (P)
autora	Resume a suspended automation run (C)
autosa	Suspend current automation run (C)
printer	Printer device (P)
status	Display status of all experiments (C)
walkup	Walkup automation (M)

enterdialog Start a dialog window using enterexp file (M)

Applicability
Systems with automation.

Syntax
`enterdialog`

Description
Internal macro used by enter to start a dialog window using the `enterexp` file in the `dialoglib` directory.

See also
NMR Spectroscopy User Guide; User Programming, VnmrJ Walkup

Related

enter	Enter sample information for automation run (M,U)

epage Emails Output

Description Used in place of the page command, this macro directs the output to email.

Syntax epage

Related page, eplot

eplot Emails PostScript

Description Used in place of the page command, this macro directs PostScript output to email.

See also eplot

Related page, epage

eread Transfer file from remote source (M,U)

Applicability Systems with limNET protocol software installed.

Syntax (From VnmrJ) `eread(local_file,remote_node,remote_file)`
(From UNIX) `eread local_file remote_node remote_file`

Description Copies a remote file to the local host. It will not overwrite a preexisting file.

Arguments `local_file` is the file name of the local host. If `local_file` is not a dot file (i.e., starts with "."), `eread` uses the "I1" and "I2" values of the remote file to create an extension and then append it to the local file name.

`remote_node` is a symbolic node name for a specified node file. Use the command dnode to list nodes defined on your system. The names of the remote computers or "nodes" available to the limNET protocol are contained in the file `/vnmr/nodes`. *Note that this is not the same file as the name of the remote computers available to the Internet protocol (IP), which are contained in the file* `/etc/hosts`. Each user only needs to know the "names" of relevant nodes.

`remote_file` is the name of file to be transferred from the remote host.

Examples (From VnmrJ) `eread('osv700','VXR4000','dsk1.osv700')`
(From UNIX) `eread osv700 VXR4000 dsk1.osv700`

See also *NMR Spectroscopy User Guide*

Related dnode Display list of valid limNET nodes (M,U)
ewrite Transfer file to remote destination (M,U)

ernst — Calculate the ernst angle

Syntax `ernst(t1,<90degree>)`

Applicability VnmrJ 3.1

Description Calculate the `ernst` angle pulse with a guess at `t1` and the 90-degree pulse calibration and sets `pw`. If there is a parameter `pw90` and no second parameter is entered, `pw90` is taken as the 90-degree pulse. An entered 2nd argument resets `pw90`.

errlog — Display Recent VNMR Error Messages

Syntax

Applicability VnmrJ 3.1

Description The `errlog` command displays the most recent VNMR error messages in the alphanumeric (`dg`) window.

Arguments Use the global parameter "`errloglen`" to control the number of lines that are displayed. If not defined, the program uses a value of 10 by default.

errloglen — Number of lines in error message display (P)

Description Sets the number of lines in the display of error messages by `errlog`.

Values Integer, default is 10.

See also *NMR Spectroscopy User Guide*

Related `errlog` Display recent error messages (P)

ewrite — Transfer file to remote destination (M,U)

Applicability Systems with limNET protocol software installed.

Syntax (From VnmrJ) `ewrite(local_file,remote_node,remote_file)`
(From UNIX) `ewrite local_file remote_node remote_file`

Description Takes a preexisting local file and copies it to a remote host. The file cannot preexist on the remote host.

Arguments `local_file` is the file name of the local host.

`remote_node` is a symbolic node name for a specified node file. Use the command `dnode` to list nodes defined on your system. The names of the remote computers or “nodes” available to the limNET protocol are contained in the file `/vnmr/nodes`. *Note that this is not the same file as the name of the remote computers available to the Internet*

Protocol (IP), which are contained in the file `/etc/hosts`. Each user only needs to know the “names” of relevant nodes.

`remote_file` is the name of file to be transferred from the remote host.

Examples (From VnmrJ) `ewrite('osv700','VXR4000','dsk1.osv700')`
(From UNIX) `ewrite osv700 VXR4000 dsk1.osv700`

See also *NMR Spectroscopy User Guide*

Related `dnode` Display list of valid limNET nodes (M,U)
`eread` Transfer file from remote source (M,U)

exec Execute a VNMR command

Syntax `exec('command') - execute a VNMR command`
`exec('command'):$ret -` execute a VNMR command and report success or failure

Applicability VnmrJ 3.1

Description The `exec` command allows an arbitrary VNMR command or macro to be executed. It lets a macro construct a character string which is a VNMR command or macro and then execute that command or macro.

Some macros and commands abort. This causes the calling macro to also abort. By using `exec` with a return value, whether or not the called macro aborted or not is returned as a macro variable. The calling macro is not aborted.

For example, in the simple macro macroB

`write`('line3','got to here')

If macroB aborts, the write command is not executed and the calling macro aborts.

If a return argument is given to `exec`, it will be set to 0 if the called macro aborts and it will be set to 1 if the called macro does not abort. For example, in the following macro

```
exec('macroB'):$ret
if ($ret = 0) then
  write('line3','macroB aborted')
else
  write('line3','macroB did not abort')
endif
```

one or the other write commands will execute, depending on whether macroB aborts. The calling macro does not abort, but continues executing its instructions.

The `aborton` and `abortoff` mechanism can also control whether or not the calling macro aborts if its called macro (macroB in the above examples) aborts. However, continued execution of the called macro is not guaranteed. For example, `abortoff`

macroB

`aborton`

write('line3','got to here')

will often execute the write command, whether or not macroB aborts. However, if macroB calls `aborton` and subsequently aborts, or if macroB calls another macro that calls `aborton`, and one of those macros aborts, then the calling macro will abort before the write command is executed. Using the `exec` command with a return argument, as in `exec('macro'):$ret` , guarantees that execution of the calling macro will continue.

Examples `exec($cmdstr):$ret` - execute the contents of $cmdstr as a VNMR command

execpars Set up the exec parameters (M)

Description Set up the exec parameters as listed in `/vnmr/execpars`.

See also *User Programming*

Related		
	`apptype`	Application type (P)
	`execplot`	Execute plotting macro (P)
	`execprep`	Execute prepare macro (P)
	`execprescan`	Execute prescan macro (p)
	`execproc`	Execute processing macro (P)
	`execsetup`	Execute setup macro (P)

execplot Execute plotting macro (P)

Description Defines which plotting macro to use to plot this experiment.

See also *User Programming*

Related		
	`apptype`	Application type (P)
	`plot`	Automatically plot spectra (M)

execprep Execute prepare macro (P)

Description Defines which prepare macro to use to prescan this experiment.

See also *User Programming*

Related		
	`apptype`	Application type (P)
	`acquire`	Acquire data (M)
	`plot`	Automatically plot spectra (M)

execprescan Execute prescan macro (P)

Description Defines which prescan macro to use to prescan this experiment.

See also *User Programming*

Related

apptype	Application type (P)
acquire	Acquire data (M)

execproc Execute processing macro (P)

Description Defines which processing macro to use to process this experiment.

See also *User Programming*

Related

apptype	Application type (P)
acquire	Acquire data (M)

execprocess Execute processing macro (P)

Description Defines which processing macro to use to process this experiment.

See also *User Programming*

execsetup Execute setup macro (P)

Description Defines which setup macro to use to prescan this experiment.

See also *User Programming*

Related

apptype	Application type (P)
cqexp	Load experiment from protocol (M)
sqexp	Load experiment from protocol (M)

exists

Syntax

```
exists(name,'parameter'[,tree]):$x - does a parameter
exist?
exists(name,'file'<,perm>):$x - does a file exist?
exists(name,'ascii'):$x - is a file an ASCII text file
exists(name,'directory'):$x - is a file a directory
exists(name,'parlib'):$x,$path - does a parlib entry
exist
exists(name,'psglib'):$x,$path - does a psglib entry
exist
```

```
exists(name,'command'):$x - does a command or macro
exist?
exists(name,'maclib'):$x - does a macro exist?
exists(name,directory<,'errval'>):$x - does a file or
directory exist in one of the "applications directories"
```

Applicability VnmrJ 3.1

Description Allows checking for the existence of a parameter, file, command, parlib entry, or macro from within a macro. Allows checking if a file is an ASCII text file or is a directory. Returns 1, if file or parameter exists, or the query is true; else 0. If the 'parameter' keyword is used, an optional variable tree name can be supplied. The variable trees are 'current', 'global', 'processed', 'usertree', and 'systemglobal'. The default tree is 'current'.

If the '`file`' keyword is used, an optional permission test can be supplied. Without the permission test, simple existence of the file is checked. Access permission can be checked by passing the character `r` for read permission, `w` for write permission, and `x` for execute permission. One, two, or three characters can be passed in a single argument. For example,

`exists`('/vnmr/conpar','file','rw')

checks not only that the file /vnmr/conpar exists, but also that the current user has read and write access to that file. The ascii option checks if the named file is an ascii file. The directory option checks if the named file is a directory.

The `parlib` name will be searched for. If it is not found, a .par will be appended and the appended name will be searched for. The `parlib` option will also return the absolute path of the parameter set. The search path for `parlib` is defined by the VnmrJ administrator interface, using the "applications directories", or `appdirs`.

The `psglib` name will be searched for. If it is not found, a .c will be appended and the appended name will be searched for. The `psglib` option will also return the absolute path of the parameter set. The search path for `psglib` is defined by the VnmrJ administrator interface, using the "applications directories", or `appdirs`.

Macros may reside in various places, as determined by the "applications directories", or appdirs. Typical places include the users vnmrsys/maclib directory and /vnmr/maclib.

When macros are executed, the `appdirs` are searched in order. Exists will return a 0 if the macro is not found in any of the `appdirs`. It will return a 1, 2, or larger integer, depending on if it is found in the first, second, third, etc appdir.

The command keyword is very similar to the maclib keyword, except that it firsts checks to see if the name represents a built-in Vnmr command.

If the name is neither a built-in command nor a macro, exists will return a 0. If the name represents a built-in command, exists will return a 1. If name is a macro, exists will return either 2, 3, 4, or 5. The return value identifies in which directory the macro is located.

The number is 1 greater than the value returned by the maclib keyword. That is, if the command

exists('macroname','maclib'):r1

sets r1 equal to 1, then the command

```
exists('macroname','command'):r1
```

will set r1 equal to 2.

The `exists` command with the maclib keyword is a specific case of a general mechanism to search for files and directories in the "applications directories", or `appdirs`. The first argument to exists is a file name and the second argument is any subdirectory in an `appdir`. For example, the second argument could be the following:

shapelib - to search for shapes.

manual - to search for manuals

probes - to search for probes

shims - to search for shims

It can be any directory in an `appdir`. It need not be a standard directory. For example, it could be bin to search for standalone executable programs. One could execute these standalone executable programs using a construction along the following lines.

```
exists($myprog,'bin'):$e,$myprogPath
  if ($e) then
    shell($myprogPath):$res
  else
    write('line3','%s: Program %s has not been installed',$0,$myprog)
  endif
```

The second argument to exists can be set to '' to search for files in the top-level of the `appdirs`. For example,

```
exists('pulsecal','')
```

will search for pulsecal in the top-level of all `appdirs`. The directory name can also be multi-level, as in

exists(probename,'probes/'+probe)

The first argument may also be set to '', in which case exists will check for directories in the `appdirs`.

This generic form of `exists` will return one or two values to the calling macro. The first return value is an integer indicating in which appdir the file is found. The exists command will return a 0 if the file is not found. It will return a 1, 2, or larger integer, depending on if it is found in the first, second, third, etc appdir. An optional third argument can be provided. This will be the return value if the file is not found. For example,

```
exists('nomacro','maclib',-1):$ok
```

will set $ok to -1 if the "nomacro" does not exist in any of the `appdirs`. This can be used by the interface designed so that a button may be either "grayed out" of removed if a macro or some other file does not exist.

The second optional return value is the absolute path to the found file. If the file does not exist, the second return value will not be set.

See also — See the `which` macro for an example on the use of the command keyword.

exit — Macro to call vnmrexit

Syntax — `exit`
`vnmrexit`

Applicability — VnmrJ 3.1

Description — The command `vnmrexit` exits from the vnmr system in a graceful manner. It writes parameters and data to the disk, removes lock files and restores the terminals(if on a GraphOn). The macro `exit` calls the command `vnmrexit` to exit from `vnmr`. As a macro, `exit` provides a user some flexibility in defining other things to do when exiting.

expactive — Determine if the experiment has an active acquisition

Syntax — `expactive(n)<:$ans> - determine if experiment n has an active acquisition`
`expactive('user')<:$ans> - determine if current user has an active or queued experiment`
`expactive('auto')<:$ans> - determine if system is in automation mode`
`expactive('current')<:$ans> - determine current active experiment number and user`
`expactive<:$ans> - determine if current experiment has an active acquisition`

Applicability — VnmrJ 3.1

Description — `expactive` will determine whether an acquisition is active or pending in the current experiment. An experiment number n, where n is a number from 1 to 9999, may be supplied to expactive to determine if an acquisition is active or pending in experiment n.

Arguments — Without a return argument, `expactive` displays the results on line 3. If a return argument is appended to the `expactive` command, it will be set to the following:

- -1 - acquisition is not possible (for example, it is a data station)
- 0 - no acquisition is active in the requested experiment
- 1 - an acquisition is active in the requested experiment

- 2 or larger if an acquisition is queued in the requested experiment. Subtract 1 from the value to determine its position in the acquisition queue.

If the keyword 'user' is supplied as an argument, `expactive` will determine if the current user has an active or queued experiment. Without a return argument, `expactive('user')` displays the results on line 3. If a return argument is appended to the `expactive('user')` command, it will be set as in the case above.

If the keyword 'auto' is supplied as an argument, expactive will determine if the system is in automation mode. Without a return argument, `expactive('auto')` displays the results on line 3. If a return argument is appended to the `expactive('auto')` command, it will be set to 1 if the system is in automation mode, 0 otherwise.

If the keyword 'current' is supplied as an argument, expactive will determine which experiment, if any, has an active acquisition command running. Without a return argument, `expactive('current')` displays results on line 3. An experiment is still considered active if it holds up additional acquisitions during its `wexp` processing by means of the 'wait' flag. If a return argument is appended to the `expactive('current'):$exp` command, it will be set to the following:

- -1 - acquisition is not possible (for example, it is a data station)
- 0 - no acquisition is active
- n - an acquisition is active in experiment "n"

If a second return argument is appended to the

expactive('current'):$exp,$user

command, the second argument will be set to the user that started the acquisition. If the system is running in automation mode, this second argument will be set to '`auto`'. If no acquisition is running, this second argument will be set to '`nobody`'.

expfit — Unix program for making a least squares fit to a polynomial or exponential curve.

Syntax `expfit option(s) <analyze.inp  >analyze.list`

Applicability VnmrJ 3.1

Description The program expfit does a least-squares curve fitting to the data supplied in 'analyze.inp'. Macros are available for the specialized uses of analyze such as '`t1`' and '`kinetics`'. They avoid the need for the user to select options and get the correct file format. In the regression mode, the type of curve fitting, ('poly1',...) must be selected. For regression (generalized curve fitting), the regression section in the Operation Manual gives the input file format and describes the menus that permit options choices indirectly through menu buttons.

Files

The text file analyze.inp which for t1, t2, kinetics, contact_time, and regression, contains:
<optional descriptive text line>
<optional y-axis title - regression only>
number of peaks(data sets)   number of (x,y) pairs per peak and, regression only, x scale type   y scale type
<NEXT number of (x,y) pairs for this peak >
peak index
x y   (first peak,first pair)
x y   (first peak,second pair)
......
<NEXT number of (x,y) pairs for this peak >
peak index
x y   (second peak, first pair)
......
In the regression mode the line beginning with 'NEXT' is inserted at the start of each data set when the number of pairs per peak is variable. In this case the header contains the maximum number of pairs per peak. For t1, t2, kinetics, and contact_time, information from the file 'fp.out' and from the array 'xarray' are used to construct this file, therefore, it is necessary to run 'fp' prior to 'analyze'. For regression, this file is made by running "expl('regression')". For 'diffusion', 'contact_time', and, if not in regression mode, poly1 and poly2, it is slightly different:
List of <number> x-y data pairs (6 strings)
<Descriptive text line>
<X-values> <Y-values> (2 strings without blanks)
x y   (first peak,first pair) (continues as above)
'expfit' also makes a file 'analyze.out', which is used by 'expl' to display the results of the analysis in addition to output to the standard output which is usually directed to 'analyze.list'.

Options

The following options are implemented in 'analyze':
t1, ,Perform T1 analysis (default)
t2Perform T2 analysis
`kinetics`   Perform kinetics analysis decreasing peak height
`increment`   Perform kinetics analysis with increasing peak height
listExtended listing for each peak
`diffusion`   A special analysis for diffusion experiments
contact_time A special analysis for solids cross-polarization spin-lock experiments
`regression` Sets regression mode, signifies generalized curve fitting with choices poly1, poly2, poly3, and exp
poly0With regression, calculates mean
poly1With regression, a linear fitting
poly2With regression, a quadratic fitting
poly3With regression, a cubic curve fitting
expWith regression, an exponential curve fitting

Examples   expfit d2 T1 list <analyze.inp >analyze.out
expfit regression exp list <analyze.inp >analyze.out

expl — Display data on the screen

Syntax
```
expl- display the data for all lines on the screen
expl(line#, line#,..)- display selected lines only
expl('regression',line#,..) - display selected data sets
for regression analysis
```

Applicability VnmrJ 3.1

Description Display or plot exponential curves resulting from t1, t2, or Kinetics analysis. Display or plot of Polynomial Curves from Diffusion or other type of analysis. No argument displays first 8 curves if that many along with the data points. Otherwise selected curves are plotted. sc, wc, sc2, and wc2 control the size of plot.

Options

'regression' signifies the beginning of generalized curve fitting. Expl displays the data in 'regression.inp' as unconnected points, and also uses 'regression.inp' to create the file 'analyze.inp', which serves as input to 'analyze' for curve fitting.

'linear', 'square', 'log' provide for plotting of the data points against the square or log of the data. The first keyword controls x-axis scale, the second keyword controls the y-axis. Default is to 'linear'.

'link' causes the data points to be connected rather than a plot of the theoretical curve.

'nocurve' produces a plot of data points only.

'tinysymbol' produces a plot with small-scale data point symbols.

'nosymbol' produces a plot of the curve only.

'noclear' does not erase the graphics screen before drawing the plot.

'oldbox' is used to plot an additional curve on an existing plot. Only the first data set in analyze.out is plotted. It causes the program to get box and scale description from expl.out in the current experiment. When the 'oldbox' option is used, a required second argument identifies the curve number and data point symbol, which will be used to represent the data.

This second argument is a number from 1 to 8.

'file' followed by a filename replaces analyze.out as the input to expl.

Files:

'analyze.out' file is the data input file except for regression when it is 'regression.inp'.

'expl.out' saves certain display/plot parameters.

Format for regression input, 'regression.inp':

Text Line (Optional)
Second text line (Optional) displayed along Y scale
nsets npairs
<NEXT>
x y (first set, first pair)
x y (first set, second pair)
......
<NEXT>
x y (second set, first pair)
......

The optional text lines must not begin with a digit.

The line beginning with 'NEXT' is inserted at the start of each data set

when the number of pairs per peak is variable. In this case, set 'nsets' and 'npairs' to 0.
Limits:
2048 points maximum from a data set.
2048 points maximum from all sets displayed/plotted.
8 data sets maximum displayed/plotted.
128 data sets maximum are read.

Examples
`expl`- display from the first up to the sixth curve with data points from 'analyze.out'
`expl`(1,3,6)- display curves with indexes 1, 3, and 6 from 'analyze.out' with data points
`expl`(1,3,6)- plot the data
`expl`('regression')- display the data in the first up to the sixth data set in 'regression.inp'
`expl`('regression',4,5)- display the data in the fourth and the fifth data set in 'regression.inp'

See also
See `expl` in the Commands Manual for the file format of `analyze.out`

`expladd` Add another diffusion analysis to current display (M)

Applicability: Systems with the diffusion option.

Syntax: `expladd(integral_region)`

Description: Adds results of another diffusion analysis to the currently displayed results.

Arguments: `integral_region` specifies the number of the region whose results are to be added to the existing graph.

Examples: `expladd(1)`

See also: *NMR Spectroscopy User Guide*

Related:

`expl`	Display exponential or polynomial curves (C)
`pexpl`	Plot exponential or polynomial curves (C)
`pexpladd`	Add another diffusion analysis to current plot (M)

`explib` Display experiment library (M)

Syntax: explib

Applicability: VnmrJ 3.1

Description: Displays the currently available experiment files. For each experiment, `explib` displays the name of the experiment and its subexperiments, whether an acquisition is active or its position in the acquisition queue, the current size of the experiments, the pulse sequence currently active in the experiments, and the first 50 characters of the text file in the experiment. `explib` also displays a message if the system is in automation mode.

See also *NMR Spectroscopy User Guide*; *VnmrJ Walkup*

explist — Display current experiment chain and approx. time for each (M)

See also Displays approximate time for each experiment in a chained experiment.

Related `autotime` Display approximate time for automation (M)

explog — Display an experiment's log file

Applicability VnmrJ 3.1

Description Each acquisition generates a log file which includes when the experiment started, any acquisition errors which may have occurred, and when the experiment finished. This information may be displayed with the explog macro. This information is stored in the experiment's acqfil directory in a text file named `log`.

exptime — Display experiment time

Syntax
```
exptime<:$time,$msg>
exptime('filename')<:$time,$msg>
exptime('usertree')<:$time,$msg>
```

Applicability VnmrJ 3.1

Description `exptime` estimates the experiment time for the current `seqfil`, using the parameters in the current experiment. "`exptime('filename')`" estimates the experiment time of the specified filename.
If `exptime` fails, a -1 is returned. If a second return argument is given, the error message is returned. The `exptime` command will accept the filename of a parameter set. It will use those parameters for the time calculations. The return values are the same as above.

Arguments `exptime` estimates the experiment time, using the parameters in the current experiment. If a return argument is used, the time in seconds is returned. The `exptime` command will accept the keyword `usertree`, and use the parameters currently loaded into the `usertree` for the time calucations. The return values are the same as above.

Examples
```
exptime
exptime('/vnmr/parlib/PROTON.par/procpar'):$sec
fread('/vnmr/parlib/HSBC.par/procpar','usertree')
```

```
setvalue('ni',600,'usertree')
exptime('usertree'):$sec
```

F

f	Set display parameters to full spectrum
f19	Automated fluorine acquisition (M)
f19p	Process 1D fluorine spectra (M)
f1coef	Coefficient to construct F1 interferogram (P)
f2coef	Coefficient to construct F2 interferogram (P)
fastuserlogin	Gateway macro for fastuserlogin function. (M)
fattn	Fine attenuator (P)
fb	Filter bandwidth (P)
fbc	Apply baseline correction for each spectrum in an array (M)
fdm1	Set, write 1D FDM parameters, run FDM (M)
fid_scan	Start up the interactive acquisition display process
fiddc3d	3D time-domain dc correction (P)
fiddle	Perform reference deconvolution (M)
fiddle_examples	Illustrates some of the simple ways that fiddle can be used to extract worthwhile results from poor quality data
fiddled	Perform reference deconvolution subtracting alternate FIDs (C)
fiddleu	Perform reference deconvolution subtracting successive FIDs (C)
fiddle2d	Perform 2D reference deconvolution (C)
fiddle2D	Perform 2D reference deconvolution (C)
fiddle2dd	2D reference deconvolution subtracting alternate FIDs (C)
fiddle2Dd	2D reference deconvolution subtracting alternate FIDs (C)
fidmax	Find the maximum point in an FID (C)
fidpar	Add parameters for FID display in current experiment (M)
fidsave	Save data (M)
fifolpsize	FIFO loop size (P)

file	File name of parameter set (P)
files	Interactively handle files (C)
filesinfo	Return file information for files display (C)
filtfile	File of FIR digital filter coefficients (P)
findxmlmenu	Find an xml menu (M)
fitspec	Perform spectrum deconvolution (C, U)
fixgrd	Convert gauss/cm value to DAC (M)
fixpar	Correct parameter characteristics in experiment (M)
fixpar3rf	Create parameters for third rf channel (M)
fixpar4rf	Create parameters for fourth rf channel (M)
fixpar5rf	Create parameters for fifth rf channel (M)
fixgrdR	Converts Gradient Strength to DAC values
fixup	Adjust parameter values selected by setup macros (M)
fixpsg	Update psg libraries (M)
flashc	Convert compressed 2D data to standard 2D format (C)
flipflop	Set up parameters for FLIPFLOP pulse sequence (M)
Fluorine	Set up parameters for 19F experiment (M)
flush	Write out data in memory (C)
fn	Fourier number in directly detected dimension (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
fn2	Fourier number in 2nd indirectly detected dimension (P)
fn2D	Fourier number to build up 2D DOSY display in freq. domain (P)
focus	Send keyboard focus to input window (C)
foldcc	Fold INADEQUATE data about two-quantum axis (C)
foldj	Fold J-resolved 2D spectrum about f_1=0 axis (C)
foldt	Fold COSY-like spectrum along diagonal axis (C)
fontselect	Open FontSelect window (C)
format	Format a real number or convert a string for output (C)
fp	Find peak heights or phases (C)
fpi	Report integral values from arrayed spectra. (M)
fpmult	First point multiplier for np FID data (P)
fpmult	First point multiplier for "np" FID data

fpmult1	First point multiplier for ni interferogram data (P)
fpmult2	First point multiplier for ni2 interferogram data (P)
fr	Full recall of a display parameter set (M)
framecmd	Create a new frame of image, text, and inset with 'new' option
fread	Read parameters from file and load them into a tree (C)
fsave	Save parameters from a tree to a file (C)
fsq	Frequency-shifted quadrature detection (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft1da	Fourier transform phase-sensitive data (M)
ft1dac	Combine arrayed 2D FID matrices (M)
ft2d	Fourier transform 2D data (C)
ft2da	Fourier transform phase-sensitive data (M)
ft2dac	Combine arrayed 2D FID matrices (M)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M,U)
full	Set display limits for a full screen (C)
fullsq	Display largest square 2D display (M)
fullt	Set display limits for a full screen with room for traces (C)

f   Set display parameters to full spectrum

Syntax f

Applicability VnmrJ 3.1

Description This commands sets the display parameters "sp" and "wp" up for a full display of a 1D spectrum. If an FID is displayed, the parameters "sf" and "wf" will be set for a full display. In multi-dimensional data sets, the parameters for both displayed dimensions will be set up. For 2D data sets, the parameters "sp", "wp", "sp1", and "wp1" would be set. For planes of higher dimensional data sets, the appropriate two groups of sp wp, sp1 wp1, and sp2 wp2, parameter pairs will be set.

f19   Automated fluorine acquisition (M)

Syntax f19<(solvent)>

Description	Prepares parameters for automatically acquiring a standard ^{19}F spectrum. The parameter `wexp` is set to `'procplot'` for standard processing. If `f19` is used as the command for automation via the `enter` program, then the macro `au` is supplied automatically and should not be entered on the MACRO line of the `enter` program. However, it is possible to customize the standard `f19` macro on the MACRO line by following it with additional commands and parameters. For example, `f19 nt=1` uses the standard `f19` setup but with only one transient.
Arguments	`solvent` is the name of the solvent. In automation mode, the solvent is supplied by the `enter` program. The default is `'CDCl3'`
Examples	`f19` `f19('DMSO')`
See also	*NMR Spectroscopy User Guide*

Related

`au`	Submit experiment to acquisition and process data (M)
`enter`	Enter sample information for automation run (C)
`f19p`	Process 1D fluorine spectra (M)
`proc1d`	Processing macro for simple (non-arrayed) 1D spectra (M)
`procplot`	Automatically process FIDs (M)
`wexp`	When experiment completes (P)

`f19p` Process 1D fluorine spectra (M)

Description	Processes non-arrayed 1D fluorine spectra using a set of standard macros. `f19p` is called by `proc1d`, but can also be used directly. Fully automatic processing (up to a point where a spectrum could be plotted) is provided: Fourier transformation (using preset weighting functions), automatic phasing (`aphx` macro), select integral regions (`hregions` macro), adjust integral size (`integrate` macro), vertical scale adjustment (`vsadjc` macro), avoiding excessive noise (`noislm` macro), threshold adjustment (if required, `thadj` macro), and referencing to the TMS signal, if present (`tmsref` macro).
See also	*NMR Spectroscopy User Guide*

Related

`aphx`	Perform optimized automatic phasing (M)
`f19`	Automated fluorine acquisition (M)
`hregions`	Select integral regions for proton spectra (M)
`integrate`	Automatically integrate 1D spectrum (M)
`noislm`	Avoids excessive noise (M)
`proc1d`	Processing macro for simple (non-arrayed) 1D spectra (M)
`thadj`	Adjust threshold (M)
`tmsref`	Reference spectrum to TMS line (M)
`vsadjh`	Adjust vertical scale for proton spectra (M)

f1coef Coefficient to construct F1 interferogram (P)

Description Holds the coefficient to construct an F1 interferogram for 2D and 3D transformation. Coefficients are used by the ft2da and ft3d macros. If f1coef has a null value, ft2da uses the “standard” coefficients. f1coef is created by the par2d macro.

Values Series of coefficients, separated by spaces (not a comma), and stored as a string variable. For example, the coefficient for standard States-Hypercomplex data set is f1coef='1 0 0 0 0 0 -1 0'.

See also *NMR Spectroscopy User Guide*

Related	
f2coef	Coefficient to construct F2 interferogram (P)
ft2da	Fourier transform phase-sensitive data (M)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M,U)
make3dcoef	Make 3D coefficients file from 2D coefficients (M)
par2d	Create 2D acquisition, processing, display parameters (M)

f2coef Coefficient to construct F2 interferogram (P)

Description Holds the coefficient to construct an F2 interferogram for 2D and 3D transformation. Coefficients are used by the ft2da('ni2') and ft3d macros. If f2coef has a null value, ft2da('ni2') uses the “standard” coefficients. f2coef is created by the par3d macro.

Values Series of coefficients, separated by spaces (not a comma), and stored as a string variable. For example, the coefficient for standard States-Hypercomplex data set is f2coef='1 0 0 0 0 0 -1 0'.

fastuserlogin (M) Gateway macro for fastuserlogin function.

Syntax

Applicability VnmrJ 3.1

Description On systems with VnmrJ 3.1 and above, this macro manages the FASTuser switch located in VnmrJ-User Preferences. Enabling the FASTuser switch allows users to quickly login and logout of VnmrJ.

Arguments

Examples

See also

fattn Fine attenuator (P)

Description Configuration parameter for whether the current rf channel has a fine attenuator. The value is set using the label Fine Attenuator in the Spectrometer Configuration window (opened from config).

Values 0 specifies the fine attenuator is not present on the channel (Not Present choice in Spectrometer Configuration window).

4095 specifies the fine attenuator is present on the channel (Present choice in Spectrometer Configuration window).

See also *VnmrJ Installation and Administration; User Guide: Solids; CP/MAS Installation*

Related

config	Display current configuration and possibly change it (M)
dpwrf	First decoupler fine power (P)
tpwrf	Observe transmitter fine power (P)

fb Filter bandwidth (P)

Description Sets the bandwidth of the audio filters, which prevents noise of higher frequency than the spectral limits from “folding in” to the spectrum. Because the transmitter is in the center of the spectrum, the range of audio frequencies that must be filtered out is half the spectral width sw (e.g., for a spectral width of 4000 Hz, frequencies higher than ±2000 Hz should be filtered out). The audio filters have some attenuation at frequencies lower than their nominal cutoff frequency, which is the frequency at which signals have been attenuated by 3 dB (50%). This impacts on quantitative accuracy near the edges of the spectrum so that the standard value of fb is 10% more than half of sw.

fb is automatically changed whenever the spectral width sw is changed and thus is normally not a user-entered parameter. For example, typing sw=4000 automatically sets fb=2200, which is 10% more than 2000 Hz. After changing the value of sw, fb can be changed.

Values if sw is 500,000 or less: 1000 to 256000 Hz, 1000-Hz steps.

if sw is greater than 500,000: 256 kHz, 1 MHz.

See also *NMR Spectroscopy User Guide*

Related

sw	Spectral width in directly detected dimension (P)
mrfb	Set the filter bandwidths for multiple receivers (P)

fbc Applies 'bc' type baseline correction to all the spectra in an array

Syntax fbc

Applicability VnmrJ 3.1

Description The macro `fbc` applies 'bc' type baseline correction to all the spectra in an array. The partial integral mode should be used to set integral regions to include all significant signals, while leaving as large an area of baseline as possible blank. This minimises systematic errors in diffusion coefficient fits caused by baseline errors.

fdm1 Set, write 1D FDM parameters, run FDM (M)

Syntax `fdm1<(filename<,n1, v1<, n2, v2<...>>>)>`
or
`fdm1 (i)` for the i-th trace

Description Sets 1D Filter Diagonalization Method (FDM) parameters to the default values, writes the parameters to the `curexp/datdir/fdm1.inparm` file, and runs a stand-alone C++ program (`/vnmr/bin/fdm1d`).

Arguments `filename` is the FID file; the default is `curexp+'acqfil/fid'`.

`n1, n2...` is one or more following variable names (the order is arbitrary):

`axis`	–1 (default) to reverse the spec.
`cheat`	No cheat if `cheat=1`, lines are narrower if `cheat<1`.
`cheatmore`	No cheatmore if `cheatmore=0`.
`error`	Error threshold for throwing away poles.
`fidfmt`	FID format: VnmrJ or ASCII.
`fdm`	1 for FDM; –1 for Digital or Discrete Fourier Transform.
`fn_Sp1D`	Spectrum file; default is `curexp/datadir/fdm1.parm`.
`Gamm`	Smoothing width (line broadening).
`Gcut`	Maximum width for a pole.
`idat`	Data type of ASCII FID file –4 for complex data, ignored if data is in VnmrJ format.
`i_fid`	The i-th trace of the FID.
`kcoef`	If `kcoef > 0`, use `'complicated' dk(k). -1` is always preferred.
`Nb`	Number of basis functions in a single window.
`Nbc`	Number of coarse basis vectors.
`Npower`	Number of spectrum data points.
`Nsig`	Number of points to use.
`Nskip`	Number of points to skip.
`par`	Line list file; default is `curexp/datadir/fdm1.parm`
`rho`	`rho=1` is optimal.
`specfmt`	Spec format: VnmrJ or ASCII.
`spectyp`	Spectrum type: complex (default), real imag, or abs.
`ssw`	A test parameter.
`t0`	Delay of the first point.
`theta`	Overall phase of FID (rp in radians).
`wmax`	Maximum spectrum frequency in hertz.
`wmin`	Minimum spectrum frequency in hertz.

`v1, v2...` is the value for the variable(s).

Examples `fdm1('cheat',0.8)`
`fdm1('Nsig',3000,'Nb',20,1'Gamm',0.5)`

See also *NMR Spectroscopy User Guide*

fid_scan Start up the interactive acquisition display process

Syntax `fid_scan`

Applicability VnmrJ 3.1

Description The interactive acquisition display process allows interactive shimming on the FID or spectrum. The pulse sequence and parameter set for the FID / spectrum display is whatever is set in the current experiment / workspace. The normal interactive tools for FID and spectral displays (df and ds commands) are available in this interactive mode. Automatic locking, shimming, steady states, and robot control are turned off by passing the 'fidscan' argument to the au command.

The mechanism used for this interactive display is based on the au / wbs tools. The fid_scan macro does an au with the bsclear and fidscan arguments and sets wbs='fid_display'. The fid_display macro does the actual data display at block size intervals. The fidscanmode parameter controls the type of display to use. It is a list of flag characters to select various options. Possible values for the "fidscanmode" parameter include:

- 'r' - displays the reals (as a trace, not in "filled" mode)
- 'i' - displays the imaginaries
- 'ri' - displays both the reals and the imaginaries
- 'f' - displays the FID in "filled" mode. In this mode, the 'envelope' and 'dots' mode (see dotflag parameter) are not available.
- 'rf' - display the "reals" in "filled" mode.
- 's' - displays the spectrum
- 'e' - displays the envelope

By default, a block size of 1 is used for fidscan mode. However, this can be changed by creating and setting a 'fidshimnt' parameter. Setting the fidshimnt=1 has the special effect of turning automatic phase cycling (i.e., oph) off. Setting fidshimnt=8, for example, will average 8 scans before the result is displayed.

Related ft3d Perform a 3D FT on a 3D FID data set

fiddc3d Flag for 3D time-domain DC correction

Syntax `fiddc3d`

Applicability VnmrJ 3.1

Description fiddc3d is a flag whose default value is 'nnn'. fiddc3d is created by the macro `par3d` if the former does not already exist. The first character of fiddc3d in the 3-character string refers to the F3 dimension (sw,np,fn); the second character, to the F1 dimension (sw1,ni,fn1); and the third character, to the F2 dimension (sw2,ni2,fn2). Each character may take one of two values: 'n', for no time-domain DC correction along the relevant dimension, and 'y', for time-domain DC correction along the relevant dimension.

The time-domain DC correction occurs immediately after any LP (linear prediction) operations and before all other operations on the time-domain data.

Related `ft3d` Perform a 3D FT on a 3D FID data set

fiddle Perform reference deconvolution

Syntax `fiddle('option'[,'filename',][,'option',['filename']][,startno][,finishno][,increment])`

Applicability VnmrJ 3.1

Description This program performs reference deconvolution, using a reference signal with known characteristics to correct instrumental errors in experimental 1D or 2D spectra. The commands can take multiple string and numeric arguments, in the format descibed under OPTIONS below.

Reference deconvolution of 1D spectra

Only spectra that contain a well-resolved reference signal dominated by a single component (i.e. not a simple multiplet) are suitable for reference deconvolution. Fourier transform the raw fid with `ft`, preferably having zero filled (i.e. set `fn` >= 2*np). (If there are sinc wiggles, use `wft` with `gf` = at*0.6.) Set the reference line to the chosen signal using the `rl` command, and then use two cursors either side of the line to define a region of spectrum which includes all of the reference signal plus a little clear baseline but no other signals. This reference region will be used to define the instrumental lineshape.

Next, decide what lineshape you would like to convert the instrumental lineshape to, and set the weighting parameters accordingly. Thus if you want a 1 Hz wide Lorentzian, set lb to 1 and all other weighting parameters to 'n'. Bear in mind the signal-to-noise ratio penalty for resolution enhancement: if the experimental line is 2 Hz wide and you set `lb`=0, you get an infinitely sharp line with infinitely poor S/N. For most purposes a sensible strategy is to set lb to _minus_ the expected _natural_ linewidth, and choose gf to give reasonable S/N; this should convert the instrumental lineshape to Gaussian. Where the signals of interest are broader than those of the reference, resolution enhancement can easily be obtained by making lb more negative. Once you have set the weighting parameters, the command `fiddle` will carry out the reference deconvolution and display the corrected spectrum. The integral should remain unchanged, so any resolution enhancement will result in an increase in the amplitude of both signal and noise. To save the corrected data it is necessary to use the option '`writefid`' when doing the reference deconvolution, e.g. `fiddle('writefid','correctedfid')` will store the file 'correctedfid.fid' in the current working directory.

The options '`writefid`','<filename>' and '`readcf`','<filename>' will write and read the correction function respectively. Thus performing reference deconvolution on one fid using fiddle with the 'writecf' option and then using fiddle with 'readcf' to process another fid will use the first correction function to correct the second fid. This can be useful for heteronuclear lineshape correction (provided that the spectral widths for the two nuclei are in the ratio of the respective magnetogyric ratios), or for correcting spectra in which a reference signal has been suppressed (e.g. an INADEQUATE spectrum could be corrected for lineshape errors using a correction function derived from the normal carbon spectrum).

To correct a series of spectra in an arrayed or 2D experiment, use numeric arguments just as with ft: fiddle(1) will correct spectrum 1, fiddle(2,3) spectra 2 and 3, and so on.

Many reference signals have satellites; for example as well as the familiar one-bond carbon-13 satellites, TMS has singlet satellite signals from coupling to silicon-29 and quartet satellites (normally unresolved) from three-bond coupling to carbon-13. For most purposes carbon-13 satellites are small enough to be ignored, but where high accuracy is required or there are stronger (e.g. silicon-29) satellites, satellite signals can be included in the specified form of the ideal reference signal by invoking the 'satellites' option. The directory/vnmr/satellites contains a file TMS which contains details of the TMS satellite signals; the command fiddle ('satellites','TMS') will allow for the satellite signals when deconvoluting using TMS as a reference. For information on how to construct satellite files for other reference signals, see the file /vnmr/satellites/README.

To perform corrected difference spectroscopy, use fiddled to produce the corrected difference between successive spectra (this will halve arraydim). Since the main aim of reference deconvolution here is to optimise the purity of the difference spectrum, the target lineshape would normally be chosen to give the best possible S/N; this corresponds to choosing a target lineshape approximately twice the width of the raw experimental signals of interest. The command fiddleu produces corrected differences between successive fids and the first fid.

Reference deconvolution of 2D spectra

The commands fiddle2d/fiddle2D and fiddle2dd/fiddle2Dd function in just the same way as the parent fiddle program. Since the principal objective in 2D reference deconvolution is usually the reduction of t1-noise, ideal lineshape parameters are normally chosen for optimum S/N ratio rather than resolution enhancement. To perform 2D reference deconvolution, choose fn (preferably >=2*np) and fn1, then ft the raw data (as mentioned earlier, if there is significant signal left at the end of at it may be necessary to use wft with gf set). Display the first increment with ds(1), adjust the phase of the reference signal, and use rl to select the reference signal. In earlier versions, it was necessary to create a parameter phinc to anticipate the changes in the reference signal phase with increasing evolution time, but the current algorithm adjusts the phase automatically (unless the option 'noaph' is selected). The deconvolution will set the reference signal phase as a function of t1 so as to place the reference signal at frequency rfp1 in f1, so remember to set rfl1 and rfp1 before using fiddle2D or the f1 frequencies may change unexpectedly.

Define the reference region with the two cursors as usual, then type the command fiddle2D('writefid','<filename>') (or fiddle2D if a 2D difference spectrum is required, as with corrected HMBC). The 'writefid' option is essential, as fiddle2D on its own does not store the corrected time-domain data. If phase-sensitive gradient-enhanced 2D data are to be processed, alternate fids will have opposite phase modulations (i.e. the experimental array will alternate N-type and P-type pathways), and the option 'alternate' should be used.

Once the deconvolution is complete, the corrected 2D fid data can be read into an experiment and processed as normal (though if fiddle2D has been used, arraydim will no longer match the arrays set and it may be necessary to set the arguments to wft2d explicitly rather than using wft2da, or adjust the parameters manually).

Arguments The options available are as follows:

- alternate: Alternate reference phase +- (for phase sensitive gradient 2D data)
- autophase: Automatically adjust phase
- displaycf: Stop at display of correction function
- fittedbaseline: Use cubic spline baseline correction defined by the choice of integral regions
- invert: Invert the corrected difference spectrum/spectra
- noaph: Do not automatically adjust zero order phase of reference region
- nodc: Do not use dc correction of reference region
- nohilbert: Do not use Hilbert transform algorithm; use extrapolated dispersion mode reference signal unless option ...
- noextrap: Is also used
- normalise: Keep the corrected spectrum integrals equal to that of the first spectrum
- readcf: Read correction function from file '<filename>'; the argument 'filename' must immediately follow 'readcf'
- satellites: Use satellites defined in '<filename>' in ideal reference region; '<filename>' should be in /vnmr/satellites
- stop1: Stop at display of experimental reference fid
- stop2: Stop at display of correction function
- stop3: Stop at display of corrected fid
- stop4: Stop at display of first corrected fid
- verbose: Display information about the course of the processing in the main window
- writecf: Write correction function to file '<filename>'; the argument 'filename' must immediately follow 'writecf'

- writefid: Write out corrected fid to '<filename>'; if '<filename>' does not begin with / it is assumed to be in the current working directory

See also J. Taquin, Rev. Physique App., 14 669 (1979).

G.A. Morris, JMR 80 547 (1988).

G.A. Morris & D. Cowburn, MRC 27 1085 (1989).

A. Gibbs & G.A. Morris JMR 91 77 (1991).

A. Gibbs, G.A. Morris, A.G. Swanson and D. Cowburn, J.Magn.Reson. 101, 351-356 (1993).

G.A. Morris, in Chapter 16 of "Signal Treatment and Signal Analysis in NMR", ed. D.N. Rutledge, Elsevier, 1997.

G.A. Morris, H. Barjat and T.J. Horne, Prog. NMR Spectrosc., 31, 197 (1997).

Related

fiddled	Perform subtracting alternate fids
fiddleu	Perform subtracting successive fids from the first
fiddle2D	Perform 2D reference deconvolution
fiddle2dd	Perform 2D reference deconvolution subtracting alternate fids

fiddle_examples Illustrates some of the simple ways that fiddle can be used to extract worthwhile results from poor quality data

Applicability VnmrJ 3.1

Description This is a small collection of fids recorded on an old XL300 and converted to Vnmr format, and illustrates some of the simple ways that fiddle can be used to extract worthwhile results from poor quality data. The three files are:
mixture: a mixture of acetone and ethanol in CDCl3, with very poor shimming and severe spinning sidebands
ODCB: a (folded) spectrum of a sample containing ODCB and TMS, again recorded with very poor shimming and severe spinning sidebands
NOED: an arrayed pair of fids for an NOE difference experiment with gated irradiation [see Magn.Reson. Chem. 27, 1085-1089 (1989)], this time with OK shimming (but nasty decoupler spikes)
To try out fiddle with these files, simply load them into an experiment, type text and follow the instructions displayed.

fiddled Perform reference deconvolution subtracting alternate FIDs (C)

Description Produces the corrected difference between successive spectra. Refer to the description of fiddle for details.

See also *NMR Spectroscopy User Guide*

Related `fiddle` Perform reference deconvolution

`fiddleu` Perform reference deconvolution subtracting successive FIDs (C)

Description Produces corrected differences between successive FIDs and the first FID. Refer to the description of `fiddle` for details.

See also *NMR Spectroscopy User Guide*

Related `fiddle` Perform reference deconvolution

`fiddle2d` Perform 2D reference deconvolution (C)

Description Functions the same as the `fiddle` program except `fiddle2d` performs 2D reference deconvolution. Refer to the description of `fiddle` for details.

See also *NMR Spectroscopy User Guide*

Related `fiddle` Perform reference deconvolution

`fiddle2D` Perform 2D reference deconvolution (C)

Description Functions the same as the `fiddle` program except `fiddle2D` performs 2D reference deconvolution. Refer to the description of `fiddle` for details.

See also *NMR Spectroscopy User Guide*

Related `fiddle` Perform reference deconvolution

`fiddle2Dd` 2D reference deconvolution subtracting alternate FIDs (C)

Description Functions the same as the `fiddle` program except `fiddle2Dd` performs 2D reference deconvolution. Refer to the description of `fiddle` for details.

See also *NMR Spectroscopy User Guide*

Related `fiddle` Perform reference deconvolution

fidmax Find the maximum point in an FID

Syntax `fidmax<(trace)>:$max`

Applicability VnmrJ 3.1

Description `fidmax` finds the absolute maximum value in an FID. With no arguments, `fidmax` uses the currently active FID, selected by `df` or `select`. A FID index may be supplied as an optional argument. For data collected using nf > 1, if `cf` is active, then the maximum of only that `cf` element will be returned. If the cf parameter is "`off`", then the maximum of all `cf` elements will be returned. Note that the maximum value returned by `fidmax` is divided by the value of '`ct`'.

Examples `fidmax:$max`

`fidmax(1):$max`

`fidmax(arraydim):$max`

fidpar Add parameters for FID display in the current experiment

Syntax `fidpar`

Applicability VnmrJ 3.1

Description All new parameter sets have the FID display parameters `dotflag`, `axisf`, `vpf`, `vpfi`, `crf`, and deltaf defined. Old parameter sets may not have these parameters defined. The macro `fidpar` is provided to create all these FID display parameters in the current experiment.

fidsave Save data (M)

Description Macro to save data. It uses `svfdir` and `svfname` to construct the data filename.

fifolpsize FIFO loop size (P)

Description Configuration parameter for the size of the FIFO loop. The size depends on which controller board is present on the system—the Output board, the Acquisition Controller board, or the Pulse Sequence Controller board (refer to the description of the `acquire` statement in the manual *User Programming* for information on identifying the boards). The value is set using the label Fifo Loop Size in the Spectrometer Configuration window (opened by `config`).

Values 2048

See also *VnmrJ Installation and Administration*

Related	
config	Display current configuration and possibly change it (M)

file File name of parameter set (P)

Description Contains the file name of the parameter set returned by a rt or rtp command. This parameter is reset when the go command is issued. If the system is not in automation mode (auto='n'), file is reset to the 'exp' value. If the system is in automation mode (auto='y'), file is set to the path of the directory where the data is stored.

See also *NMR Spectroscopy User Guide*

Related	
auto	Automation mode active (P)
go	Submit experiment to acquisition (C)
rt	Retrieve FID (C)
rtp	Retrieve parameters (C)

files Interactively handle files (C)

Syntax `files<(files_menu)>`

Description Brings up the interactive file handling program. With this program, the mouse and keyboard are used to copy, delete, rename, change directories, and load and save experiment data. The files command uses the graphics window to display file names. A mouse clicked on a file name selects it and the file name is displayed in reverse video. Various operations can be conducted on one or more selected files. The menus used for the files program are placed in the standard menulib directories. Refer to the manual *NMR Spectroscopy User Guide* for more information on using menus, and refer to the manual *User Programming* for information on programming menus.

Arguments files_menu is the files menu to control the menu buttons; the default menu is 'files_main' or the last active files menu.

Examples

```
files
files('files_dir')
```

See also *User Programming*

Related	
filesinfo	Return files display information (C)
tape	Control tape options of files program (P)

filesinfo Return file information for files display (C)

Syntax (1) `filesinfo('number'):$number_files`
(2) `filesinfo('name'<,file_number>):$file`
(3) `filesinfo('redisplay')`

Description Allows access to the list of files selected from the files interactive display. filesinfo is normally used only by the macros that implement the menu functions of the file system and not entered from the keyboard. The command will not execute unless the files program is active.

Arguments 'number' is a keyword to return the number of files selected in the files display, or 0 if no files have been selected.

$number_files is the return variable when 'number' is used.

'name' is a keyword to return a list of file names selected in the files display.

file_number is a number following the 'name' keyword to return only the file name in the list given by file_number.

$file is a string variable that returns the file name when 'name' is used.

'redisplay' is a keyword that causes the current contents of the directory to be displayed. This display is useful after making changes in the directory, such as deleting or creating a file.

See also *User Programming*

Related		
Related	files	Interactively handle files (C)

filtfile File of FIR digital filter coefficients (P)

Description Specifies name of a file of FIR (finite impulse response) digital filter coefficients. This file is a text file with one real filter coefficient per line (complex filters are not supported). If the parameter filtfile does not exist in the current experiment, enter addpar('downsamp') or addpar('oversamp') to add it. Entering addpar('downsamp') creates the digital filtering and downsampling parameters downsamp, dscoef, dsfb, dslsfrq, and filtfile. Similarly, entering addpar('oversamp') creates digital filtering and oversampling parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp.

Values File name. The file must be in the user's vnmrsys/filtlib directory.

Related	addpar	Add selected parameters to current experiment (M)
	def_osfilt	Default value of osfilt (P)
	downsamp	Downsampling factor applied after digital filtering (P)
	dscoef	Digital filter coefficients for downsampling (P)
	dsfb	Digital filter bandwidth for downsampling (P)

dslsfrq	Bandpass filter offset for downsampling (P)
oscoef	Digital filter coefficients for oversampling (P)
osfb	Digital filter bandwidth for oversampling (P)
osfilt	Oversampling filter for real-time DSP (P)
oslsfrq	Bandpass filter offset for oversampling (P)
oversamp	Oversampling factor for acquisition (P)
pards	Create additional parameters used for downsampling (M)
paros	Create additional parameters used for oversampling (M)

findxmlmenu Find an xml menu (M)

Description Find an xml menu. Used by the menu system to find and display VnmrJ menus.

fitspec Spectrum deconvolution

Syntax `fitspec<('option')>`

Applicability VnmrJ 3.1

Description Fit lorentzian and/or gaussian curves to experimental data. `fitspec` uses input from a text file "`fitspec.inpar`", which describes the starting values for a number of lines, which should be fitted to an experimental spectrum, and creates an output file "`fitspec.outpar`", which contains the fitted values for these lines. Furthermore, the resulting line frequencies are also stored in the parameter "`slfreq`", and the resulting amplitudes in "`sla`".

The files "fitspec.inpar" and "fitspec.outpar" contain the following information for each line:

frequency intensity line width gaussian fraction

A * after any of the numbers indicates, that that parameter should not be fitted.

The command `fitspec` in VNMR actually prepares a file "fitspec.indata", which contains the spectral data (as a text file), to which the data should be fit, and then executes the external program "`fitspec`", which is stored in "/vnmr/bin". This program uses as an input the files "fitspec.inpar" and "fitspec.indata", and produces after completion the output file "fitspec.outpar". This file is is then read by VNMR and uses to set "`slfreq`" and "sla".

fitspec('usell')

The file "fitspec.inpar" can be prepared from a line listing automatically with the command `fitspec('usell')`. This option of "fitspec" uses the information from the last line listing (stored in "`llfrq`" and "llamp"), and the parameters "slw", "vs", "rfl" and "rfp" to prepare that file. All lines are set to the same line width "slw" and the gaussian fraction is set to 0. If other starting values are required, this file should be edited.

fitspec('setslfreq')

If the output data from a spectrum deconvolution has to be used in a spin simulation, this can be done automatically, if first the spin system is defined and then the deconvolution is done, because `fitspec` saves it's results in "slfreq" and "slamp", which serve as input for the iterative spin simulation. If the spin system is defined after the deconvolution is complete, the contents of "slfreq" and "sla" is lost, but the result of the deconvolution is still available in "fitspec.outpar". In this case, the option "fitspec('setslfreq')" just copies the information from "fitspec.outpar" back into "slfreq" and "sla".

fixgrd — Convert gauss/cm value to DAC (M)

Syntax `fixgrd(gradient_value):parameter`

Description Uses the gcal value in the probe table to return the DAC value for a specified gradient strength.

Arguments `gradient_value` is the required gradient strength in gauss/cm.

`parameter` is any local variable or VnmrJ variable.

Examples `fixgrd(20):gzlvl`

Related

gcal	Gradient calibration constant (P)

fixpar — Correct parameter characteristics in experiment (M)

Applicability VnmrJ 3.1

Description After bringing parameters into the current experiment with convert, rt, rtp, or rtv, `fixpar` is automatically executed. `fixpar` updates old parameter characteristics and reconciles parameter differences due to the hardware on the spectrometer. If a macro userfixpar exists, `fixpar` runs it also. This allows an easy mechanism to customize parameter sets.

Related

convert	Convert data set from a VXR-style system (C)
fixpar3rf	Create parameters for third rf channel (M)
fixpar4rf	Create parameters for fourth rf channel (M)
parfix	Update parameter set (M)
parversion	Version of parameter set (P)
rt	Retrieve FIDs (C)
rtp	Retrieve parameters (C)
rtv	Retrieve individual parameters (C)

updatepars	Update all parameter sets saved in a directory (M)
userfixpar	Macro called by fixpar (M)

fixpar3rf Create parameters for third rf channel (M)

Applicability Systems with a second decoupler.

Description Checks for the existence of all acquisition parameters related to the second decoupler. Any parameters found to be absent are created, characterized, and initialized by the macro. fixpar3rf is run as a part of the standard fixpar macro if the system configuration parameter numrfch is greater than 2 (i.e., the number of rf channels on the system is set at 3 or more).

fixpar4rf Create parameters for fourth rf channel (M)

Applicability Systems with a third decoupler.

Description Checks for the existence of all acquisition parameters related to the third decoupler. Any parameters found to be absent are created, characterized, and initialized. fixpar4rf is run as a part of the standard fixpar macro if the system configuration parameter numrfch is greater than 3 (i.e., the number of rf channels on the system is set at 4).

fixpar5rf Create parameters for fifth rf channel (M)

Applicability Systems with a deuterium decoupler channel as the fourth decoupler.

Description Checks for the existence of all acquisition parameters related to the fourth decoupler. Any parameters found to be absent are created, characterized, and initialized. fixpar5rf is run as a part of the standard fixpar macro if the system configuration parameter numrfch is greater than 4 (i.e., the number of rf channels on the system is set at 5).

fixgrdR Converts Gradient Strength to DAC values

Description Converts a given DAC value to gradient strength based on the value of gcal in the probe file. This is the reverse of fixrgd.

Syntax fixgrdR(gradientstrength):$DAC_value

Description fixgrdR(3):$DAC_value

Related fixrgd

fixup — Adjust parameter values selected by setup macros (M)

Description Called by the experiment setup macros h1, c13, hc, hcapt, capt, and hcosy. As provided, the text of fixup is all in quotes so that it does nothing. It is intended to provide each user with a mechanism to make adjustments to values selected by the setup macros.

fixpsg — Update psg libraries (M)

Description Used by patchinstall to recompile the psg files and create new psg libraries libpsglib.so in /vnmr/lib.

flashc — Convert compressed 2D data to standard 2D format

Syntax `flashc<(<'nf'><,'ms'|'mi'|'rare'<,traces><,echoes>)>`

Applicability VnmrJ 3.1

Description Rearranges 2D "fid" data files from compressed formats to standard format or from standard format to compressed format. Compressed data is taken using the "nf" parameter to specify the number of fids in the second dimension of a 2D experiment. In other words compressed data is acquired as one large uninterrupted "multifid" acquisition.

Before the 6.0 release, arrayed or multislice compressed images (seqcon='nscnn'), had to be reformatted to a standard 2D format, using "flashc" before a "ft2d" could be performed on the data. Now using "ft2d('nf',<index>)" this is no longer necessary, and processing time may even be enhanced by reformatting data from the standard format (seqcon='ncsnn') to the compressed format. However for compressed compressed 2D (seqcon='nccnn'), "flashc(...)" or "flashc('nf',...)" must be run.

For 3D data sets "flashc" is not needed. The "ft3d" routine will handle standard, compressed (seqcon='nncsn'), or compressed-compressed (seqcon='nnccn') 3D data.

The flashc command reads the file "fid" in the "acqfil" subdirectory of the current experiment. The data is reordered and written back out to the same "fid" file. Thus, the original "fid" file is lost. Precautions are taken so that in the event of an error during processing, the original "fid" file will be preserved. Also, before running a simple check is done by flashc to prevent it from being executed more than once in an experiment on the same data set. The simple check against multiple executions of `flashc` looks for the parameter 'flash_converted' which `flashc` creates when it is run. To rerun `flashc` the parameter can be removed with the following commands:

```
destroy('flash_converted')
```

```
destroy('flash_converted','processed')
```

Compressed-compressed or Standard to Compressed Format

Using "`ft2d` ('nf',<index>)", `flashc` really only has to be used to convert a completely compressed multislice, multiecho, or multi-image sequence. However, for a large standard multi-slice experiment (seqcon='ncsnn') a performance benefit may be achieved in converting the data to a compressed format. When converting to a compressed format the first argument must always be '`nf`'. When converting completely compressed or "rare" type sequences, the first argument is a string defining the type of compression. This string can either be 'ms' for multislice, 'mi' for multi-image, or 'rare' for multi-echo "rare" type fast imaging data sets. The second argument defines the number of images slices or array elements to retain.

The values of four VNMR parameters are changed by `flashc`.

- "`ni`" is set to 1 if no argument is provided.
- "`nf`" is set to the value of "nf" divided by the multislice "ms" or multi-image "mi" value.
- "`arraydim`" is set to the product of its original value and the value of the "traces" argument.
- "arrayelemts" is set to 1 if no parameters were arrayed during data acquisition or to 2 if any parameter was arrayed during data acquisition.

Compressed to Standard Format

`flashc` can convert a completely compressed multislice, multiecho, or multi-image sequence. It can also convert a "rare" type sequence with a compressed phase-encode echo train. When converting completely compressed or "rare" type sequences, the first argument is a string defining the type of compression. This string can either be 'ms' for multislice, 'mi' for multi-image, or 'rare' for multi-echo "rare" type fast imaging data sets. The second argument defines the number of compressed traces to retain for each "`ni`" and "`nf`" will be set to this number after "flashc" has been run.

The values of four VNMR parameters are changed by `flashc`.

- "`nf`" is set to the value of the "traces" argument, or to 1 if no argument is provided.
- "`ni`" is set to the value of "nf" divided by the multislice "ms" or multi-image "mi" value.
- "`arraydim`" is set to the product of its original value and the original value of "nf".
- "`arrayelemts`" is set to 1 if no parameters were arrayed during data acquisition or to 2 if any parameter was arrayed during data acquisition.

Examples Compressed-compressed or Standard to Compressed Format

- flashc('nf'): Standard to compressed
- flashc('nf','ms',ns): Compressed phase-encode and multi-slice
- flashc('nf','mi',ns): Compressed multi-image and phase-encode

Compressed to Standard Format

- flashc: Simple compressed phase-encode
- flashc('ms',ns): Compressed phase-encode and multi-slice
- flashc('mi',ns): Compressed multi-image and phase-encode
- flashc('rare',ns,etl)

Related

arraydim	Dimension of experiment (P)
ft2d	Fourier transform 2D data (C)
ft3d	Fourier transform 3D data (C)
nf	Number of FIDs(P)
ni	Number of increments in 1st indirectly detected dimension(P)
seqcon	Acquisition loop control (P)

flipflop Set up parameters for FLIPFLOP pulse sequence (M)

Applicability Systems with solids module.

Description Sets up a multipulse parameter set for tuning out “phase glitch” in the probe and pulse amplifier.

See also *User Guide: Solid-State NMR*

Fluorine Set up parameters for 19F experiment (M)

Description Set Up parameters for ^{19}F experiment.

flush Write out data in VNMR memory

Applicability VnmrJ 3.1

Description The VNMR program keeps current data and parameters in memory buffers. Normally, this information is not written to disk until one exits VNMR or joins another experiment. Use this command to write out this information. One application is if you want to access the experimental data from a program separate from the VNMR program.

fn — Fourier number in directly detected dimension (P)

Description: Selects the Fourier number for the Fourier transformation along the directly detected dimension. This dimension is often referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc.

Values: `'n'` or a number equal to a power of 2 (minimum is 32). If `fn` is not *entered* exactly as a power of 2, it is automatically rounded to the nearest higher power of 2 (e.g., setting `fn=32000` gives `fn=32768`). `fn` can be less than, equal to, or greater than `np`, the number of directly detected data points:

- If `fn` is less than `np`, only `fn` points are transformed.
- If `fn` is greater than `np`, `fn` minus `np` zeros are added to the data table ("zero-filling").
- If `fn='n'`, `fn` is automatically set to the power of 2 greater than or equal to `np`.

fn1 — Fourier number in 1st indirectly detected dimension (P)

Description: Selects the Fourier number for the Fourier transformation along the first indirectly detected dimension. This dimension is often referred to as the f_1 dimension of a multi-dimensional data set. The number of increments along this dimension is controlled by the parameter `ni`.

Values: `fn1` is set in a manner analogous to the parameter `fn`, with `np` being substituted by `2*ni`.

See also: *NMR Spectroscopy User Guide*

Related:

`fn`	Fourier number in directly detected dimension (P)
`fn2`	Fourier number in 2nd indirectly detected dimension (P)
`ni`	Number of increments in 1st indirectly detected dimension (P)
`np`	Number of data points (P)

fn2 — Fourier number in 2nd indirectly detected dimension (P)

Description: Selects the Fourier number for the Fourier transformation along the second indirectly detected dimension. This dimension is often referred to as the f_2 dimension of a multidimensional data set. The number of increments along this dimension is controlled by the parameter `ni2`. `fn2` is set in a manner analogous to the parameter `fn`, with `np` being substituted by `2*ni2`.

See also: *NMR Spectroscopy User Guide*

Related:

`fn`	Fourier number in directly detected dimension (P)
`fn1`	Fourier number in 1st indirectly detected dimension (P)

ni2	Number of increments in 2nd indirectly detected dimension (P)
np	Number of data points (P)

fn2D Fourier number to build up 2D DOSY display in freq. domain (P)

Description In 2D DOSY sequences (Dbppste, DgcsteSL, Doneshot, Dbppsteinept), replaces fn when setting up the 2D display.

See also *NMR Spectroscopy User Guide*

Related

ddif	Synthesize and display DOSY plot (C)
dosy	Process DOSY experiments (M)

focus Send keyboard focus to input window (C)

Description Sends keyboard focus to the input window. This is only useful for macro programming.

See also *User Programming*

foldcc Fold INADEQUATE data about 2-quantum axis

Syntax `foldcc`

Applicability VnmrJ 3.1

Description `foldcc` symmetrizes 2D INADEQUATE data along the P-type double-quantum axis and applies an automatic DC baseline correction. The command `foldcc` functions for both hypercomplex and comlex 2D data.

foldj Fold J-resolved 2D spectrum about the F1=0 axis

Applicability VnmrJ 3.1

Description `foldj` symmetrizes heteronuclear 2D-J or rotated homonuclear 2D-J experiments about the F1=0 axis and functions with both complex and hypercomplex 2D data.

foldt Fold COSY-like spectrum along diagonal axis

Syntax `foldt(<sym_op>)`

Applicability VnmrJ 3.1

Description `foldt(<sym_op>)` folds COSY-like correlation spectra about the diagonal. The 2D spectrum must exhibit a P-type diagonal in order for `foldt` to work properly. [A P-type diagonal is one which goes from the bottom left-hand side to the top right-hand side of the contour display.] The argument `sym_op` can take three string values: 'symm', 'triang' and 'covar'. The default value is 'symm'.

Arguments If sym_op = 'symm', the folding process performs a symmetrization of the data by replacing every two symmetry-related points with the one point therein which is the smallest in magnitude. If sym_op = 'triang', the folding process performs a triangularization of the data by replacing every two symmetry-related points with their geometric mean.

If sym_op = 'covar', for "covariance NMR", the folding process answers the question of whether the two symmetry-related points are correlated. If the product of the two points (a and b) is greater than 0.0, the two points are each replaced with the sqrt(a*b). Otherwise, the two points are set to 0.0. The command foldt functions for both hypercomplex and complex 2D data but requires that `fn`=`fn1`.

fontselect Open FontSelect window (C)

Description Opens the FontSelect window for defining fonts in window panes created by `setgrid`. A different font can be selected for every window pane combination of rows and columns. Separate fonts can also be selected for a large or small overall graphic window.

See also *NMR Spectroscopy User Guide*

Related

`curwin`	Current window (P)
`jwin`	Activate current window (M)
`mapwin`	List of experiment numbers (P)
`setgrid`	Activate selected window (M)
`setwin`	Activate selected window (C)

format

Description Formats a real number into a nice string for output/converts a string into upper case or lower case for output/tests a string to determine if it can represent a real number/interconverts string representations of real numbers and real numbers

Syntax `Two arguments:`

```
format(stringvar,'upper'):stringvar
format(stringvar,'lower'):stringvar
format(stringvar,'isreal'):ans
```

Syntax `Three arguments:`

```
format(realvar,n,m):$sval
format(realvar,n,m):$rval
format(stringvar,n,m):$sval
format(stringvar,n,m):$rval
```

where realvar is a variable of real type. n is the length, m is the precision (number to the right of the decimal point. stringvar is a string variable. $sval is a string return value. $rval is a real return value.

Applicability VnmrJ 3.1

Description `format` can be used for the following:

- formats a real number into a nice string for output
- converts a string into upper case or lower case for output
- tests a string to determine if it can represent a real number
- interconverts string representations of real numbers and real numbers

Arguments If the command is given two arguments, the first argument may be a string or real variable and the action depends on the value of the second argument. If the second argument is 'upper', this command will convert the first argument to all upper case characters. If the second argument is 'lower', this command will convert the first argument to all lower case characters. If the second argument is 'isreal', this command will test the first argument to see if it satisfies the rules for a real number. It will return a 1 in the first argument can represent a real number and a 0 otherwise.

If the command is given three arguments, the first argument must be a real number or string holding a real number. If it is a string variable, it must satisfy the rules for a real number. The 'isreal' option above can be used for this purpose. This command will format it into either a string with length n and precision m or another real number of length n and precision m. If you want to return the value into a string, if it is a temporary dollar parameter (e.g., $sval), the parameter will need to be initialized as a string by first setting it to a string (e.g., $sval='').

Examples `format(a,5,2):sa` If a=24.1264 then string `sa`='24.13'
`format(solvent,'lower'):n1` If solvent='CDCl3' then n1='cdcl3'
`format($1,'isreal'):$a` Will set $a to 1 if $1 represents a number.
`$sval=''` "Initialize $sval to a string variable"`$snum = '143.92'`
`$rnum = 32.75`

Examples `Format real value $rnum = 32.75`
`format($rnum,3,1):$sval` Will set $sval to the string '32.8'
`format($rnum,3,1):$rval` Will set $rval to the number 32.8

Examples `Format string value $snum = '143.92'`

`format($snum,3,1):$sval`	Will set $sval to the string '143.9'
`format($snum,3,1):$rval`	Will set $rval to the number 143.9

fp Find peak heights or phases (C)

Syntax `fp<(<'phase',><index1,index2,...>)>`

Applicability VnmrJ 3.1

Description Following a line listing (either `dll` or `nll`), `fp` measures the peak height of each peak in an array of spectra. The results of the analysis are written to a text file `fp.out` in the current experiment directory. If the `npoint` parameter is defined in the current parameter set and this parameter is “on,” it determines the range of data points over which a maximum is searched when determining peak heights. The possible values of `npoint` are `1` to `fn`/4. The default is `2`.

Arguments `'phase'` is a keyword to measure the phase of each peak instead of height.

`index1,index2,...` restricts measuring peak heights or phases to the lines listed.

Examples

```
fp
fp(1,3)
fp('phase')
```

See also *NMR Spectroscopy User Guide*

Related

`dll`	Display listed line frequencies and intensities (C)
`fn`	Fourier number in directly detected dimension (P)
`getll`	Get line frequency and intensity from line list (C)
`nl`	Position cursor at the nearest line (C)
`nll`	Find line frequencies and intensities (C)
`npoint`	Number of points for `fp` peak search (P)

fpi Report integral values from arrayed spectra.

Syntax

```
fpi<('bc')>
fpi<('dc')>
fpi<('t1')>
```

Applicability VnmrJ 3.1

Description Following the definition of integral regions (either by hand, or using the region command), "`fpi`" measures the height of each integral in an array of spectra. If the keyword '`bc`' or '`dc`' is specified, one of commands is used to flatten the baseline or remove any baseline offset prior to evaluating the integrals. The results of the analysis are written into the text file "`fpi.out` " in the current experiment directory.

"`fpi`" always works on the entire spectrum, i.e., it will produce a report on all defined integral regions. "`fpi`" will indicate the integration limits in ppm units if "`axis='p'`" - if you prefer Hz units, set "`axis='h'`" prior to calling "`fpi`".

The resulting output, "`curexp+'/fpi.out'`" does NOT comply with the VNMR commands for T1 analysis etc. - however, if an argument '`t1`' is used, "`fpi`" and creates a file "`curexp+'/fp.out'`" which can be used for `T1, T2` etc. analysis (note that in this case the "line positions" marked in this file are mid-points of the respective integral region).

Arguments '`bc`' - optional baseline correction on each spectrum

'`dc`' - optional offset/drift correction on each spectrum

'`t1`' - optional creation of "`curexp+'/fp.out'`" which is compatible with "t1" and related commands

The '`bc`' and '`dc`' arguments cannot be combined.

Examples `fpi`

`axis='h' fpi axis='p'`

`fpi('dc')`

`fpi('bc','t1')`

`fpi('t1','dc')`

`fpi('t1')`

See also

Related `fp` Find peak heights or phases (C)

fpmult First point multiplier for np FID data (P)

Applicability VnmrJ 3.1

Description Allows error correction if the first point of an FID is misadjusted. In a 1D experiment, this adjustment influences the overall integral of the spectrum. For *n*-dimensional experiments, if the correction is not made, “ridges” can appear. In 2D experiments, the ridges appear as “f_2 ridges.” In 3D experiments, the ridges appear as “f_3 ridges.” These ridges can clearly be seen in the noise region on the top and bottom of a 2D spectrum (when `trace='f1'`) as a low-intensity profile of the diagonal. The sign and intensity of the ridges is controlled by the magnitude of `fpmult`.

It has been recognized that the first point of a FID that is sampled at exactly time equal to zero must be multiplied by 0.5 for the Fourier transform to function properly. The `fpmult` parameter gives you a method to fine-tune the actual correction factor.

Values Default is 1.0, except that if the processing involves backward extension of the time-domain data with linear prediction, the default changes to 0.5. If `fpmult` is set to `'n'`, `fpmult` takes on its default value.

See also *NMR Spectroscopy User Guide*

Related
fpmult1	First point multiplier for ni interferogram data (P)
fpmult2	First point multiplier for ni2 interferogram data (P)
np	Number of data points (P)
trace	Mode for *n*-dimensional data display (P)
wft2da	Weight and Fourier transform phase-sensitive data (M)

fpmult First point multiplier for "np" FID data

Applicability VnmrJ 3.1

Description For 2D experiments such as NOESY, TOCSY, or ROESY, one should run cfpmult on the transformed first increment, prior to typing wft2da, to minimize "F2 ridges" in the final 2D spectrum. This macro calculates an fpmult value for the dataset (which will then be used by wft2da).

One may do this manually for a 2D dataset by typing

fpmult=1.0 wft(1) cdc

in the VNMR command line and noting whether the spectrum (essentially the baseline) moves up or down when "dc" is typed. One should vary fpmult until the "dc" correction (jump in the baseline) is as small as possible. With care, one can set fpmult to two decimal places. Typical values for fpmult are 1.00-2.00. The default value for fpmult is 1.0.

This only needs to be performed for cosine-type experiments, such as NOESY, ROESY, or TOCSY where both the t2 FID and the t1 interferogram decay. The macro (cfpmult) might give incorrect values for first increments of experiments having baseline distortions (i.e. water suppression with 11-echo or 1331); in such cases manual optimization of fpmult is more suitable.

Why should you bother adjusting fpmult? If the first point in t1 of a 2D spectrum is misadjusted, the result will be the appearance of a series of "F2 ridges". These ridges can clearly be seen in the noise region on the top and bottom of a 2D spectrum (when trace = 'f1') as a low-intensity profile of the diagonal. The sign and intensity of the ridges is controlled by the magnitude of misset of fpmult.

It has been recognised that the first point of a FID which is sampled at exactly t2 = 0 must be multiplied by 0.5 for the Fourier Transform to function properly. The first point of a FID influences the overall integral of the resulting spectrum. The fpmult parameter gives one a way to fine-tune the actual correction factor.

NOTE: When processing 2D data, unless "lp" is approximately 0, FPMULT will affect both the DC offset and the curvature of the spectrum.

See also alfa and calfa

fpmult1 First point multiplier for "ni" interferogram data

Applicability VnmrJ 3.1

Description fpmult1 and fpmult2 operate on "ni" and "ni2" hypercomplex or complex interferogram data, respectively, in a manner analogous to fpmult. In many 2D and 3D experiments, the t1 (ni) and t2 (ni2) values are adjusted so that there is no first-order phasing in the F1 and F2 dimensions. In this case, fpmult1 and fpmult2 should be 0.5, the default value. If the t1 and t2 values are adjusted so that there is a 180-degree first-order phase correction, fpmult1 and fpmult2 should be 1.0.

fpmult2 First point multiplier for "ni2" interferogram data

Applicability VnmrJ 3.1

Description fpmult1 and fpmult2 operate on "ni" and "ni2" hypercomplex or complex interferogram data, respectively, in a manner analogous to fpmult. In many 2D and 3D experiments, the t1 (ni) and t2 (ni2) values are adjusted so that there is no first-order phasing in the F1 and F2 dimensions. In this case, fpmult1 and fpmult2 should be 0.5, the default value. If the t1 and t2 values are adjusted so that there is a 180-degree first-order phase correction, fpmult1 and fpmult2 should be 1.0.

fr Recall all display parameters from set #n, n=1..9 (n)

Syntax fr(n)

Applicability VnmrJ 3.1

Description fr(n) performs a full recall of the display parameter set, setting all current display parameters to those values.

Arguments A second argument can be given to these commands. It prevents them from causing the automatic update of interactive programs that may be displayed.

framecmd Create a new frame

Syntax

```
framecmd('new','image',x,y,width,height,'imagefilepath'):$id
framecmd('new','text',x,y,width,height,'textfilepath'<,
color,font,fontsize>):$id
framecmd('new','inset',x,y,width,height<,cr,delta<,cr1,
delta1>>):$id
framecmd('delete',$id)
```

framecmd('hide',$id)
framecmd('show',$id)

Applicability VnmrJ 3.1

Description `framecmd` will create a new frame of image, text, and inset with 'new' option. The type of image can be GIF, PNG, JPEG, or other image format supported by 'convert' program.

Arguments The range of x is 0 at the left edge of the chart and `wcmax` at the right edge of the chart.
The range of y is 0 at the bottom edge of the chart and `wc2max` at the top edge of the chart.
The range of width is 0 to `wcmax`.
The range of height is 0 to `wc2max`.
The color, font, and fontsize can be adjusted with text editor in VnmrJ window.

fread Read in variables from a file and load them in a tree

Syntax `fread(filename[,tree[,'reset', 'value', 'newonly']])`
`filename is a valid file with proper variable format.`
`tree can be current, global, processed, systemglobal, or usertree. 'reset' keyword can only be used if tree is specified.`

Applicability VnmrJ 3.1

Description This command reads in vnmr variables from a file and loads them into a tree. The variable trees are 'current', 'global', 'processed', 'systemglobal', and 'usertree'. It can read from any file that has variables stored in the correct vnmr format. The default tree is 'current'.

Arguments A "reset" option causes the variable tree to first be cleared before the new variable file is read. Without this option, variables read from a file are added to the existing preloaded variables. In order to use the 'reset' option, the tree must also be specified. A "value" option causes only the values of the variables in the file to be loaded. If a preloaded variable does not already exist, a new one is not created. Parameter attributes are not changed. Enumerated values are not changed. In order to use the 'value' option, the tree must also be specified.
A "newonly" option causes only those variables in the file which do not already exist in the tree to be loaded. In order to use the 'newonly' option, the tree must also be specified.
The 'reset', 'newonly', and 'value' options are mutually exclusive.
NOTE: if variables are read into the 'global' tree, certain parameters will not be loaded. These are important system parameters that should not be changed. These parameters are: userdir, systemdir, curexp, autodir, auto, operator, vnmraddr, and acqaddr.
The 'usertree' is available for use. By default, it has no parameters stored in it. It would typically be used by a macro for temporary parameter storage. All of the parameter utility commands, such as setlimit, setprotect, setvalue, getvalue, fsave, etc. will work with

'usertree' as the optional tree argument. A special incantation of fread with a empty string as the filename will clear parameters from 'usertree'. That is, fread('','usertree') clears 'usertree'.
Note that passing an empty string as the filename with other parameter trees generates an error. For example, fread('','current') is an error. As with all the parameter utility commands, the other arguments also work with 'usertree'. In the case of fread, this means that the 'reset', 'value', and 'newonly' options are valid for 'usertree'.

Examples `fread('var1')`: read in variables from file var1 into current tree.
`fread('sampvar','global')`: read in variables from file sampvar into global tree.

fsave Save parameters from a tree to a file (C)

Syntax `fsave(file<,tree>)`

Applicability VnmrJ 3.1

Description Writes parameters from a parameter tree to a file.

Arguments `file` is the name of the file, which can be any valid file for which the user has write permission. If the file already exists, it will be overwritten.

`tree` is one of the keywords `'global'`, `'current'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the `create` command for more information on types of trees.

Examples
```
fsave('var1')
fsave('sampvar','global')
```

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`destroy`	Destroy a parameter (C)
`display`	Display parameters and their attributes (C)
`fread`	Read parameters from file and load them into a tree (C)
`svp`	Save parameters from current experiment (C)

fsq Frequency-shifted quadrature detection (P)

Description Selects whether to use frequency-shifted quadrature detection. When `fsq` is turned on, if `dsp` is on, the observe frequency is offset by `oslsfrq`, and the digital filter is also offset by `oslsfrq`. The default value of `oslsfrq` is `1.25*sw`.

The effect of `fsq` is to offset only the digital filter by `oslsfrq`. The observe frequency must be offset by `oslsfrq` by modifying the pulse sequence as described in the manual *NMR Spectroscopy User Guide*.

Values `'n'` turns frequency-shifted quadrature detection off. `'y'` turns it on.

See also *NMR Spectroscopy User Guide*

Related		
	dsp	Type of DSP for data acquisition (P)
	oslsfrq	Bandpass filter offset for oversampling (P)
	oversamp	Oversampling factor for acquisition (P)
	sw	Spectral width in directly detected dimension (P)

ft Fourier transform 1D data (C)

Syntax (1) `ft<(<options,><'nf'><,start><,finish><,step>)>`
(2) `ft('inverse',exp_number,expansion_factor)`

Applicability VnmrJ 3.1

Description In syntax 1, performs a Fourier transform on one or more 1D FIDs without weighting applied to the FID. `ft` executes a left-shift, zero-order phase rotation, and a frequency shift (first-order phase rotation) according to the parameters lsfid, phfid, and lsfrq, respectively, on the time-domain data, prior to Fourier transformation. The type of Fourier transform to be performed is determined by the parameter proc. Solvent suppression is turned on or off with the parameters ssfilter and ssorder. For arrayed data sets, `ft` Fourier transforms all of the array elements. To Fourier transform selected array elements, `ft` can be passed numeric arguments.

In syntax 2, `ft` performs an inverse Fourier transform of the entire spectrum. (VnmrJ does not currently support inverse Fourier transformation of arrayed 1D or 2D data sets.)

Arguments `options` can be any of the following (all string arguments must precede the numeric arguments):

- `'acq'` is a keyword to check if any elements of a multi-FID experiment have already been transformed. If so, these previously transformed elements will not be retransformed.
- `'dodc'` is a keyword for all spectra to be dc corrected independently.
- `'nodc'` is a keyword to not perform the usual dc drift correction.
- `'nods'` is a keyword to prevent an automatic spectral display (ds) from occurring. This outcome is useful for various plotting macros.
- `'noft'` is a keyword to skip the Fourier transform, thereby allowing use of all spectral manipulation and plotting commands on FIDs.
- `'zero'` is a keyword to zero the imaginary channel of the FID prior to the Fourier transform. This zeroing occurs after any FID phasing. Its use is generally limited to wideline solids applications.

`'nf'` is a keyword that makes a single FID element containing nf traces to be transformed as if it were nf separate FID elements. If `'nf'` precedes the list of numeric arguments, the rules for interpreting the numeric arguments change slightly. Passing no numeric arguments results in the transformation of all nf traces in the first FID element. Passing a single numeric argument results in the transformation of all nf traces in the requested FID element (e.g., `ft('nf',3)` transforms all nf traces for element 3). Regardless of the requested FID element, the resulting spectra are labeled as 1 to nf because multiple elements cannot be transformed using ft('nf'). Subsequent numeric arguments are interpreted as previously described.

start is the index of a particular element to be transformed. For an array, start is the index of the first element to be transformed.

finish is the index of the last element to be transformed for an array.

step specifies the increment between successive elements that are to be transformed for an array. The default is 1.

'inverse' is a keyword specifying an inverse Fourier transform.

exp_number is the number of the experiment, from 1 to 9, for storing the resulting FID from the inverse Fourier transform.

expansion_factor defines the expansion of the spectrum before the inverse Fourier transform is performed. This argument is equivalent to a multiplier for the fn parameter. The multiplier is restricted to between 1 and 32 and is rounded up internally to the nearest power of 2.

Examples

```
ft
ft(1)
ft(3,7)
ft(2,10,2)
ft('nf',3)
```

See also *NMR Spectroscopy User Guide*

Related

dcrmv	Remove dc offsets from FIDs in special cases (P)
fn	Fourier number in directly detected dimension (P)
lsfid	Number of points to left-shift the np FID (P)
lsfrq	Frequency shift of the fn spectrum in Hz (P)
nf	Number of FIDs (P)
phfid	Zero-order phasing constant for np FID (P)
proc	Type of processing on the np FID (P)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
wft	Weight and Fourier transform 1D data (C)

wft — Weight and Fourier Transform 1D data

Syntax

Applicability VnmrJ 3.1

Description The commands wft and ft perform a Fourier transform on one or more 1D FID's with or without weighting applied to the FID, respectively. Both commands execute a left-shift, zero-order phase rotation, and a frequency shift (first-order phase rotation) according to the parameters "lsfid", "phfid", and "lsfrq" respectively, on the time-domain data prior to the weighting (if appropriate) and Fourier transformation. All string arguments supplied to these two commands must precede the numeric arguments, e.g., ft('nodc','noft',1,10,2). The type of Fourier transformation is determined by the parameter "proc". Solvent suppression is turned on or off with the parameters ssfilter and ssorder.

INVERSE FOURIER TRANSFORMATION:
ft('inverse', expnum, expansion_factor) performs an inverse FT, storing the resulting fid in the experiment defined by the second argument (first numeric argument). The expansion_factor defines the expansion of the spectrum before the inverse FT is performed. This argument is equivalent to a multiplier for the "fn" parameter, must lie between 1 and 32, and is rounded up internally to the nearest power of 2. Note that this command performs an inverse FT of the entire spectrum. Vnmr does not currently support the inverse FT of arrayed 1D or 2D data sets.

Arguments STRING ARGUMENTS:
ft('acq') checks if any elements of a multi-FID experiment have already been transformed. If so, these previously transformed elements will not be retransformed.
ft('nodc') does not perform the fid drift correction. ft('dodc') does perform the fid drift correction. The global parameter dc1d determines the default if neither 'nodc' nor 'dodc' is used. If the global parameter dc1d does not exists, or it exists and is set to 'y', then fid drift correction is performed. If the dc1d parameter exists and is set to 'n', fid drift correction is not performed. ft('nods') prevents an automatic spectral display (ds) from occurring. This is useful for various plotting macros. ft('noft') skips the actual ft step, thereby allowing to use all spectral manipulation and plotting commands on FID's. ft('zero') zeroes the imaginary channel of the FID prior to the Fourier transform. This zeroing occurs after any FID phasing. Its use will be generally limited to wideline solids applications. ft('nf') allows a single FID element containing 'nf' traces to be transformed as if it were 'nf' separate FID elements.

NUMERIC ARGUMENTS:
For arrayed data sets, both of these commands will Fourier transform all of the array elements. To Fourier transform selected elements of the array, wft and ft can be passed numeric arguments. Passing a single numeric argument will transform only that element. For example, wft(3) will transform only array element 3. Passing two numeric arguments will transform the inclusive array elements. For example, wft(3,7) will transform array elements 3, 4, 5, 6, and 7. Passing three numeric arguments is similar to passing two arguments with the addition that the third argument is used as the increment between successive elements that are to be transformed. For example, wft(2,10,2) will transform elements 2, 4, 6, 8, and 10. This use of numeric parameters is identical to the scheme used for displaying spectra with the dss command and other related commands. If the string argument 'nf' precedes the list of numeric arguments, the rules for interpreting the numeric arguments change slightly. Passing no numeric arguments results in the transformation of all 'nf' traces in

the first FID element. Passing a single numeric argument results in the transformation of all 'nf' traces in the requested FID element. For example, ft('nf',3) transforms all 'nf' traces for element 3. Regardless of the requested FID element, the resulting spectra are labeled as 1 to nf since multiple elements cannot be transformed using ft('nf'). Subsequent numeric arguments are interpreted as previously described.

Examples

See also

Related	proc	parameter
	ssfilter	parameter
	ssorder	parameter

ft1d — Fourier transform along f_2 dimension (C)

Syntax (1) `ft1d(element_number)`
(2) `ft1d<('nf',element_number)`
(3) `ft1d<(<options,><coefficients>)>`

Applicability VnmrJ 3.1

Description Performs the first Fourier transformation along the f_2 dimension, without weighting, and matrix transposition. `ft1d` allows the display of t_1 interferograms with the dcon and dconi commands. For arrayed 2D FID data, a single array element can be weighted and transformed using syntax 1 or 2. The keyword `'nf'` is used in syntax 2 to specify that the 2D data is collected in the compressed form using `'nf'`. Complex and hypercomplex interferograms can be constructed explicitly by supplying a series of options and coefficients using syntax 3.

For information on real as opposed to complex Fourier transforms, see the descriptions of the proc, proc1, and proc2 parameters. For information about Hadamard transforms, see the description of the proc1 parameter and the *VnmrJ NMR Liquids* user guide. For information on left-shifting, zero-order phase rotation, and frequency shifting of the FID and interferogram time-domain data during the 2D Fourier transformation, see the descriptions of the parameters lsfid, lsfid1, lsfid2, phfid, phfid1, phfid2, lsfrq, lsfrq1, and lsfrq2, as appropriate. For information on the lfs (low-frequency suppression) and zfs (zero-frequency suppression) solvent suppression options, see the description of the parameters ssfilter and ssorder, and the macro parfidss.

Arguments `element_number` is a single array element to be weighted and transformed.

`options` can be the keywords 'ptype' or 'ntype' but neither serve a useful function because the differential effect of these arguments is applied only during the course of the second Fourier transformation. The default is 'ntype'.

coefficients are a series of coefficients according to the following scheme: RR1 is the coefficient used to multiply the real part (first R) of spectra set 1 before it is added to the real part (second R) of the interferogram. IR2 would thus represent the contribution from the imaginary part of spectra set 2 to the real part of the interferogram, and so on. The scheme is depicted below.

```
ft1d(RR1,IR1,RR2,IR2,...,RI1,II1,RI2,II2,...)
```

where:

```
RR1*REAL(w2,element=1) ->  REAL(t1)
IR1*IMAG(w2,element=1) -> + REAL(t1)
RR2*REAL(w2,element=2) -> + REAL(t1)
IR2*IMAG(w2,element=2) -> + REAL(t1)
. . .
RI1*REAL(w2,element=1) ->  IMAG(t1)
II1*IMAG(w2,element=1) -> + IMAG(t1)
RI2*REAL(w2,element=2) -> + IMAG(t1)
II2*IMAG(w2,element=2) -> + IMAG(t1)
. . .
```

See also *NMR Spectroscopy User Guide*

Related

dconi	Interactive 2D data display (C)
ft2d	Fourier transform 2D data (C)
lsfid	Number of complex points to left-shift np FID (P)
lsfid1	Number of complex points to left-shift ni interferogram (P)
lsfid2	Number of complex points to left-shift ni2 interferogram (P)
lsfrq	Frequency shift of the fn spectrum (P)
lsfrq1	Frequency shift of the fn1 spectrum (P)
lsfrq2	Frequency shift of the fn2 spectrum (P)
parfidss	Create parameters for time-domain solvent subtraction (M)
phfid	Zero-order phasing constant for np FID (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
phfid2	Zero-order phasing constant for ni interferogram (P)
proc	Type of processing on np FID (P)
proc1	Type of processing on ni interferogram (P)
proc2	Type of processing on ni2 interferogram (P)
pmode	Processing mode for 2D data (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
wft2d	Weight and Fourier transform 2D data (C)

wft1d(coefficients) Weight and Fourier transform F2 of 2D data

Syntax

Applicability VnmrJ 3.1

Description `wft1d` and `ft1d` perform the first Fourier transformation along the F2 dimension, with and without weighting respectively, and matrix transposition. This allows the display of `t1` interferograms with the "`dcon`" and "`dconi`" commands.

ft1da Fourier transform phase-sensitive data (M)

Syntax `ft1da(<arg1> , <arg2>)`

Applicability VnmrJ 3.1

Description Performs the first (f_2) transform of a 2D transform or the first part of a 3D transform. Otherwise, `ft1da` has the same functionality as the `ft2da` command. See the description of `ft2da` for further information. For information about Hadamard transforms, see the description of the `proc1` parameter and the *VnmrJ NMR Liquids* user guide.

Arguments `options` are the same as used with `ft2da`. See `ft2da` for details.

See also *NMR Spectroscopy User Guide*

Related

`ft2d`	Fourier transform 2D data (C)
`ft2da`	Fourier transform phase-sensitive data (M)
`wft1da`	Weight and Fourier transform phase-sensitive data (M)
`wft2da`	Weight and Fourier transform phase-sensitive data (M)

ft1dac Combine arrayed 2D FID matrices (M)

Syntax `ft1dac<(<mult1><,mult2>,...<,multn>)>`

Applicability VnmrJ 3.1

Description Allows ready combination of 2D FID matrices within the framework of the 2D Fourier transformation program. No weighting is performed. `ft1dac` requires that the data be acquired either without f_1 quadrature or with f_1 quadrature using the TPPI method. This macro is used for TOCSY (with multiple mixing times).

Arguments `mult1,mult2,...,multn` are multiplicative coefficients. The `nth` argument is a real number and specifies the multiplicative coefficient for the `nth` 2D FID matrix.

Related

`ft2dac`	Combine arrayed 2D FID matrices (M)
`Tocsy`	Set up parameters for TOCSY pulse sequence (M)
`wft1da`	Weight and Fourier transform phase-sensitive data (M)
`wft1dac`	Combine arrayed 2D FID matrices (M)

ft1dac and wft1dac Help file for wft1dc macro used to combine arrayed 2D FID matrices

Syntax `wft1dac( <mult1> , <mult2> , <mult3> , ... )`

Applicability VnmrJ 3.1

Description This macro allows the ready combination of 2D FID matrices within the framework of the 2D-FT program.

Arguments The nth argument is a real number and specifies the multiplicative coefficient for the nth 2D FID matrix. It currently requires that the data be acquired either without F1 quadrature or with F1 quadrature using the TPPI method. WFT1DAC functions in an analogous manner.

Examples E.COSY and TOCSY (with multiple mixing times).

ft2d Fourier transform 2D data (C)

Syntax (1) `ft2d(array_element)`
(2) `ft2d('nf'<array_element>)`
(3) `ft2d<(<options,><plane_number,><coefficients>)>`
(4) `ft2d('ni'|'ni2',element_number,increment)`
(5) `ft2d('ni'|'ni2',increment,<coefficients>)`

Applicability VnmrJ 3.1

Description Performs the complete 2D Fourier transformation, without weighting, in both dimensions. If the first Fourier transformation has already been done using `ft1d`, `wft1d`, `ft1da`, or `wft1da`, the `ft2d` command performs only the second (t_1) transform.

For arrayed 2D FID data, a single array element can be weighted and transformed using syntax 1. If the data is collected in “compressed” form using `'nf'`, syntax 2 must be used. Complex and hypercomplex interferograms can be constructed explicitly by supplying a series of coefficients using syntax 3. If an arrayed 3D data set is to be selectively processed, the format of the arguments to `ft2d` changes to syntax 4. For example, `ft2d('ni',1,2)` performs a 2D transform along `np` and `ni` of the second `ni2` increment and the first element within the explicit array. This command yields a 2D `np`–`ni` frequency plane.

Arrayed 3D data sets can also be subjected to 2D processing to yield 2D absorptive spectra. If the States-Haberkorn method is used along both f_1 (`ni` dimension) and f_2 (`ni2` dimension), there are generally 4 spectra per (`ni`,`ni2`) 3D element. In this case, using syntax 5, entering `ft2d('ni2',2,<16 coefficients>)` performs a 2D transform along `np` and `ni2` of the second `ni` increment using the 16 coefficients to construct the 2D t_1-interferogram from appropriate combinations of the 4 spectra per (`ni`,`ni2`) 3D element.

If there are `n` data sets to be transformed, as in typical phase-sensitive experiments, `4*n` coefficients must be supplied. The first `2*n` coefficients are the contributions to the real part of the interferogram, alternating between absorptive and dispersive parts of the successive data sets. The next `2*n` coefficients are the contributions to the imaginary part of the interferogram, in the same order. Thus, using the definition that the first letter refers to the source data set, the second letter refers to the interferogram, and the number identifies the source data set, we have the following cases:

Data sets	*Coefficient order*
1	RR1, IR1, RI1, II1
2	RR1, IR1, RR2, IR2, RI1, II1, RI2, II2
3	RR1, IR1, RR2, IR2, RR3, IR3, RI1, II1, RI2, II2, RI3, II3
. . .	. . .

The coefficients are often `1`, `0`, or `-1`, but this is not always the case. Any non-integral coefficient can be used, and as many coefficients can be nonzero as is desired. Up to 32 coefficients can be supplied, which at 4 per data set allows the addition, subtraction, etc., of eight 2D data sets (e.g., 8 different phase cycles).

For information on real as opposed to complex Fourier transforms, see the descriptions of the `proc`, `proc1`, and `proc2` parameters. For information about Hadamard transforms, see the description of the `proc1` parameter and the *VnmrJ NMR Liquids* user guide. For information on left-shifting, zero-order phase rotation, and frequency shifting of the FID and interferogram time-domain data during the 2D Fourier transformation, see the descriptions of the parameters `lsfid`, `lsfid1`, `lsfid2`, `phfid`, `phfid1`, `phfid2`, `lsfrq`, `lsfrq1`, and `lsfrq2`, as appropriate. For information on the lfs (low-frequency suppression) and zfs (zero-frequency suppression) solvent suppression options, see the description of parameters `ssfilter` and `ssorder`, and macro `parfidss`.

Arguments `array_element` is a single array element to be transformed.

`options` can be any of the following (all string arguments must precede the numeric arguments):

- `'ptype'` is a keyword to transform P-type data to yield a P-type contour display.
- `'ntype'` is a keyword to transform N-type data to yield a P-type contour display. This is the default.
- `'t2dc'` is a keyword to apply a dc correction to each t_2 FID prior to the first Fourier transform. The last 1/16-th of the time domain data is used to calculate the dc level.
- `'t1dc'` is a keyword to apply a dc correction to each t_1 interferogram prior to the second Fourier transform. The last 1/16-th of the time domain data is used to calculate the dc level.

- 'f2sel' is a keyword to allow only preselected f_2 regions to be transformed along t_1. The t_1 interferograms in the non-selected f_2 regions are zeroed but *not* transformed. The same mechanism used to select baseline regions for baseline correction (bc) is used to select the f_2 regions to be transformed along t_1. Set intmod='partial' and partition the integral of the spectrum into several regions. The even numbered f_2 regions (e.g., 2, 4, 6) are transformed along t_1; the odd numbered regions are not transformed along t_1.
- 'nf' is a keyword to transform arrayed or multi-slice 2D data that has been collected in the compressed form as single 2D FIDs with multiple (nf) traces.
- 'ni2' is a keyword to transform non-arrayed 2D data that have been collected with ni2 and sw2 (instead of ni and sw1). addpar('3d') creates the necessary processing parameters for the 'ni2' operation.
- 'noop' is a keyword to not perform any operation on the FID data. This option is used mainly to allow macros, such as wft2da, to have the same flexibility as commands.

coefficients are a series of coefficients according to the following scheme: RR1 is the coefficient used to multiply the real part (first R) of spectra set 1 before it is added to the real part (second R) of the interferogram. IR2 would thus represent the contribution from the imaginary part of spectra set 2 to the real part of the interferogram, and so forth. The scheme is depicted below.

ft2d(RR1,IR1,RR2,IR2,...,RI1,II1,RI2,II2,...)

where:

```
RR1*REAL(w2,element=1) ->   REAL(t1)
IR1*IMAG(w2,element=1) -> + REAL(t1)
RR2*REAL(w2,element=2) -> + REAL(t1)
IR2*IMAG(w2,element=2) -> + REAL(t1)
. . .
RI1*REAL(w2,element=1) ->   IMAG(t1)
II1*IMAG(w2,element=1) -> + IMAG(t1)
RI2*REAL(w2,element=2) -> + IMAG(t1)
II2*IMAG(w2,element=2) -> + IMAG(t1)
```

'ni' is a keyword to selectively transform a particular np-ni 2D plane within a non-arrayed 3D data set. To identify the plane, 'ni' is followed by the plane_number argument, an integer from 1 through ni2.

'ni2' is a keyword to selectively transform a particular np-ni2 2D plane within a non-arrayed 3D data set. To identify the plane, 'ni2' is followed by the plane_number argument, an integer from 1 through ni.

element_number is the number of an element within the explicit array when selectively processing an arrayed 3D data set; it ranges from 1 to ni2

increment is the increment within the explicit array when selectively processing an arrayed 3D data set; it ranges 1 to arraydim/(ni*ni2).

Examples `ft2d(1,0,0,0,0,0,1,0)`
`ft2d(1)`
`ft2d('nf',3)`
`ft2d('ptype',...)`

See also *NMR Spectroscopy User Guide*

Related

dconi	Interactive 2D data display (C)
dcrmv	Remove dc offsets from FIDs in special cases (P)
fpmult	First point multiplier for np FID data (P)
fpmult1	First point multiplier for ni interferogram data (P)
ft1d	Fourier transform along f_2 dimension (C)
lsfid	Number of complex points to left-shift np FID (P)
lsfid1	Number of complex points to left-shift ni interferogram (P)
lsfid2	Number of complex points to left-shift ni2 interferogram (P)
lsfrq	Frequency shift of the fn spectrum (P)
lsfrq1	Frequency shift of the fn1 spectrum (P)
lsfrq2	Frequency shift of the fn2 spectrum (P)
parfidss	Create parameters for time-domain solvent subtraction (M)
phfid	Zero-order phasing constant for np FID (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
proc	Type of processing on np FID (P)
proc1	Type of processing on ni interferogram (P)
proc2	Type of processing on ni2 interferogram (P)
pmode	Processing mode for 2D data (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
wft1d	Weight and Fourier transform f_2 for 2D data (C)
wft2d	Weight and Fourier transform 2D data (C)

wft2d(coefficients) Weight and Fourier transform 2D data

Syntax

Applicability VnmrJ 3.1

wft2d and ft2d perform the complete 2D Fourier transformation, with and without weighting in both dimensions respectively. For arrayed 2D FID data, a single array element can be transformed using, as an example, "ft2d(array element number)". Complex and Hypercomplex interferograms can be constructed explicitly using the following coefficient table:
ft2d(rr1,ir1,rr2,ir2,...,ri1,ii1,ri2,ii2,...)

where
rr1 * REAL(w2, element=1) --> REAL(t1)
ir1 * IMAG(w2, element=1) --> + REAL(t1)
rr2 * REAL(w2, element=2) --> + REAL(t1)
ir2 * IMAG(w2, element=2) --> + REAL(t1)[etc.]

ri1 * REAL(w2, element=1) --> IMAG(t1)
ii1 * IMAG(w2, element=1) --> + IMAG(t1)
ri2 * REAL(w2, element=2) --> + IMAG(t1)
ii2 * IMAG(w2, element=2) --> + IMAG(t1)[etc.]

Arrayed hypercomplex data can be transformed by supplying the array index followed by the eight coefficients needed to construct the interferograms:

`ft2d`(array_element_number, rr1,ir1,rr2,ir2,ri1,ii1,ri2,ii2)

This is used in the special case where phase=1,2 and phase has the highest precedence in the array parameter, as for example, array='gzlvl1,phase'.

`ft2d`('ptype') will transform P-type data to yield a P-type contour display. `ft2d`('ntype') will transform N-type data to yield a P-type contour display. The same applies to `wft2d`. Although `ft1d` and `wft1d` will accept the string arguments 'ptype' and 'ntype', it serves no useful function in these two commands since the differential effect of these two arguments is applied only during the course of the second Fourier transformation.

`ft2d`('t2dc') causes a DC correction to be applied to each t2 FID prior to the first FT; `ft2d`('t1dc') causes a DC correction to be applied to each `t1` interferogram prior to the second FT. In both cases, the last 1/16-th of the time domain data is used to calculate the DC level.

`ft2d`('f2sel') allows only pre-selected F2 regions to be transformed along `t1`; the `t1` interferograms in the non-selected F2 regions are zeroed but NOT transformed. The same mechanism used to select baseline regions for baseline correction (`bc`) is used to select the F2 regions which are to be transformed along `t1`. Set intmod='partial' and partition the integral of the spectrum into several regions. The even numbered F2 regions, e.g., 2, 4, etc., will be transformed along `t1`; the odd numbered ones will not be transformed along `t1`.

`ft2d`('`nf`') transforms a non-arrayed 2D experiment which has been collected as a single 2D FID with multiple (`nf`) traces. In this example, each trace of the 2D FID corresponds to `t2` time domain data collected at an incremented value of t1.

`ft2d`('nods') and `wft2d`('nods') prevents the spectrum display following the transform.

The 'noft' option to `ft1d`, `wft1d`, `ft2d`, and `wft2d` prevents the actual Fourier transform step. `ft1d, ft2d` ('noft') will Fourier transform the `t2` time domain data but not the resulting `t1` interferograms. Both axes will be treated as frequency axes. `ft2d`('noft') will present the FID data, interpreted as if both axes are frequency axes. Other operations, such as weighting, solvent suppression, etc., will be performed as requested. Just the actual FT step is bypassed when this option is given.

ft2d ('noop') does not perform any operation on the FID data. It is used mainly to allow macros, e.g., wft2da, to have the same flexibility as actual VNMR commands.

ft2d ('ni2') transforms non-arrayed 2D data which have been collected with ni2 and sw2 (instead of ni and sw1). par3d creates the necessary processing parameters for the ft2d('ni2') operation. ft2d('ni',#) is used to selectively transform a particular "np-ni" 2D plane within a non-arrayed 3D data set; # is an integer which can range from 1 to ni2 in this example. ft2d('ni2',#) is used to selectively transform a particular "np-ni2" 2D plane within a non-arrayed 3D data set; # is an integer which can range from 1 to ni in this example. If an arrayed 3D data set is to be selectively processed, the format of the arguments to ft2d changes. For example, ft2d('ni',#1,#2) performs a 2D transform along np and ni of the #2-th ni2 increment and the #1-th element within the explicit array. This yields a 2D "np-ni" frequency plane. #1 ranges from 1 to ni2; and #2, from 1 to [arraydim/(ni*ni2)].

Arrayed 3D data sets can also be subjected to 2D processing to yield 2D absorptive spectra. If the States-Haberkorn method is used along both F1 (ni dimension) and F2 (ni2 dimension), there will generally be 4 spectra per (ni,ni2) 3D element. In this case, the command ft2d('ni2',#1, <16 coefficients>) would perform a 2D transform along np and ni2 of the #1-th ni increment using the ensuing 16 coefficients to construct the 2D t1-interferogram from appropriate combinations of the 4 spectra per (ni,ni2) 3D element.

See also For information on real vs. complex Fourier transforms, see the manual entry for "proc#". For information on left-shifting, zero-order phase rotation, and frequency shifting of FID and/or interferogram time-domain data during the 2D FT, see manual entries for "lsfid#", "phfid#", or "lsfrq#" respectively. For information on the lfs and zfs solvent suppression options, see manual entries for "ssfilter", "ssorder", and parfidss.

For information on Hadamard transforms, see the manual entries for "ht" and "proc1".

Related

phfid	parameter
lsfid	parameter
phfid1	parameter
lsfid1	parameter
phfid2	parameter
lsfid2	parameter
proc	parameter
proc1	parameter
proc2	parameter
pmode	parameter
ssorder	parameter
ssfilter	parameter
parfidss	command

ft2da Fourier transform phase-sensitive data (M)

Syntax ft2da(<arg1> , <arg2>)

Applicability VnmrJ 3.1

Description Processes 2D FID data and 2D planes at particular t_1 or t_2 times from a 3D data set for a pure absorptive display. ft2da differs from wft2da only in that, in the case of wft1da, weighting of the time-domain data is performed prior to the FT. ft2da functions analogously to ft1da and wft1da, except that ft2da and wft2da perform only the f_2 Fourier transform. For information about Hadamard transforms, see the description of the proc1 parameter and the *VnmrJ NMR Liquids* user guide.

Macros ft1da, wft1da, ft2da, and wft2da function for hypercomplex 2D FID data (phase=1,2) and for TPPI 2D FID data (phase=3 or phase=1,4) acquired either with ni or ni2. If the data were acquired with ni, no additional arguments need be used with the macros. If the data were acquired with ni2, the keyword 'ni2' must be used.

For phase=1,2:wft2da=wft2d('ptype',1,0,0,0,0,0,1,0)

For phase=3: wft2da=wft2d(1,0,0,0)

For phase=1,4:wft2da=wft2d('ptype',1,0,0,0,0,0,1,0)

Macros ft1da, wft1da, ft2da, and wft2da support selective 2D processing within a 3D FID data set. All permutations of hyercomplex and TPPI modes of data acquisition in t_1 and t_2 can be handled. For selective f_2f_3 processing, the numeric argument immediately following the 'ni2' keyword is interpreted to be the t_1 increment number, which specifies the particular f_2f_3 plane (plane_number, see below) to be processed. For selective f_1f_3 processing, the t_2 increment number either follows the keyword 'ni', which is optional, or is associated with the first numeric argument that does not immediately follow a 'bc' keyword.

For information on real as compared to complex Fourier transformation, see the description of proc or proc1. For information on the lfs (low-frequency suppression) and zfs (zero-frequency suppression) solvent suppression options, see the description of parameters ssfilter and ssorder, and the macro parfidss.

Arguments options can be any of the following (the order is not important):

- 'ntype', 't2dc', 't1dc', and 'f2sel' are keywords that function the same as when supplied to the ft2d and wft2d commands. Refer to the ft2d command for a description of these options.
- 'bc' is a keyword for a baseline correction of the phase-corrected f_2 spectra prior to the f_1 Fourier transform. The baseline regions must have been previously determined. A polynomial order of 1 (a spline fit) or a higher polynomial order must be specified by inserting a numerical argument following 'bc'.
- 'dc' is a keyword for a drift correction (dc) of the f_2 spectra prior to the f_1 Fourier transformation.

- 'ni' is a keyword to selectively transform a particular np-ni 2D plane within a non-arrayed 3D data set. To identify the plane, 'ni' is followed by plane_number, an integer from 1 through ni2.
- 'ni2' is a keyword to selectively transform a particular np-ni2 2D plane within a non-arrayed 3D data set. To identify the plane, 'ni2' is followed by plane_number, an integer from 1 through ni.
- 'old' is a keyword to allow data acquired before the February 25, 1988, software release to be processed correctly. 'old' does not function for selective 2D processing within 3D data sets. If no ni2 or ni plane_number is given, it is assumed that the data set is only 2D in either ni2 or ni, respectively.

See also *NMR Spectroscopy User Guide*

Related

f1coef	Coefficient to construct F1 interferogram (P)
f2coef	Coefficient to construct F2 interferogram (P)
ft1da	Fourier transform phase-sensitive data (M)
parfidss	Create parameters for time-domain solvent subtraction (M)
phase	Phase selection (P)
proc	Type of processing on the np FID (P)
proc1	Type of processing on the ni interferogram (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
wft1da	Weight and Fourier transform phase-sensitive data (M)
wft2da	Weight and Fourier transform phase-sensitive data (M)

ft2dac Combine arrayed 2D FID matrices (M)

Syntax `ft2dac<(<mult1><,mult2>,...<,multn>)>`

Applicability VnmrJ 3.1

Description Allows ready combination of 2D FID matrices within the framework of the 2D FT program. No weighting is performed. Data must be acquired either without f_1 quadrature or with f_1 quadrature using the TPPI method. ft2dac is used with TOCSY (with multiple mixing times).

Arguments mult1,mult2,...,multn are multiplicative coefficients. The nth argument is a real number and specifies the coefficient for the nth 2D FID matrix.

Related

ft1dac	Combine arrayed 2D FID matrices (M)
Tocsy	Set up parameters for a TOCSY pulse sequence (M)
wft1dac	Combine arrayed 2D FID matrices (M)
wft2dac	Combine arrayed 2D FID matrices (M)

ft2dac and wft2dac Help file for wft2dc macro used to combine arrayed 2D FID matrices

Syntax `wft2dac( <mult1> , <mult2> , <mult3> , ... )`

Applicability VnmrJ 3.1

Description This macro allows the ready combination of 2D FID matrices within the framework of the 2D-FT program. The nth argument is a real number and specifies the multiplicative coefficient for the nth 2D FID matrix. It currently requires that the data be acquired either without F1 quadrature or with F1 quadrature using the TPPI method. WFT2DAC functions in an analogous manner.

Examples E.COSY and TOCSY (with multiple mixing times)

ft3d Perform a 3D FT on a 3D FID data set

Syntax `ft3d(<>)`

Applicability VnmrJ 3.1

Description `ft3d` is a macro which executes the program ft3d in the VNMR system \`bin\` directory ($vnmrsystem/bin). The environmental parameter PATH specifies the list of directories through which UNIX searches until it finds an executable `ft3d` program.

Arguments The first string argument which is a non-keyword is 'datadir'. 'datadir' (without the /data subdirectory appended) is an optional argument which specifies the output directory for the 3D spectral data file(s). The default directory for the 3D spectral data is curexp/datadir3D. nfiles (an integer) is an optional argument which specifies the number of 3D data files (data1 to data\`nfiles\`) used to store the transformed 3D data. nfiles must be <= 32. If nfiles is entered, distributed F1F2 processing will be performed by the `ft3d` program if possible.

If the optional keyword 'nocoef' is submitted as an argument to the `ft3d` macro, VNMR will not create a 3D coefficient file prior to invoking the `ft3d` program. This is useful if one has modified an existing 3D coefficient file and does not want it to be overwritten prior to the 3D transform. By default, `ft3d` calls the `make3dcoef` macro to create a coefficient file using `f1coef` and `f2coef` string parameter values.

The 't1t2' and 't2t1' are optional arguments to explicitly define the order of `t1` and `t2` arrays (other than ni and ni2). By default the macro looks at array parameter to make a decision and in that case if any parameter other than phase and phase2 are arrayed the macro aborts.

The next set of optional keywords for `ft3d` pertain to plane extraction following the complete 3D FT. 'xall' indicates that all three 2D plane types, F1F3, F2F3, and F1F2, are to be automatically extracted at the end of the 3D FT. The output directory for the extracted 2D planes is the same as that for the 3D spectral data except that the former uses the /extr subdirectory whereas the latter uses the /data subdirectory. 'f1f3', 'f2f3', and 'f1f2' can be used to select any combination of plane types to be extracted. The ft3d macro allows the user to submit any of these keywords more than once. The program getplane, however, will display an error and abort if any one plane type is multiply defined for extraction.

The 3D FID data must be loaded into the experiment in which the `ft3d` macro is to be run. The `ft3d` program is started up in background mode by this macro so that VNMR remains free for interactive processing. In other words, one can start a 3D transform from within `exp4` and, at the same time, continue with any 1D or 2D processing of the 3D FID data within the same experiment using VNMR. If the 'fg' argument is given to `ft3d`, then the processing is done if foreground. No additional processing will be possible until the ft3d program has finished.

The optional 'noft' argument is similar to the 'noft' arguments to `ft2d` and `ft`. The Fourier transform step will be skipped in all three dimensions. In contradistinction to the 1D and 2D analogs, the 'noft' argument to `ft3d` causes all processing to be skipped; no weighting, phasing, etc. are performed. All axes will be treated as frequency axes.

Within the /data 3D data subdirectory, there are the following files and further subdirectories:

- data1 through data#: These are the actual binary 3D spectral data files. The number of data files depends upon the size of the largest 2D plane and the value for the UNIX environmental parameter `memsize` if nfiles is not entered.
- info: This is a directory which stores the 3D coefficient text file (`coef`), the binary information file (`procdat`), the 3D parameter set (`procpar3d`), and the automation file (auto). The first three files are created by the `set3dproc`() command within VNMR. The last file is created by the `ft3d` program.
- log: This is a directory which stores the log files produced by the `ft3d` program. `f3` contains all log output for the F3 transform. For the F2 and F1 transforms, there are two log file for each data file, one for the F2 transform (f2.#) and one for the F1 (f1.#). The master one for the F2 transform and one for the F1. The file contains all the log output produced by the master `ft3d` program.

The order of the arguments to the ft3d macro is not important.

Related	`set3dproc`	command
	`killft3d`	macro
	`getplane`	macro
	`make3dcoef`	macro
	`fiddc3d`	parameter
	`specdc3d`	parameter
	`ptspec3d`	parameter
	`ssfilter`	parameter
	`ssorder`	parameter

ntype3d	parameter
f1coef	parameter
f2coef	parameter

full Set display limits for a full screen (C)

Applicability VnmrJ 3.1

Description Sets the horizontal control parameters (sc and wc) and the vertical control parameters (sc2 and wc2) to produce a display (and subsequent plot) on the entire screen (and page). For 2D data, space is left for the scales.

Related

center	Set display limits for center of screen (C)
fullt	Set display limits for full screen with room for traces (C)
left	Set display limits for left half of screen (C)
right	Set display limits for right half of screen (C)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

fullsq Display largest square 2D display (M)

Description Adjusts sc, sc2, wc, and wc2 parameters to show the largest possible square 2D display.

Related

full	Set display limits for a full screen (C)
fullt	Set display limits for a full screen with room for traces (C)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

fullt Set display limits for a full screen with room for traces (C)

Applicability VnmrJ 3.1

Description Sets the horizontal control parameters (sc and wc) and the vertical control parameters (sc2 and wc2) to produce a display (and subsequent plot) in the entire screen (and page) with room for traces (dconi). For 2D data, space is left for the scales.

Related	`center`	Set display limits for center of screen (C)
	`full`	Set display limits for a full screen (C)
	`left`	Set display limits for left half of screen (C)
	`right`	Set display limits for right half of screen (C)

G

g2pul_ecc	Setup macro for eddy current compensation parameters (M)
ga	Submit experiment to acquisition and FT the result (M)
gain	Receiver gain (P)
gap	Find gap in the current spectrum (M)
gaussian	Set up unshifted Gaussian window function (M)
gcal_	Local value of the conversion factor between gradient in DAC points and gradient in G/cm (P)
gcal	Gradient calibration constant (P)
gcoil	Current gradient coil (P)
Gcosy	Convert the parameter to a gradient COSY experiment (M)
gdiff	Diffusion gradient level (P)
Gdqcosy	Convert the parameter to a gradient DQCOSY experiment (M)
get1d	Select a 1D experiment for processing (M)
get2d	Select a 2D experiment for processing (M)
getdim	Return dimensionality of experiment (M)
getemailaddr	Get email addresses from a file
geterror	Return or display an acquisition error
getfile	Get information about directories and files (C)
getgamma	Retrieves Gamma from /vnmr/nuctabref
getht	Retrieve/Save a Hadamard frequency list from a file
getlcdata	An LC-NMR communications macro
getlimit	Get the limits of a variable in a tree (C)
getll	Get intensity and line frequency of line (C)
getoffset	Sets offset based on current reference parameters
getparam	Retrieve parameter from probe file (M)
getplane	Extract planes from a 3D spectral data set (M)

getplottertype	Retrieves plotter information
getppm	Returns Cursor Value in ppm
getreg	Get frequency limits of a specified region (C)
getsampglobal	Loads sample global parameters
getshimmethods	Get proshim methods list (M)
getsn	Get signal-to-noise estimate of a spectrum (M)
gettoken	Utility macro to separate a string into tokens (M)
gettxt	Get text file from VnmrJ data file (C)
gettype	Get the type of a variable (C)
getvalue	Get value of parameter in a tree (C)
gf	Prepare parameters for FID/spectrum display in acqi (M)
gf	Gaussian function in directly detected dimension (P)
gf1	Gaussian function in 1st indirectly detected dimension (P)
gf2	Gaussian function in 2nd indirectly detected dimension (P)
gflow	Flow encoding gradient level (P)
gfs	Gaussian shift const. in directly detected dimension (P)
gfs1	Gaussian shift const. in 1st indirectly detected dimension (P)
gfs2	Gaussian shift const. in 2nd indirectly detected dimension (P)
Ghmbc	Convert the parameter to a gradient HMBC experiment (M)
ghmqc	Set up a PFG HMQC pulse sequence (M)
Ghmqc	Convert the parameter to a gradient HMQC experiment (M)
gHMQC15	Set up parameters for ^{15}N gHMQC experiment (M)
gHMQC_d2	Set up parameters for ^{15}N gHMQC experiment using dec. 2 (M)
gHMQC_d213	Set up parameters for ^{13}C gHMQC experiment using dec. 2 (M)
ghmqcps	Set up a PFG HMQC phase-sensitive pulse sequence (M)
ghsqc	Set up a PFG HSQC pulse sequence (M)
Ghsqc	Convert the parameter to a gradient HSQC experiment (M)
gHSQC15	Set up parameters for ^{15}N gHSQC experiment (M)
gHSQC_d2	Set up parameters for ^{15}N gHSQC experiment using dec. 2 (M)
gHSQC_d213	Set up parameters for ^{13}C gHSQC experiment using dec. 2 (M)
Ghsqctoxy	Convert parameters for gradient HSQCTOXY experiment (M)
gilson	Open the Gilson Liquid Handler window (C)

gilson	Allow starting the Gilson Liquid Handler GUI (M)
gin	Return current mouse position and button values (C)
globalauto	Automation directory name (P)
glue	Create a pseudo-2D dataset (M)
gmapshim	Start gradient autoshimming (M)
gmapshim_au	Start acquisition with gradient shimming (M)
gmapspin	Enable or disable spinning during gradient shimming (P)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
gmap_findtof	Gradient shimming flag to first find tof (P)
gmap_z1z4	Gradient shimming flag to first shim z1-z4 (P)
gmax	Maximum gradient strength (P)
gmqcosy	Set up PFG absolute-value MQF COSY parameter set (M)
gnoesy	Set up a PFG NOESY parameter set (M)
go_<pslabel>	Experiment-Specific Runtime Macro
go	Submit experiment to acquisition (M)
gradfit	Calculates fit coefficients describing the variation of gradient strength with position in calibration of non-uniform pulsed field gradients
go_	Pulse sequence setup macro called by go, ga, and au (M)
gpat-gpat3	Gradient shape (P)
gplan	Start interactive image planning (C)
gradientdisable	Disable PFG gradients (P)
gradientshaping	Activate shaping on the gradient pulses (P)
gradstepsz	Gradient step size (P)
gradtype	Gradients for X, Y, and Z axes (P)
graphis	Return the current graphics display status (C)
grayctr	Gray level window adjustment (P)
graysl	Gray level slope (contrast) adjustment (P)
grecovery	Eddy current testing (M)
grid	Draw a grid on a 2D display (M)
groupcopy	Copy parameters of group from one tree to another (C)
gspoil	Spoiler gradient level (P)

gsspat	Slice-select gradient shape (P)
gtnnoesy	Set up a PFG TNNOESY parameter set (M)
gtnroesy	Set up a PFG absolute-value ROESY parameter set (M)
gtotlimit	Gradient total limit (P)
gtrim	Trim gradient level (P)
gxmax,gymax,gzmax	Maximum gradient strength for each axis (P)
gzlvl	Pulsed field gradient strength (P)
gzsize	Number of z-axis shims used by gradient shimming (P)
gzwin	Spectral width percentage used for gradient shimming (P)

g2pul_ecc Setup macro for eddy current compensation parameters (M)

Applicability Systems with Agilent Cold Probes

Description Setup macro for pulse sequence used to determine the eddy current compensation parameters.

ga Submit experiment to acquisition and FT the result (M)

Syntax `ga<(<'nocheck'><,'next'><,'wait'>)>`

Description Performs experiment described by the current acquisition parameters, checking parameters `loc`, `spin`, `gain`, `wshim`, `load`, and `method` to determine the necessity to perform various actions in addition to simple data acquisition. This may involve a single FID or multiple FIDs, as in the case of arrays or 2D experiments. `ga` causes the data to be automatically weighted and Fourier transformed (`wft`) at the end of each FID data acquisition.

Before starting the experiment, `ga` executes two user-created macros if they exist. The first is `usergo`, a macro that allows the user to set up general conditions for the experiment. The second is a macro whose name is formed by `go_` followed by the name of the pulse sequence (from `seqfil`) to be used (e.g., `go_s2pul`, `go_dept`). The second macro allows a user to set up experiment conditions suited to a particular sequence.

Arguments `'nocheck'` is a keyword to override checking if there is insufficient free disk space for the complete 1D or 2D FID data set to be acquired.

`'next'` is a keyword to put the experiment started with `ga('next')` at the head of the queue of experiments to be submitted to acquisition.

'wait' is a keyword to stop submission of experiments to acquisition until wexp processing of the experiment, started with ga('wait'), is finished.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (M)
change	Submit a change sample experiment to acquisition (M)
gain	Receiver gain (P)
go	Submit experiment to acquisition (M)
go_	Pulse sequence setup macro called by go, ga, and au (M)
load	Load status of displayed shims (P)
loc	Location of sample in tray (P)
lock	Submit an Autolock experiment to acquisition (C)
method	Autoshim method (P)
sample	Submit change sample, Autoshim experiment to acquisition (M)
seqfil	Pulse sequence name (P)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
spin	Sample spin rate (P)
su	Submit a setup experiment to acquisition (M)
usergo	Experiment setup macro called by go, ga, and au (M)
wft	Weight and Fourier transform 1D data (C)
wshim	Conditions when shimming is performed (P)

gain Receiver gain (P)

Description Sets receiver gain or, by setting gain='n', enables Autogain for automatic adjustment of gain. Low gain in multiline, high-dynamic-range samples can cause a number of problems, including intermodulation distortions and extra lines in the spectrum. Too high a gain, on the other hand, can cause receiver overload and consequent baseline distortions. Autogain capability allows the observe channel to be set optimally for detecting and digitizing NMR signals from a wide variety of samples.

Autogain adjusts the observe channel gain such that the NMR signal takes about 50 percent of the maximum range of the ADC. This setting allows a comfortable leeway for variations in signal. The program begins acquisition in the normal manner but the first transient (after any requested steady state transients) is examined for signal level. If the intensity is too low or too high, the gain is changed and the process is repeated until the intensity is within the proper range, and then normal acquisition commences. The final gain value used for the experiment is stored and when the experiment is finished, setting gain='y' results in the value being displayed in the dgs parameter group.

If the gain is reduced by the Autogain procedure such that the noise does not trigger the least significant 1 or 2 bits in the ADC and the signal still overloads either the receiver or ADC, the system stops and displays a message indicating Autogain failure.

Values 0 to 60, in steps of 2 dB (60 represents highest possible receiver gain and 0 lowest). On 500-750-MHz systems, low-band gain is limited from 18 to 60.

`'n'` enables Autogain, in which the gain is automatically adjusted at the start of acquisition for an optimum value. After the acquisition is finished, setting `gain='y'` then allows the value of gain to be read. `gain='n'` may not be used for arrayed experiments.

See also *NMR Spectroscopy User Guide*

Related `dgs` Display group of special/automation parameters (M)
`gf` Prepare parameters for FID/spectrum display in `acqi` (M)

gap Find gap in the current spectrum (M)

Syntax `gap(gap,height):found,position,width`

Description Looks for a gap between the lines of the currently displayed spectrum. It can be used to automatically place inserts, parameter printouts, trace labels, etc. The search starts on the left side (low-field end) of the spectrum.

Arguments `gap` is the width of the desired gap.

`height` is the starting height (same as the lower limit for the insert).

`found` is a return value that is set to 1 if the search is successful, or set to 0 if unsuccessful.

`position` is a return value that is set to the distance from the left edge of the chart (not the plot) to the left end of the gap (3 mm from the nearest peak to the left, positioning with “left gravity”) if the search is successful, or set to the position (no spacing to the nearest line) of the largest gap found if unsuccessful.

`width` is a return value set to the total width of the first gap if the search is successful, or set to the width of largest gap found if unsuccessful.

Examples `gap(120,80);$1,$2,$3`

See also *User Programming*

gaussian Set up unshifted Gaussian window function (M)

Syntax `gaussian<(<t1_inc><,t2_inc>)>`

Description Sets up an unshifted Gaussian window function in 1, 2, or 3 dimensions. The macro checks whether the data is 1D, 2D, and 3D.

Arguments `t1_inc` is the number of t1 increments. The default is `ni`.
`t2_inc` is the number of t2 increments. The default is `ni2`.

See also *NMR Spectroscopy User Guide*

Related

`ni`	Number of increments in 1st indirectly detected dimension (P)
`ni2`	Number of increments in 2nd indirectly detected dimension (P)
`pi3ssbsq`	Set up pi/3 shifted sinebell-squared window function (M)
`pi4ssbsq`	Set up pi/4 shifted sinebell-squared window function (M)
`sqcosine`	Set up unshifted cosine-squared window function (M)
`sqsinebell`	Set up unshifted sinebell-squared window function (M)

gcal_ Local value of the conversion factor between gradient in DAC points and gradient in G/cm

Syntax `gcal_`

Applicability VnmrJ 3.1

Description `gcal_` is a local copy of the conversion factor from DAC points to G/cm for the probe used. `gcal_` is set equal either to the value in the current probe file, if available, or to the global value gcal, by the macro makedosyparams invoked when a DOSY pulse sequence is run, and does not normally need to be set manually.

See also gcal

gcal Gradient calibration constant (P)

Applicability Systems with the pulsed field gradient or the imaging module.

Description Stores the proportionality constant between the parameter values (DAC units) controlling the desired gradient and the intensity of the gradient expressed in gauss/cm. The gradients generated in the magnet require calibration of the gain on the gradient compensation board so that coordinate data, slice positions, and the field of view can be set up accurately. gcal should be located in each user's `vnmrsys/global` file.

Values Number that is probe dependent, in gauss/cm-DAC unit. On the Performa I PFG module, 0.00028 to 0.00055 gauss/cm-DAC unit is nominal; On the Performa II, 0.0014 to 0.0025 gauss/cm-DAC unit is nominal.

See also *VnmrJ Imaging NMR*

Related `setgcal` Set gradient calibration constant (M)

gcoil Current gradient coil (P)

Description Reserved parameter that specifies which physical gradient set is currently installed. This allows convenient updating of important gradient characteristics when one gradient set is interchanged for another. When set, gcoil reads the gradient table file of the same name in /vnmr/imaging/gradtables and sets the gradient calibration parameters.

gcoil is local to each individual experiment. It is normally set the same as sysgcoil for acquiring new data, but can be set to other gradient names when working with saved data or data from another instrument. Each possible gradient name should have an associated file of that name located in the directory /vnmr/imaging/gradtables. Look at any file in this directory for an example of the proper gradtable format, or use the macro creategtable to make new gradtables entries.

If the parameter gcoil does not exist in a parameter set and a user wants to create it, you must set the protection bit that causes the macro _gcoil to be executed when the value for gcoil is changed. There are two ways to create gcoil:

- Use the macro updtgcoil, which will create the gcoil parameter if it does not exist and set the correct protection bits.
- Enter the following commands:

```
create('gcoil','string')
setprotect('gcoil','set',9)
```

gcoil and the associated gradient calibration parameter gmax is updated with the values listed in the table on the right each time a parameter set is retrieved, or when an experiment is joined. In the rare case that a gradtables file is modified, but the value of gcoil is not changed, manually force an update of the calibration parameters. Updating may be accomplished either by setting gcoil to itself, for example, gcoil=gcoil, or by using the macro _gcoil.

Table 1:

Variable Name	*Value*
boresize	22.50 cm
gmax	5.00 gauss/cm

Be aware that if an old dataset is returned and processed, gradient parameters associated with that dataset will replace any new gcoil *parameters.*

The table is a gradient table (gradient coil name: asg33) for a horizontal imaging system with all three axes set to the same maximum gradient strength.

Table 2:

Variable Name	*Value*
boresize	5.10 cm
trise	0.000200 sec
gxmax	29.00 gauss/cm

On the right is a gradient table (gradient coil name: tc203) for a three-axis gradient set with unequal maximum gradient strength.

See also *User Programming*

Related	gmax	Maximum gradient strength (P)
	setgcoil	Assign sysgcoil configuration parameter (M)
	sysgcoil	System gradient coil (P)
	updtgcoil	Update gradient coil (M)

Gcosy Convert the parameter to a gradient COSY experiment (M)

Applicability Systems with the pulsed field gradient or the imaging module.

Description Converts a 1D standard two-pulse sequence parameter set into a set ready to run a PFG (pulsed field gradient) absolute-value COSY experiment.

See also *NMR Spectroscopy User Guide*

gdiff Diffusion gradient level (P)

Description Predefined parameter available for use in setting a diffusion gradient level, often paired with the timing parameters tdiff or tdelta.

Gdqcosy Convert the parameter to a gradient DQCOSY experiment (M)

Description Convert the parameter to a gradient Dqcosy experiment

get1d Select a 1D experiment for processing (M)

Syntax `get1d<(experiment)>`

Description In nonautomation mode, the macros hcosy, hcapt, capt, hcdept, and cdept all acquire two or more data sets in the experiment in which the macro was executed. These data sets are stored, complete with Fourier transformed data. The data sets are also stored directly in the experiment. The get1d macro is used to select which data set should be active for processing in that experiment. After get1d is executed, data can be stored in the conventional way with the svf command (e.g., when hcosy completes, get1d can be used to process the 1D data set).

Arguments `experiment` is the 1D data set to be used for processing. The default is the `'H1'` experiment.

Examples `get1d`
`get1d('apt')`

See also *NMR Spectroscopy User Guide*

Related

`capt`	Automated carbon and APT acquisition (M)
`cdept`	Automated carbon and DEPT acquisition (M)
`get2d`	Select a 2D experiment for processing (M)
`hcapt`	Automated proton, carbon, and APT acquisition (M)
`hcdept`	Automated proton, carbon, and DEPT acquisition (M)
`hcosy`	Automated proton and COSY acquisition (M)
`svf`	Save FIDs in current experiment (C)

get2d Select a 2D experiment for processing (M)

Syntax `get2d<(experiment)>`

Description In nonautomation mode, the macros `hcosy`, `hcapt`, `capt`, `hcdept`, and `cdept` all acquire two or more data sets in the experiment in which the macro was executed. These data sets are stored complete with Fourier transformed data. The data sets are also stored directly in the experiment. The `get2d` macro is used to select which data set should be active for processing in that experiment. After entering `get2d`, data may be stored in the conventional way with the `svf` command. For example, following completion of `hcosy`, `get2d` can be used to process the 2D data set.

Arguments `experiment` is the 2D data set that should be used for processing. The default is the `'relayh'` experiment.

Examples `get2d('hetcor')`

See also *NMR Spectroscopy User Guide*

Related

`get1d`	Select a 1D experiment for processing (M)
`svf`	Save FIDs in current experiment (C)

getdim Return dimensionality of experiment (M)

Syntax `getdim:dimensions`

Description Used in other macros to determine the number of dimensions of the current data set. Many macros make decisions based on whether a data set is multidimensional or 1D. `getdim` makes it easier to access this information.

Arguments `dimensions` is a return variable giving the number of dimensions of the data. If `ni3` is 2 or greater, `dimensions` is set to 4; if `ni2` is 2 or greater, `dimensions` is set to 3; if `ni` is 2 or greater, `dimensions` is set to 2; and if `ni` is less than 2 or undefined, `dimensions` is 1.

Examples `getdim:r1`

See also *NMR Spectroscopy User Guide*

Related	`ni`	Number of increments in 1st indirectly detected dimension (P)
	`ni2`	Number of increments in 2nd indirectly detected dimension (P)
	`ni3`	Number of increments in 3rd indirectly detected dimension (P)

getemailaddr Get email addresses from a file

Description For a given operator, this macro will use emails found in a comma separated "`.csv`" file or a space separated "`.txt`" file stored in the `/vnmr` directory.

The file must be named "emailaddress.csv" or "emailaddress.txt".

Syntax Email addresses in the space separated .csv file should appear as follows:

"krish krish@agilent.com

lydia lydia@agilent.com

dave dave@agilent.com"

Email addresses in the comma separated .txt file should appear as follows:

"krish,kris@agilent.com

lydia,lydia@agilent.com

dave,dave@agilent.com"

Applicability VnmrJ 3.1

geterror

Syntax `geterror:$str`
`geterror(errorNumber):$str`

Applicability VnmrJ 3.1

Description "`geterror`" will translate an error number into a descriptive string. With no argument, geterror will use acqstatus[2], which is the parameter that holds any acquisition related error. Alternatively, an error number may be supplied as an argument. If a return value is used, the error string is return to the calling macro. Otherwise, the error string is displayed.

Arguments The optional errorNumber is an integer representing an error.

Examples geterror:$res

Related `acq_errors` manual entry mapping error numbers to descriptive text.

getfile Get information about directories and files (C)

Syntax (1) `getfile(directory):$number_files`
(2) `getfile(directory,file_index):$file,$extension`

Description Returns information about the number of files in a directory or about a particular file in a directory.

Arguments `directory` is the name of the directory for which information is desired.

`number_files` is the number of files in the directory, with dot files (e.g., `.login`) ignored.

`file_index` is the number of file for which information is desired (the order is UNIX-dependent).

`file` is the name of the file, excluding any extension, identified by the `index` (see examples below).

`extension` is the extension of the file name identified by the `file_index`. For example, if `file_index` points to the file named `s2pul.fid`, `getfile` returns the string `s2pul` to `$file` and the string `fid` to `$extension`. If the file name pointed to has no extension (e.g., `dummy`), no value is returned to `$extension`. If the file name has more than one extension, only the last extension is returned to `$extension` (e.g., the file `fid.tmp.par` returns `fid.tmp` to `$file` and `par` to `$extension`).

Complete paths (full file names) can be reconstructed like this:

```
getfile('dir',i):$filename,$ext
if ($ext='') then $path='dir'+'/'+$filename
else $path='dir'+'/'+$filename+'.'+$ext
endif
```

Paths for the `rt` command can be reconstructed like this:

```
$path='dir'+'/'+$filename.
```

Examples

```
getfile('dir'):$entries
$temp = 0
while ($temp < $entries)
   $temp = $temp + 1
   getfile('dir',$temp):$filename,$ext
   ...
endwhile
```

See also *User Programming*

getgamma Retrieves Gamma from /vnmr/nuctabref

Description Retrieves value of gamma for a nucleus from /`vnmr/nuctabref`.

Syntax `getgamma('nucleus')`

See also `getgamma(tn)`

getht Retrieve/Save a Hadamard frequency list from a file.

Syntax
```
getht(<'htfrq1' <,'htbw1'>>)
getht(<'save' <,'htfrq1'>>)
```

Applicability VnmrJ 3.1

Description The `getht` macro is used to retrieve a Hadamard frequency line list from a file, and sets the Hadamard parameters in an experiment. It may also be used to save a Hadamard frequency line list from the current experiment into a file.

File format:

The format of the file is the same as the Line List display in the Edit HT Freq dialog. The first line is an optional title, specifying:

```
frequency [units]  bandwidth [units]
```

frequency units are Hz or ppm.

Units of Hz are measured from center of spectrum for Hadamard frequencies. The units label is set to [Hz from center]. Units of ppm are referenced to the current spectrum in the experiment. In a 1D, it is referenced to the direct acquisition dimension. In a 2D, it is referenced to F1. bandwidth units are assumed to be in Hz.

The second and subsequent lines are a list of frequencies and bandwidths. The bandwidth column is optional, and assumed to be 20 Hz (or the current value of htbw1) if not specified.

Arguments Usage for retrieving:

getht(<'htfrq1' <,'htbw1'>>)

If there is no first argument, `htfrq1` is used as the Hadamard frequency parameter. If there is no second argument, `htbw1` is used as the Hadamard bandwidth, if the bandwidth is specified in the file.

In this usage, the macro retrieves the Hadamard frequency line list from a file in the current workspace directory, and sets the parameter values. It also shows the parameters in the Line List display in the Edit HT Freq dialog (`editht` macro), if open. The file to be copied is in `curexp`, e.g.

```
/export/home/vnmr1/vnmrsys/exp1/htfrq1.ll
```

Usage for saving:

getht(<'save' <,'htfrq1'>>)

If the first argument is 'save', the Hadamard frequency list is copied from the Edit HT Freq line list display to the current workspace directory. If there is no second argument, htfrq1 is used as the Hadamard frequency parameter name. If a second argument is specified, it is used as the Hadamard frequency parameter name for the save file, e.g. getht('save','htfrq2') saves the file curexp + '/htfrq2.ll'.

Arguments htfrq1 - Hadamard frequency list in indirect dimension, in ppm or Hz from center of spectrum.

htbw1 - Hadamard bandwidth in indirect dimension, in Hz. It may be a single value or a list of values for each element in the htfrq1 list.

tn - nucleus used for frequency list.

Examples Example #1:

freq [Hz from center]
1172.37
327.69
-346.37
-1292.10

In Example #1, the Hadamard frequencies are in Hz from the center of the spectrum.

Example #2:

freq [ppm]	bw [Hz]
7.930	20
5.819	16
4.134	20
1.770	20

In Example #2, the Hadamard frequencies are in ppm, referenced to the current spectrum. The frequency bandwidth is set to 20 Hz for most of the frequencies, except for the second frequency, which is set to 16 Hz.

See also ht
HsqcHT
tocsyHT
editht
sethtfrq1
htfrqdisp
dll

getlcdata An LC-NMR communications macro

Applicability VnmrJ 3.1

Description This macro starts the LC data file listener (/vnmr/tcl/bin/fileListen) so that when the LC system sends a data file it is received and transferred to the appropriate experiment or automation directory. It is not necessary to use getlcdata in normal operation as the LC data file listener is automatically started when the start LC NMR run button is pressed. As described in the text above, `getlcdata` may be desirable for the transfer of the LC data after runs using the analyte collector where the original LC run and the NMR analysis are well separated in time.

`getlimit` get the limits of a variable in a tree (C)

Syntax `getlimit(name[,tree]):$max,$min,$step,$index`

Description `getlimit` displays or returns the limits of a variable in a tree.

The returned values are the max value, min. value, step size, and index. The fourth argument will return a 0 if the parameter is not using an indexed table lookup for the maximum, minimum, and step size. If the parameter is using the table lookup mechanism, the fourth argument will be set to the index for that table.

The variable trees are `current` (the default), `global`, `processed`, or `systemglobal`.

Arguments `name` — the name of the variable

`tree` — the variable tree: `current` (the default), `global`, `processed`, or `systemglobal`.

Examples `getlimit('np'):$max,$min,$step,$index`

sets `$max` to 128000, `$min` to 32, `$step` to 2 and `$index` to 0

`getlimit('lockfreq','systemglobal'):$max`

sets `$max` to 160

`getlimit('dpwr'):$max,$min,$step,$index`

sets `$max` to 49, `$min` to 0 `$step` to 1 and `$index` to 9

Related

`setlimit`	Set limits of a parameter in a tree (C)
`setprotect`	Set protection mode of a parameter (C)

`getll` Get intensity and line frequency of line (C)

Syntax `getll(line_number)<:height,frequency>`

Description Finds the height and frequency of line from a line listing. It assumes a previous line list using `dll`.

Arguments `line_number` is the number of the line in the line list.

`height` is the intensity of the specified line.

`frequency` is the line frequency with units defined by the parameter `axis`.

See also *User Programming*

Related

`axis`	Axis label for displays and plots (P)
`dll`	Display listed line frequencies and intensities (C)

fp	Find peak heights (C)
nll	Find line frequencies and intensities (C)

getoffset — Sets offset based on current reference parameters

Description Sets offset based on current reference parameters rather than output of setref macro. The input argument is Hz.

Syntax `getoffset('frequency')`

See also `getoffset(320)`

getparam — Retrieve parameter from probe file (M)

Syntax `getparam(param<,nucleus>):$value`

Description Retrieves the value of a parameter from the current probe file. The name of the probe file is referenced from the parameter probe.

Arguments `param` is the name of the parameter to be retrieved.

`nucleus` is the nucleus to be retrieved from the probe file. The default is the current value of the parameter tn

`value` is a return variable with the value of the retrieved parameter.

Examples
```
getparam('tpwr'):tpwr
getparam('dmf','H1'):$dmf
```

See also *NMR Spectroscopy User Guide*

Related

addnucleus	Add new nucleus to existing probe file (M)
addparams	Add parameter to current probe file (M)
addprobe	Create new probe directory and probe file (M)
probe	Probe type (P)
setparams	Write parameter to current probe file (M)
tn	Nucleus for the observe transmitter (P)
updateprobe	Update probe file (M)

getplane — Extract planes from a 3D spectral data set (M)

Syntax `getplane<(<data_dir><,plane_dir><,plane_type>)>`

Description Executes the program `getplane` in the VnmrJ system `bin` directory (`$vnmrsystem/bin`). `getplane` checks whether there is sufficient file space on the disk partition to accommodate the extracted planes. If space is insufficient, `getplane` writes an error to the VnmrJ text window and aborts. `getplane` does not delete the output plane directory if it is run multiple times to individually extract different plane types.

Arguments `data_dir` specifies the directory (without the `/data` subdirectory) containing the input 3D spectral data. The first non-keyword argument to `getplane` is always taken to be `data_dir`.

`plane_dir` specifies the directory (without the `/extr` subdirectory) in which the extracted planes are to be stored. The second non-keyword argument to `getplane` is always taken to be `plane_dir`. If `plane_dir` is not specified, `data_dir` also specifies the output plane directory. If both `data_dir` and `plane_dir` are not specified, the input data directory and the output plane directory are set to `curexp/datadir3d`. The parameter `plane` is always set equal to the output plane directory.

`plane_type` can be any of the following keywords:

- `'xall'` is a keyword to extract all three 2D plane types: f1f3, f2f3, f1f2.
- `'f1f3'`, `'f2f3'`, `'f1f2'` are keywords to extract their respective 2D planes.
- Any of these keywords can be submitted more than once to the `getplane` macro, but the `getplane` program displays an error and aborts if any one plane type is defined for extraction more than once.

Examples

```
getplane
getplane('data3d.inp,'data3d.planes','f1f3','f2f3')
```

See also *NMR Spectroscopy User Guide*

Related

`dplane`	Display a 3D plane (M)
`dproj`	Display a 3D plane projection (M)
`dsplanes`	Display a series of 3D planes (M)
`ft3d`	Perform a 3D Fourier transform (M)
`nextpl`	Display the next 3D plane (M)
`path3d`	Path to currently displayed 2D planes from a 3D data set (P)
`plane`	Currently displayed 3D plane type (P)
`plplanes`	Plot a series of 3D planes (M)
`prevpl`	Display the previous 3D plane (M)

getplottertype The getplottertype command retrieves plotter information.

Syntax

```
getplottertype:$rasterValue,$plotterType
getplottertype('plottername'):$rasterValue,$plotterType
getplottertype('plottername','osname'):$osname
```

Applicability VnmrJ 3.1

Description The getplottertype command retrieves plotter information. With zero or one argument, it will return the "raster" value from the devicetable file and the "Type" value from the devicenames file. With no arguments, it uses the value of the plotter parameter. The returned raster values are:

- 0 - Plotters which use the HPGL language.

- 1 - Plotters which use the PCL language and are in portrait mode.
- 2 - Plotters which use the PCL language and are in landscape mode.
- 3 - Plotters which use the PostScript language and are in portrait mode.
- 4 - Plotters which use the PostScript language and are in landscape mode.

Arguments The VnmrJ name for a plotter does not need to be the same name that the computer operating system (OS) uses for the plotter / printer. The `getplottertype` with two arguments, where the first argument is the VnmrJ plotter name and the second argument is the '`osname`' keyword, will return the plotter / printer name used by the OS.

getppm Returns Cursor Value in ppm

Description Returns the value of the current cursor position in `ppm`.

Syntax `getppm:$value`

Examples `getppm:r1`

getreg Get frequency limits of a specified region (C)

Syntax `getreg(region_number)<:minimum,maximum>`

Description Returns the frequency limits of a region. The spectrum should have been previously divided into regions with the `region` command.

Arguments `region_number` specifies the number of the region.

`minimum`,`maximum` are return values set to the frequency limits, in Hz, of the specified region.

Examples
```
getreg(1):$a,$b
getreg($4):cr,$lo
getreg(R1–1):r2,r3
```

See also *User Programming*

Related

`cz`	Clear integral reset points (C)
`ds`	Display a spectrum (C)
`numreg`	Return the number of regions in a spectrum (C)
`region`	Divide spectrum into regions (C)
`z`	Add integral reset point at cursor position (C)

getsampglobal Loads sample global parameters

Description Loads sample global parameters in the current workspace from the study directory.

See also getsampglobalt

Related getsampglobal, resetsampglobal, savesampglobal, mvsampglobal, showsampglobal

getshimmethods Get proshim methods list (M)

Applicability VnmrJ 3.2

Description Scan the proshimmethods and shimmethods directories in all active appdirs. Make a sorted list of all the methods. This is used by the VnmrJ interface to provide a selection mechanism for shim methods.

getsn Get signal-to-noise estimate of a spectrum (M)

Syntax getsn:current_sn,predicted_sn

Description Estimates spectrum signal-to-noise using the following algorithm:

- Measures four adjacent 5-percent portions at the left edge of the spectrum, finding the root-mean-square noise, and taking the smallest of the four values. By measuring four different values and finding root-mean- square noise instead of peak noise, the result should be reliable even if several signals are present in the selected regions.
- Next, estimates the signal level using the vertical scale adjustment macros: vsadjh for proton, vsadjc for carbon, and vsadj for other nuclei. For carbon spectra, this algorithm ignores solvent lines and TMS. For proton spectra, in addition to ignoring the largest line in the spectrum, if the tallest line is greater than three times the height of the second tallest line, the second highest line is be used instead. For other nuclei, getsn uses the tallest line in the spectrum.
- Finally, estimates the signal-to-noise at the end of the experiment by a simple extrapolation (multiplying by the square root of nt/ct).

Arguments current_sn is a return value set to the current signal-to-noise level.

predicted_sn is a return value set to the predicted signal-to-noise level at the end of the experiment.

See also *NMR Spectroscopy User Guide*

Related

ct	Completed transients (P)
nt	Number of transients (P)
testsn	Test signal-to-noise ratio (M)
vsadj	Adjust vertical scale (M)

vsadjc	Adjust vertical scale for carbon spectra (M)
vsadjh	Adjust vertical scale for proton spectra (M)

gettoken Utility macro to separate a string into tokens (M)

Syntax `gettoken(input_string<,delimiter>):output_string, next_location`

Description Gets the first occurrence of a substring in `input_string` which is delimited by delimiter, or by the default delimiter `'$'`. The substring is returned in `output_string`. The next location in the string after the second delimiter is returned as a real in `next_location`. If there are not both one occurrence of each of the beginning delimiter and the second delimiter - in other words, if the delimiters are not paired - an empty string is returned in `output_string`, and `-1` is returned in `next_location`. If the delimited substring is the last substring in input_string, then the substring is returned as expected, but `next_location` returns `-1`.

Arguments `input_string`

The string to be tokenized delimiter is the delimiter for the tokens (default is $)

Examples `gettoken($mydirname):$mytoken, $next_location`

`gettoken($mydirname,'%'):$mytoken, $next_location`

Related `reqpartest` Tests whether required parameters are set (M)

gettxt Get text file from VnmrJ data file (C)

Syntax `gettxt(file)`

Description Copies text from a data file to the current experiment.

Arguments `file` is the name of a VnmrJ data file saved from an experiment (i.e., a directory with a `.fid` or `.par` suffix). Do not include the file name suffix.

Examples `gettxt('/vnmr/fidlib/fid1d')`

See also *NMR Spectroscopy User Guide*

Related `puttxt` Put text file into another file (C)

gettype Get the type of a variable (C)

Syntax `gettype(name[, tree])<:index, name>`

Description Displays or returns the type of an existing variable.

Arguments A “string” variable can return type `'string'` or `‘flag’`. A “real” variable can return type `'real'`, `'delay'`, `‘frequency’`, `‘pulse’`, or `'integer’`. `gettype` returns one or two values to a macro. The first value is an integer corresponding to the parameter type. The second value is the name of the parameter type. `name` can be used in commands such as `settype` and `create`.

An optional `tree` argument can be given. Variables are `'current'`, `'global'`, `'processed'`, and `'systemglobal'`.

The default is to search for the parameter in the `'current'`, `'global'`, and `'systemglobal'` trees, in that order.

Examples `gettype('dmm'):$int,$name` sets `$int` to 4 and `$name` to `'flag'`.

See also `gettype('pw'):$int,$name` sets `$int` to 6 and `$name` to `'pulse'`.

getvalue Get value of parameter in a tree (C)

Syntax `getvalue(name [,index] [,tree])<:$val>`
`getvalue(name ,'size' [,tree])<:$num>`

Description Gets the value of any parameter in a tree. The value of most parameters can be accessed simply by using their name in an expression. For example, `sw?` or `r1=np` accesses the value of `sw` and `np`, respectively. However, parameters in the processed tree cannot be accessed that way; `getvalue` can be used to get the value of a parameter in the processed tree.

Single elements of an arrayed parameter can be retrieved by suppling an optional "index". "index" defaults to 1. If the second argument is the keyword 'size', then the number of elements of the parameter can retrieved. If the parameter does not exist, a zero (0) will be returned.

Arguments `parameter` is the name of an existing parameter.

`index` is the number of a single element in an arrayed parameter. Default is 1.

`tree` is one of the keywords `'global'`, `'current'`, `'processed'`, or `'systemglobal'`. The default is `'processed'`. Refer to the `create` command for more information on the types of parameter trees.

If the second argument is the keyword `'size'`, then the number of parameter elements can be retrieved. If the parameter does not exist, a zero (0) will be returned.

`tree`

Examples `getvalue('arraydim'):$val`

`getvalue('phase','size'):$num`

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`display`	Display parameters and their attributes (C)
`setgroup`	Set group of a parameter in a tree (C)

setlimit	Set limits of a parameter in a tree (C)
setprotect	Set protection mode of a parameter (C)
settype	Change type of a parameter (C)
setvalue	Set value of any parameter in a tree (C)

gf — Prepare parameters for FID/spectrum display in acqi (M)

Description Provided as a model for preparing parameters for the FID and spectrum display in acqi. The unmodified version of this macro turns off phase cycling, autoshimming, autolocking, spin control, temperature control, sample changer control, and autogain. It also selects the current pulse sequence and parameter set by issuing the command go('acqi') and the command acqi('par'). The automation parameters cp, wshim, alock, spin, temp, loc, and gain are then reset to their original values. Users can customize gf by copying it into their private maclib directory and editing that version to suit their needs.

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display process (C)
alock	Automatic lock status (P)
cp	Cycle phase (P)
dmgf	Absolute-value display of FID data and spectrum in acqi (P)
gain	Receiver gain (P)
go	Submit an experiment to acquisition (C)
loc	Location of sample in tray (P)
spin	Sample spin rate (P)
temp	Sample temperature (P)
wshim	Conditions when shimming performed (P)

gf — Gaussian function in directly detected dimension (P)

Description Defines a Gaussian time constant of the form exp(-(t/gf)2) along the directly detected dimension. This dimension is referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc.

Values Number, in seconds. Typical value is gf='n'.

See also *NMR Spectroscopy User Guide*

Related

gf1	Gaussian function in 1st indirectly detected dimension (P)
gf2	Gaussian function in 2nd indirectly detected dimension (P)
gfs	Gaussian shift constant in directly detected dimension (P)

gf1 Gaussian function in 1st indirectly detected dimension (P)

Description Defines a Gaussian time constant of the form `exp(-(t/gf1)2)` along the first indirectly detected dimension. This dimension is referred to as the f_1 dimension of a multidimensional data set. `gf1` works analogously to the parameter `gf`. The “conventional” parameters, such as `lb` and `gf`, operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

Values Number, in seconds.

See also *NMR Spectroscopy User Guide*

Related `gf` Gaussian function in directly detected dimension (P)

gf2 Gaussian function in 2nd indirectly detected dimension (P)

Description Defines a Gaussian time constant of the form `exp(-(t/gf2)2)` along the second indirectly detected dimension. This dimension is referred to as the f_2 dimension of a multidimensional data set. `gf2` works analogously to the parameter `gf`. The `wti` program can be used to set `gf2` on the 2D interferogram data.

Values Number, in seconds.

See also *NMR Spectroscopy User Guide*

Related `gf` Gaussian function in directly detected dimension (P)

`wti` Interactive weighting (C)

gflow Flow encoding gradient level (P)

Description Predefined parameter available for use in setting a flow encoding gradient level, often paired with the timing parameter `tflow`.

See also *VnmrJ Imaging NMR*

gfs Gaussian shift const. in directly detected dimension (P)

Description Working in combination with the `gf` parameter, `gfs` allows shifting the center of the Gaussian function `exp(-((t-gfs)/gf)2)` along the directly detected dimension. This dimension is referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc. Typical value is `gfs='n'`.

See also *NMR Spectroscopy User Guide*

Related
gf Gaussian function in directly detected dimension (P)
gfs1 Gaussian shift const. in 1st indirectly detected dimension (P)
gfs2 Gaussian shift const. in 2nd indirectly detected dimension (P)

gfs1 Gaussian shift const. in 1st indirectly detected dimension (P)

Description Working in combination with the gf1 parameter, gfs1 allows shifting the center of the Gaussian function exp(-((t-gfs1)/gf1)2) along the first indirectly detected dimension. This dimension is referred to as the f_1 dimension in multidimensional data sets. gfs1 works analogously to the parameter gfs. The “conventional” parameters (i.e., lb, gf, etc.) operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

See also *NMR Spectroscopy User Guide*

Related
gf Gaussian function in directly detected dimension (P)
gf1 Gaussian function in 1st indirectly detected dimension (P)
gfs Gaussian shift const. in directly detected dimension (P)

gfs2 Gaussian shift const. in 2nd indirectly detected dimension (P)

Description Working in combination with the gf2 parameter, gfs2 allows shifting the center of the Gaussian function exp(-((t-gfs2)/gf2)2) along the second indirectly detected dimension. This dimension is referred to as the f_2 dimension in multidimensional data sets. gfs2 works analogously to the parameter gfs. The wti program can be used to set gfs2 on the 2D interferogram data.

See also *NMR Spectroscopy User Guide*

Related
gf Gaussian function in directly detected dimension (P)
gf2 Gaussian function in 2nd indirectly detected dimension (P)
gfs Gaussian shift const. in directly detected dimension (P)
wti Interactive weighting (C)

Ghmbc Convert the parameter to a gradient HMBC experiment (M)

Applicability Systems with a pulsed field gradient module.

Description Prepares an experiment for a PFG (pulsed field gradient) HMQC.
Arguments *NMR Spectroscopy User Guide*

ghmqc Set up a PFG HMQC pulse sequence (M)

Applicability Systems with a pulsed field gradient module.
Description Prepares an experiment for a PFG (pulsed field gradient) HMQC using the sequence GHMQC. The sequence sets three gradients, all separately.
Arguments *NMR Spectroscopy User Guide*

Ghmqc Convert the parameter to a gradient HMQC experiment (M)

Description Convert the parameter to a gradient HMQC experiment

gHMQC15 Set up parameters for ^{15}N gHMQC experiment (M)

Description Converts the current parameter set to a gHMQC experiment for ^{15}N.

gHMQC_d2 Set up parameters for ^{15}N gHMQC experiment using dec. 2 (M)

Description Converts the current parameter set to a gHMQC experiment for ^{15}N with decoupler 2 as ^{15}N.

gHMQC_d213 Set up parameters for ^{13}C gHMQC experiment using dec. 2 (M)

Description Converts the current parameter set to a gHMQC experiment for ^{13}C with decoupler 2 as ^{13}C.

ghmqcps Set up a PFG HMQC phase-sensitive pulse sequence (M)

Applicability Systems with a pulsed field gradient module.

Description Prepares an experiment for a PFG (pulsed field gradient) HMQC, phase-sensitive version.

See also *NMR Spectroscopy User Guide*

ghsqc Set up a PFG HSQC pulse sequence (M)

Applicability Systems with a pulsed field gradient module.

Syntax `ghsqc<(nucleus)>`

Description Converts a 1D standard two-pulse sequence parameter set into a parameter set ready to run a PFG (pulsed field gradient) HSQC experiment, either absolute value or phase sensitive.

Arguments `nucleus` is 13C or 15N. The default is 13C.

See also *NMR Spectroscopy User Guide*

Ghsqc Convert the parameter to a gradient HSQC experiment (M)

Description Convert the parameter to a gradient HSQC experiment.

gHSQC15 Set up parameters for ^{15}N gHSQC experiment (M)

Description Converts the current parameter set to a gHSQC experiment for ^{15}N.

gHSQC_d2 Set up parameters for ^{15}N gHSQC experiment using dec. 2 (M)

Description Converts the current parameter set to a gHSQC experiment for ^{15}N with decoupler 2 as ^{15}N.

gHSQC_d213 Set up parameters for ^{13}C gHSQC experiment using dec. 2 (M)

Description Converts the current parameter set to a gHSQC experiment for ^{13}C with decoupler 2 as ^{13}C.

Ghsqctoxy Convert parameters for gradient HSQCTOXY experiment (M)

Description Convert the parameter to a gradient HSQCTOXY experiment

gilson Open the Gilson Liquid Handler window (C)

Syntax `gilson`

Description Opens the Gilson Liquid Handler window, which enables setup, configuration, and operation of the VAST automatic sampler changer accessory.

See also *NMR Spectroscopy User Guide*

gilson Allow starting the Gilson Liquid Handler GUI

Applicability VnmrJ 3.1

Description When the "`gilson`" macro is invoked, a window appears on the screen and users then can select appropriate item in it to run the Gilson Liquid Handler. If an argument is passed to gilson, for example, `gilson`(1), then the `gilson` window appears and allows users to edit the details of inserting and removing samples with the Gilson. However, direct communication with the Gilson sample changer is not available.

gin Return current mouse position and button values (C)

Applicability All

Syntax `gin<(Bn_<press><release>)>:$x,$y,$b1,$b2,$b3`

Description The `gin` command reports the pointer position in relationship to the graphics window and is often used with the move and draw commands. The variables `$x` and `$y` are the x and y positions hold the pointer in millimeters. The variables `$b1`, `$b2`, and `$b3` hold the values for the state of the left, middle, and right mouse buttons.

Values `$x` is the value in the *x* direction, in millimeters, of the pointer. The range of `x` is 0 at the left edge of the chart and `wcmax` at the right edge. A value of -1 is returned if the pointer position is outside the graphics window along the x axis.

`$y` is the position of the pointer along the y axis. The range of y is -20 at the bottom of the chart to `wc2max` at the top. A value of 10000 is returned if the pointer position is outside the graphics window along the y axis.

$b1 is the state of left button; returns the value 0 if released and 1 if pressed.

$b2 is the of middle button; returns the value 0 if released and 1 if pressed.

$b3 is the of right button; returns the value 0 if released and 1 if pressed.

Arguments: no argument, returns current mouse positions and button values.

Bn_press, n=a,1,2, or 3. Wait for mouse button (any, 1, 2, or 3) or any key to be pressed.

Bn_release, n=a,1,2, or 3. Wait for mouse button (any, 1, 2, or 3) to be released or any key to be pressed.

Examples: gin('B3_press'):$x,$y,$b1,$b2,$b3
wait until button 3or any key is pressed

gin('Ba_press'):$x,$y,$b1,$b2,$b3
wait until any button or any key is pressed

gin('B1_release'):$x,$y,$b1,$b2,$b3
wait until button 1 is released or any key pressed

gin('B2_release'):$x,$y,$b1,$b2,$b3
wait until button 2 is released or any key pressed

See also: *User Programming*

Related		
	box	Draw a box on a plotter or graphics display (C)
	draw	Draw line from current location to another location (C)
	move	Move to an absolute location to start a line (C)

globalauto — Automation directory name (P)

Applicability: *VnmrJ Walkup* and systems with automation such as sample handling.

Description: A global parameter that specifies the name of a directory in which the daily automation directories or study directories are saved. This parameter is created and used by the walkup macro and the VnmrJ Walkup interface.

See also: *NMR Spectroscopy User Guide; VnmrJ Walkup*

Related		
	cqinit	Initialize liquids study queue (M)
	walkup	Walkup automation (M)

glue — Create a pseudo-2D dataset (M)

Applicability: Systems with the LC-NMR accessory.

Syntax: glue<(num_scans)>

Description: Steps through the series of FIDs, putting them into exp5 one by one as an array, and then jumps to exp5 and changes the parameters arraydim, ni, and fn1, so that the data appear to the user to be a 2D experiment, which can then be processed and displayed with standard 2D commands (wft2d, dconi, etc.). The parameter savefile should exist and should contain the base file name to which a series of FIDs have been saved as savefile.001, savefile.002, etc.

Arguments num_scans is the number of FIDs copied into the exp5 array. Typically, num_scans is used if the experiment was aborted prematurely, so that the complete num_scans worth of FIDs were not actually acquired.

See also *NMR Spectroscopy User Guide*

Related savefile Base file name for saving FIDs or data sets (P)

Applicability VnmrJ 3.1

gmapshim Start gradient autoshimming (M)

Applicability Systems with gradient shimming installed.

Syntax gmapshim<('files'|'mapname'|'quit')>

Description Starts gradient autoshimming if no arguments are used. It can also retrieve a shimmap file or quit gradient autoshimming. When the gmapshim macro is done, it automatically exits, and the previous data set is retrieved.

Arguments 'files' is a keyword to enter the gradient autoshimming files menu.

'mapname' is a keyword to display the current mapname.

'quit' is a keyword to exit from gradient autoshimming and retrieve the previous data set.

See also *NMR Spectroscopy User Guide*

Related gmapsys Run gradient autoshimming, set parameters, map shims (M)

gmapz Get parameters and files for gmapz pulse sequence (M)

gmapshim_au Start acquisition with gradient shimming (M)

Applicability Systems with gradient shimming installed.

Description If wshim is not set to 'n', gmapshim_au checks the probe file for a lock gradient map name. If the name exists, gmapshim_au executes gmapshim('glideau') to start gradient shimming followed by acquisition. If the map name does not exist, gmapshim_au starts acquisition by running au('wait').

gmapspin Enable or disable spinning during gradient shimming (P)

Description
Specifies whether or not sample spinning during gradient shimming is enabled. If spinning is enabled during gradient shimming, the pulses and delays *must* also be synchronized with the rotor period.

Values
'n' disable spinning during gradient shimming.
'y' enable spinning during gradient shimming.

Related

gmapz	Get parameters and files for gmapz pulse sequence (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gzsize	Number of z-axis shims used by gradient shimming (P)
spin	Sample spin rate (P)

gmapsys Run gradient autoshimming, set parameters, map shims (M)

Applicability
Systems with gradient shimming installed.

Syntax
(1) `gmapsys<(option)>`
(2) `gmapsys('shimmap'<,shimmap_option>)`

Description
Enters the Gradient Shimming Setup panel for setting parameters, mapping the shims, and performing autoshimming. This is the only entry point to the gradient shimming Setup panel.

If the gmapz pulse sequence is not loaded, retrieve parameters from the last shimmap used (or current mapname) or from `gmapz.par` if no shimmap exists.

Arguments
`option` is one of the following keywords:

- `'addpar'` adds gradient shimming parameters to the current parameter set.
- `'findgzlvl'` runs an experiment to calibrate `gzlvl`, `gzwin`, and `tof` to optimize the spectral window.
- `'findgzwin'` runs an experiment to calibrate `gzwin` and `tof` to optimize the spectral window.
- `'findtof'` runs an experiment to center `tof` to optimize the spectral window.
- `'rec'` displays the record of shim adjustments from the previous gradient shimming run.
- `'shim'` start autoshimming (same as Gradient Autoshim on Z button).
- `'vi'` edits the file `gshim.list`, which is used for editing shim offsets, `mapname`, or selecting coarse and fine shims.
- `'writeb0'` displays the `b0` plot calculated from the first two array elements.

'shimmap' is a keyword to run a shim mapping experiment and save the results (same as Make Shimmap button).

shimmap_option is one of the following values:

- 'auto' is a keyword to calibrate gzwin and then make a shimmap (same as Automake Shimmap button).
- 'manual' is a keyword to use shim offset values set manually from the file gshim.list and not the default values to make a shimmap.
- 'overwrite' is a keyword to make a shimmap and overwrite the current mapname if it exists.
- mapname is the prefix of the shimmap file name. The default is the user is queried for mapname before running the experiment.

See also *NMR Spectroscopy User Guide*

Related

gmapshim	Start gradient autoshimming (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
gradtype	Gradients for X, Y, Z axes (P)
gzwin	Spectral width percentage used for gradient shimming (P)
seqfil	Pulse sequence name (P)
gmap_z1z4	Gradient shimming flag to first shim z1-z4 (P)
gzsize	Number of z-axis shims used by gradient shimming (P)

gmapz Get parameters and files for gmapz pulse sequence (M)

Applicability Systems with gradient shimming installed.

Syntax gmapz<(mapname)>

Description Retrieves gradient shimming parameters to set up a gradient shimming experiment.

Arguments mapname is the name of a gradient shimmap file that must exist in the shimmaps directory. gmapz retrieves parameters and loads the shimmap file from mapname. The default is to retrieve standard gradient shimming parameters from the file gmapz.par.

See also *NMR Spectroscopy User Guide*

Related

gmapshim	Start gradient autoshimming (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmap_z1z4	Gradient shimming flag to first shim z1-z4 (P)

gmap_findtof Gradient shimming flag to first find tof (P)

Applicability Systems with gradient shimming installed.

Description When the flag is set to 'y', gradient shimming first performs a calibration to find tof before the start of shimming. This action is recommended for only homospoil deuterium gradient shimming with different solvents. The default value is 'n'.

Values 'y' turns on the flag.
'n' turns off the flag.

See also *NMR Spectroscopy User Guide*

Related

gmapshim	Start gradient autoshimming (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
tof	Frequency offset for observe transmitter (P)

gmap_z1z4 Gradient shimming flag to first shim z1-z4 (P)

Applicability Systems with gradient shimming installed.

Description When the flag is set to 'y', if gzsize is greater than 4, gradient shimming first shims on z1-z4, and then uses all shims specified by gzsize. When the flag is set to 'n'(default), all shims specified by gzsize are used.

Values 'y' turns on the flag.
'n' turns off the flag.

See also *NMR Spectroscopy User Guide*

Related

gmapshim	Start gradient autoshimming (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
gzsize	Number of z-axis shims used by gradient shimming (P)

gmax Maximum gradient strength (P)

Description The allowed maximum gradient level (absolute value) in gauss/cm. gmax is one of the calibration entries in a gradtables file. gxmax, gymax, and gzmax are used when the maximum gradient level is different for each axis in gauss/cm, which is the case for triple-axis PFG coils.

See also *VnmrJ Installation and Administration*; *VnmrJ Imaging NMR*

Related

gcoil	Current gradient coil (P)
gxmax,gymax,gzmax	Maximum gradient strength for each axis (P)
sysgcoil	System gradient coil (P)

gmqcosy Set up PFG absolute-value MQF COSY parameter set (M)

Applicability Systems with the pulsed field gradient module.

Description Converts a 1D standard two-pulse sequence parameter set into a parameter set ready to run a PFG (pulsed field gradient) absolute-value MQF COSY experiment.

See also *NMR Spectroscopy User Guide*

gnoesy Set up a PFG NOESY parameter set (M)

Applicability Systems with the pulsed field gradient module.

Description Converts a 1D standard two-pulse sequence parameter set into a parameter set ready to run a PFG (pulsed field gradient) NOESY experiment, either absolute value or phase sensitive.

See also *NMR Spectroscopy User Guide*

go_<pslabel>Experiment-Specific Runtime Macro

See also The go_<pslabel> macro, if it exists, is executed when acquisition begins on a pslabel-specific basis.

go Submit experiment to acquisition (M)

Syntax go<(<'acqi'><,'nocheck'><,'nosafe'><,'next'>
<,'sync'><,'wait'>)>

Description Performs the experiment described by the current acquisition parameters, checking parameters loc, spin, gain, wshim, load, and method to determine the necessity to perform various actions in addition to data acquisition. This may involve a single FID or multiple FIDs, as in the case of arrays or 2D experiments. go acquires the FID and performs no processing. If free disk space is insufficient for the complete 1D or 2D FID data set to be acquired, go prompts the user with an appropriate message and aborts the acquisition initiation process.

Before starting the experiment, go executes two user-created macros if they exist. The first is usergo, a macro that allows the user to set up general conditions for the experiment. The second is a macro whose name is formed by go_ followed by the name of the pulse sequence (from seqfil) to be used (e.g., go_s2pul, go_dept). The second macro allows a user to set up experiment conditions suited to a particular sequence.

Arguments 'acqi' is a keyword to submit an experiment for display by the acqi program. All operations explained above are performed, except acquisition of data is not initiated. The instructions to control data acquisition are stored so that acqi can acquire the data when the FID button is clicked. The gf macro is recommended instead of running go('acqi') directly. Using gf prevents certain acquisition events from occurring, such as spin control and temperature change. See the description of gf for more information.

'nocheck' is a keyword to override checking if there is not enough free disk space for the complete 1D or 2D FID data set to be acquired.

'nosafe' is a keyword to disable probe protection during the experiment.

'next' is a keyword to put the experiment started with go('next') at the head of the queue of experiments to be submitted to the acquisition system. If go('next') is entered, the go macro remains active until the experiment is submitted to the acquisition system, and no other VnmrJ commands are processed until the go macro finishes.

'sync' is a keyword in nonautomation mode that accomplishes the same effect as go('next') in synchronizing VnmrJ command execution with the submission of experiments to the acquisition system. The difference is that 'sync' does not put the experiment at the head of the queue.

'wait' is a keyword to stop submission of experiments to acquisition until wexp processing of the experiment, started with go('wait'), is finished.

Examples
```
go
go('nosafe')
go('next')
```

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display process (C)
au	Submit experiment to acquisition and process data
change	Submit a change sample experiment to acquisition (M)
gain	Receiver gain (P)
ga	Submit experiment to acquisition and FT the result (C)
gf	Prepare parameters for FID/spectrum display in acqi (M)
go_	Pulse sequence setup macro called by go, ga, and au (M)
load	Load status of displayed shims (P)
loc	Location of sample in tray (P)
lock	Submit an Autolock experiment to acquisition (C)
method	Autoshim method (P)
probe_protection	Probe protection control (P)

sample	Submit change sample, Autoshim exp. to acquisition (M)
seqfil	Pulse sequence name (P)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
spin	Sample spin rate (P)
su	Submit a setup experiment to acquisition (M)
usergo	Experiment setup macro called by go, ga, and au (M)
vnmrjcmd()	Commands to invoke the GUI popup (C)
wshim	Conditions when shimming is performed (P)

go_ — Pulse sequence setup macro called by go, ga, and au (M)

Syntax go_*macro*

Description Called by the macros go, ga, or au before starting an experiment. The user typically creates this macro to set up general experiment conditions. The name of the macro is formed by combining go_ with the name of the pulse sequence macro (from seqfil) to be used.

Examples
```
go_dept
go_noesy
go_s2pul
```

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (M)
ga	Submit experiment to acquisition and FT the result (M)
go	Submit experiment to acquisition (M)
seqfil	Pulse sequence name (P)
usergo	Experimental setup macro called by go, ga, and au (M)

gpat-gpat3 — Gradient shape (P)

Description Predefined string parameters available to specify gradient shapes.

See also *VnmrJ Imaging NMR*

gplan — Start interactive image planning (C)

Syntax gplan(function_name, arg1, arg2,...)

Description In VnmrJ, starts an image planning session.

Arguments `'function_name',path` is the name of an image planning function surrounded by single quotation marks.

`arg1, arg2,...` are arguments for the function, if relevant.

Examples `gplan 'clearStacks()'`
`get 'PrevStacks()'`

See also *NMR Spectroscopy User Guide*

x Multiplier for gradient pulses on alternating scans (P)

Syntax `create('gradalt','real')`

Applicability VnmrJ 3.2

Description The `zgradpulse` and `rgradient` pulse elements use the value of `gradalt` to mutliply the gradient amplitude.

No changes are made if:

- the local (`curpar`) parameter `gradalt` does not exist
- the local (`curpar`) parameter `gradalt` is set to "Not Used"
- the local (`curpar`) parameter `gradalt` is set to "1"

See also *User Programming Guide*

gradfit calculates fit coefficients describing the variation of gradient strength with position in calibration of non-uniform pulsed field gradients

Syntax `gradfit(lowfrq,highfrq,D)`

gradfit(lowfrq,highfrq,D,ncoef)

Applicability VnmrJ 3.1

Description `gradfit` calculates the coefficients of a power series to fit the measured variation of gradient strength with position during the calibration of non-uniform pulsed field gradients.

Arguments `gradfit` takes 3 or 4 arguments: lowfrq is the lower frequency limit of the signal profile, highfrq the high frequency limit, D the diffusion coefficient of the calibrant, and ncoef is the number of coefficients in the power series (default is 8).

Examples

See also `nugcalib`
`nugcal`
`powerfit`

`gradientdisable` Disable PFG gradients (P)

Description `gradientdisable` is an optional global parameter for disabling the gradient pulses. If `gradientdisable` parameter is set to `'y'`, the psg software sets the gradient dac values to 0. The gradient parameters in VnmrJ and pulse sequence are not altered. This feature works in both C psg and SpinCAD Jpsg.

To use this feature, create `gradientdisable` as a global parameter of type `'flag'`. If gradientdisable is set to `'y'`, the gradient amplitude values will be set to 0; if set to `'n'` the gradient amplitudes will be the expected values determined by the gradient parameters and pulse sequence calculations. This feature is typically used in experiments involving Cold Probes. This feature is only effective for gradient configurations, `gradtypes` of `'l'`, `'p'`, and `'t'`.

Related
`pfgon` Pulsed field gradient amplifiers on/off control (P)
`gradtype` Gradients for X, Y, and Z axes (P)

`gradientshaping` Activate shaping on the gradient pulses (P)

Applicability Systems with Varian, Inc. Cold Probes

Description Activate shaping on the gradient pulses in the pulse sequence without changing the pulse sequence source program. This feature works only the Z gradient pulses, specified using the `zgradpulse(..)` PSG statement. gradientshaping is a global parameter.

Values `gradientshaping='y'` enables this feature and produces a WURST shaping of gradient amplitudes.
`gradientshaping='n'` or destroy the parameter disables this feature and produces rectangular gradients amplitudes.

`gradstepsz` Gradient step size (P)

Description The maximum gradient DAC value. `gradstepsz` determines the type of gradient DAC board used in the system: 12-bit or 16-bit. It is used internally to convert gauss/cm gradient levels to the proper hardware DAC level.

Values Systems with 12-bit DACs (older SISCO spectrometers without gradient waveform capabilities): –2047 to +2047 units, in integer steps.

Systems with 16-bit DACs (SISCO spectrometers with gradient waveform capabilities): –32767 to +32767 units, in integer steps.

See also *VnmrJ Installation and Administration*; *VnmrJ Imaging NMR*

gradtype — Gradients for X, Y, and Z axes (P)

Applicability: Systems with pulsed field gradient (PFG) or imaging capability.

Description: Configuration parameter for systems with optional gradients for axes. The value is set using the label X Axis, Y Axis, Z Axis in the Spectrometer Configuration window (opened from `config`). The values available for each axis are None, WFG + GCU, Performa I, Performa II/III, Performa II/III + WFG, Performa XYZ, Performa XYZ + WFG, SIS (12 bit), Homospoil, and Shim DAC. WFG stands for the waveform generator; GCU stands for the gradient compensation unit; and Performa I, II, III, and XYZ are types of PFG modules.

Values: String of three characters (e.g., `'nnp'`). The first character is the gradient for the X axis, second for the Y axis, and third for the Z axis. Each axis has value `'n'` (None choice in Spectrometer Configuration window), `'w'` (WFG+GCU), `'l'` (Performa I), `'p'` (Performa II/III), `'q'` (Performa II/III + WFG), `'t'` (Performa XYZ), `'u'` (Performa XYZ + WFG), `'s'` (SIS (12 bit), or `'h'` (Homospoil). Homospoil is functional only for the Z axis.

See also: *VnmrJ Installation and Administration; NMR Spectroscopy User Guide*

Related:

`config`	Display current configuration and possibly change it (M)
`pfgon`	PFG amplifiers on/off control (P)

graphis — Return the current graphics display status (C)

Syntax: (1) `graphis:$display_command`
(2) `graphis(command):$yes_no`

Description: Determines what command currently controls the graphics window.

Arguments: `$display_command` is a return value set to the name of the currently controlling command.

`command` is the name of a command to be checked.

`$yes_no` is a return value set to 1 if the command name given by the `command` argument is controlling the graphics window, or set to 0 if it is not controlling the window.

Examples:

```
graphis:$display
if ($display='ds') then
...
endif
```

```
graphis('ds'):$ds_on
if ($ds_on) then
...
endif
```

See also: *User Programming*

Related:

`textis`	Return the current text display status (C)

grayctr Gray level window adjustment (P)

Description Controls the grayscale display available in dcon. In the dconi program, the center mouse button controls the grayscale bar, which changes the mean gray level and hence the value of grayctr. The grayctr parameter (along with the parameter graysl) records the current settings of the gray bar as the interaction changes; the value can also be set directly. The right mouse button controls the data level of the maximum data intensity. To create grayctr, enter create('grayctr','real') setgroup('grayctr','display') setlimit('grayctr',64,0,1).

To create the set of imaging parameters grayctr, dcrmv and graysl, and in the current experiment, enter addpar('image').

Values 0 to 64 (typically 32)

See also *NMR Spectroscopy User Guide*

Related	
addpar	Add selected parameters to the current experiment (M)
dcon	Display noninteractive color intensity map (C)
dconi	Interactive 2D contour display (C)
graysl	Gray level slope (contrast) adjustment (P)

graysl Gray level slope (contrast) adjustment (P)

Description Controls the grayscale display available in dcon. In the dconi program, the center mouse button controls the grayscale slope as applied to the data changes and hence the value of graysl. Negative values of graysl will invert black and white; however, negative values can be set only from the keyboard. graysl (along with the parameter grayctr) records the current settings of the gray bar as the interaction changes; the value can also be set directly. The right mouse button controls the data level of the maximum data intensity. To create graysl, enter the following command:

create('graysl','real') setgroup('graysl','display') setlimit('graysl',10,–10,0.1)

To create the set of imaging parameters graysl, dcrmv, and grayctr in the current experiment, enter addpar('image').

Values –10 to +10 (–100 to +100, typically 1)

See also *NMR Spectroscopy User Guide*

Related	
addpar	Add selected parameters to the current experiment (M)
dcon	Display noninteractive color intensity map (C)
dconi	Interactive 2D contour display (C)
grayctr	Gray level window adjustment (P)

grecovery — Eddy current testing (M)

Applicability Systems with pulsed field gradient.

Description Conditions an experiment for eddy current testing so that it is compatible with standard installation procedures.

See also *Pulsed Field Gradient Modules Installation, NMR Spectroscopy User Guide*

grid — Draw a grid on a 2D display (M)

Syntax (1) grid<(<spacing><,><color>)>
(2) grid<(start_f2,incr_f2,start_f1,incr_f1<,color>)>

Description Draws grid lines over a 2D display. Grid lines are drawn on the graphics screen in the XOR mode—entering a second grid command with identical arguments erases (not redraws) the grid displayed by the first command.

Arguments spacing specifies the approximate spacing of the grid lines, in cm. The default is intervals of approximately 1 cm, rounded so that the intervals fall at a multiple of 1, 2, or 5 (in Hz), or 1p, 2p, or 5p (in ppm).

color specifies the color of the grid lines and is one of the following keywords: 'red', 'green', 'blue', 'cyan', 'magenta', 'yellow', 'black', or 'white'. The default is 'blue'.

start_f2, incr_f2, start_f1, incr_f1 define a grid by supplying the starting and increment frequencies for f2 and f1. Add the p suffix to a value to enter it in ppm (see third example below).

Examples

```
grid
grid(1.5,'red')
grid(1p,0.5p,3p,0.5p)
```

See also *NMR Spectroscopy User Guide*

Related plgrid Plot a grid on a 2D plot (M)

groupcopy — Copy parameters of group from one tree to another (C)

Syntax groupcopy(from_tree,to_tree,group)

Description Copies a set of parameters of a group from one parameter tree to another.

Arguments from_tree, to_tree are two different parameter trees, each given by the one of the keywords 'global', 'current', or 'processed'. Refer to the create command for more information on trees.

group is the set of parameters to be copied and is one of the keywords 'all', 'sample', 'acquisition', 'processing', and 'display'.

Examples `groupcopy('processed','current','acquisition')`

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`destroy`	Destroy a parameter (C)
`destroygroup`	Destroy parameters of a group in a tree (C)
`display`	Display parameters and their attributes (C)
`setgroup`	Set group of a parameter in a tree (C)

`gspoil` Spoiler gradient level (P)

Description Predefined parameter to set a spoiler gradient level.

`gsspat` Slice-select gradient shape (P)

Description Predefined string parameter to specify a slice-select gradient shape.

`gtnnoesy` Set up a PFG TNNOESY parameter set (M)

Applicability Systems with the pulsed field gradient (PFG) module.

Description Converts a 1D standard two-pulse sequence parameter set into a parameter set ready to run a PFG NOESY experiment (either absolute value or phase sensitive) or a `gtnnoesy` experiment.

`gtnroesy` Set up a PFG absolute-value ROESY parameter set (M)

Applicability Systems with the pulsed field gradient (PFG) module.

Description Converts a 1D standard two-pulse sequence parameter set into a parameter set ready to run a PFG absolute-value ROESY experiment or a `gtnroesy` experiment.

`gtotlimit` Gradient total limit (P)

Applicability Systems with three-axis gradients

Description Sets the gradient limit, in gauss/cm, of the x, y, and z axes, summed together. This parameter is taken from an entry of the same name in a gradient table and should only exist if a gradient amplifier limits the combined output of all three gradient axis.

Related `gcoil` Read data from gradient calibration tables (P)

gtrim Trim gradient level (P)

Description Predefined parameter to set a trim gradient level.

gxmax, gymax, gzmax Maximum gradient strength for each axis (P)

Applicability Systems with three-axis gradients.

Description Defines the maximum gradient strength, in gauss/cm, for each gradient axis. These values are read in from the selected system gradient table whenever the parameter set is retrieved or the gradient coil defined by `gcoil` has changed. When the values are read in, `gmax` is set to the lowest value of the three.

The parameters `gxmax`, `gymax`, and `gzmax` are used instead of `gmax` when the gradients strengths are not equal for each axis. Unequal gradient strengths per axis are generally true for systems with three-axis PFG coils, which have a strong *z* gradient, and can be true for microimaging systems. Horizontal-bore imaging systems usually have gradients set to the same maximum value, and `gmax` can be used.

See also *NMR Spectroscopy User Guide; User Programming, VnmrJ Imaging NMR*

Related
`gcoil` Read data from gradient calibration tables (P)
`gmax` Maximum gradient strength (P)

gzlvl Pulsed field gradient strength (P)

Applicability Systems with gradient shimming installed.

Description Specifies the pulsed field gradient DAC value.

Values Range from +2047 to –2048 for 12-bit gradient module, and from +32767 to
–32768 for a 16-bit gradient module.

Related
`gzsize` Number of z-axis shims used by gradient shimming (P)
`gzwin` Spectral window percentage used for gradient shimming (P)

gzsize Number of z-axis shims used by gradient shimming (P)

Applicability Systems with gradient shimming installed.

Description Specifies the number of z-axis shims used by gradient shimming. For example, gzsize set to 4 means that gradient shimming uses shims z1 to z4. By default, coarse shims are used if present, as determined by the shimset value

Values Integer from 1 to 8.

Related

gmapshim	Start gradient autoshimming (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
gzlvl	Pulsed field gradient strength (P)
gzwin	Spectral width percentage used by gradient shimming (P)
shimset	Type of shimset (P)
gmap_z1z4	Gradient shimming flag to first shim z1-z4 (P)

gzwin Spectral width percentage used for gradient shimming (P)

Applicability Systems with gradient shimming installed.

Description Specifies the percentage of the spectral width sw used by gradient shimming for shimmap calculations. The value is set automatically with the buttons Find gzlvl/gzwin and Find gzwin in the gradient shimming system menu opened by gmapsys.

Values A real number between 0 and 100. The typical value is 50.

Related

gmapshim	Start gradient autoshimming (M)
gmapsys	Run gradient autoshimming, set parameters, map shims (M)
gmapz	Get parameters and files for gmapz pulse sequence (M)
gzlvl	Pulsed field gradient strength (P)
gzsize	Number of z-axis shims used by gradient shimming (P)
sw	Spectral width in directly detected dimension (P)
tof	Frequency offset for observe transmitter (P)

H

h1	Automated proton acquisition (M)
h1freq	Proton frequency of spectrometer (P)
h1p	Process 1D proton spectra (M)
h2cal	Calculate strength of the decoupler field (C)
halt	Abort acquisition with no error (C)
hc	Automated proton and carbon acquisition (M)
hcapt	Automated proton, carbon, and APT acquisition (M)
hcchtocsy	Set up parameters for HCCHTOCSY pulse sequence (M)
hccorr	Automated proton, carbon, and HETCOR acquisition (M)
hcdept	Automated proton, carbon, and DEPT acquisition (M)
hcosy	Automated proton and COSY acquisition (M)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)
hcmult	Execute protocol actions of apptype hcmult (M)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Power level for homodecoupling (P)
hdpwrf	Homodecoupling fine power (optional) (P)
hdres	Sets the tip angle resolution (P)
hdseq	Sets the decoupler waveform filename (P)
hdwshim	Hardware shimming (P)
hdwshimlist	List of shims for hardware shimming (P)
help	Display current help file
HELP	Help file for this tool
het2dj	Set up parameters for HET2DJ pulse sequence (M)
HETCOR	Change parameters for HETCOR experiment (M)
hetcor	Set up parameters for HETCOR pulse sequence (M)

hetcorcp1	Set up parameters for solids HETCOR pulse sequence (M)
hetcorps	Set up parameters for HETCORPS pulse sequence (M)
hetero2d	Execute protocol actions of apptype hetero2d (M)
hidecommand	Execute macro instead of command with same name (C)
hipwrampenable	High Power Amplifier Enable (P)
Hmbc	Convert the parameter to a HMBC experiment (M)
Hmqc	Convert the parameter to a HMQC experiment (M)
HMQC15	Set up parameters for ^{15}N HMQC experiment (M)
HMQC_d2	Set up parameters for ^{15}N HMQC experiment using dec. 2 (M)
HMQC_d213	Set up parameters for ^{13}C HMQC experiment using dec. 2 (M)
hmqcr	Set up parameters for HMQCR pulse sequence (M)
Hmqctoxy	Convert the parameter to a HMQCTOXY experiment (M)
HMQCTOXY15	Set up parameters for ^{15}N HMQCTOXY experiment (M)
HMQCTOXY_d2	Set up parameters for ^{15}N HMQCTOXY using decoupler 2 (M)
HMQCTOXY_d213	Set up parameters for ^{13}C HMQCTOXY using decoupler 2 (M)
hmqctoxy3d	Set up parameters for HMQC-TOCSY 3D pulse sequence (M)
ho	Horizontal offset (P)
hom2dj	Set up parameters for HOM2DJ pulse sequence (M)
homo	Homodecoupling control for the observe channel (P)
HOMODEC	Change parameters for HOMODEC experiment (M)
homo2d	Execute protocol actions of apptype homo2d (M)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amp and setting T/R to receive (P)
homorof3	Delay between setting T/R switch to receive and gating the recvr on (P)
hoult	Set parameters alfa and rof2 according to Hoult (M)
hpa	Plot parameters on special preprinted chart paper (C)
Hprescan	Proton prescan (P))
hregions	Select integral regions in proton spectrum (M)
hs	Homospoil pulses (P)
Hsqc	Convert the parameter to a HSQC experiment (M)
HSQC15	Set up parameters for ^{15}N HSQC experiment (M)
HSQC_d2	Set up parameters for ^{15}N HSQC experiment using dec. 2 (M)

HSQC_d213	Set up parameters for ^{13}C HSQC experiment using dec. 2 (M)
HsqcHT	Set up the hsqcHT experiment (M)
Hsqctoxy	Convert parameters to a HSQCTOXY experiment (M)
HSQCTOXY15	Set up parameters for ^{15}N HSQCTOXY experiment (M)
HSQCTOXY_d2	Set up parameters for ^{15}N HSQCTOXY using decoupler 2 (M)
HSQCTOXY_d213	Set up parameters for ^{13}C HSQCTOXY using decoupler 2 (M)
hsqctoxySE	Set up parameters for HSQC-TOCSY 3D pulse sequence (M)
hsrotor	Display rotor speed for solids operation (P)
hst	Homospoil time (P)
ht	Setting up and processing Hadamard experiments
htbitrev	Hadamard bit reversal flag (P)
htbw1	Hadamard pulse excitation bandwidth in ni (P)
htcal1	RF calibration flag for Hadamard waveforms in ni (P)
htfrq1	Hadamard frequency list in ni (P)
htfrqdisp	Read, write, and display Hadamard frequencies
htofs1	Hadamard offset in ni (P)
htpwr1	Power level for RF calibration of Hadamard waveforms in ni (P)
htss1	Stepsize for Hadamard waveforms in ni (P)
hzmm	Scaling factor for plots (P)
hztomm	Convert locations from Hz or ppm to plotter units (C)

h1 — Automated proton acquisition (M)

Syntax h1<(solvent)>

Description Prepares parameters for automatically acquiring a standard ^{1}H spectrum. The parameter wexp is set to 'procplot' for standard processing. If h1 is used as the command for automation via the enter command, then au is supplied automatically and should not be entered on the MACRO line of the enter program. However, it is possible to customize h1 on the MACRO line by following it with additional commands and parameters. (e.g., entering h1 nt=1 uses the standard h1 setup but with only one transient).

Arguments solvent is the name of the solvent. In automation mode, the solvent is supplied by the enter program. The default is 'CDCl3'.

Examples

```
h1
h1('DMSO')
```

See also *NMR Spectroscopy User Guide*

Related
au	Submit experiment to acquisition and process data (M)
enter	Enter sample information for automation run (C)
h1p	Process 1D proton spectra (M)
procplot	Automatically process FIDs (M)
wexp	When experiment completes (P)

h1freq Proton frequency of spectrometer (P)

Description Configuration parameter for the resonance frequency of ^{1}H as determined by the field strength of the magnet. The value is set using the label Proton Frequency in the Spectrometer Configuration window.

Values 085, 100, 200, 300, 400, 500, 600, 700, 750, 800, 900 (in MHz); 3T, 4T.

See also *VnmrJ Installation and Administration*

Related
config	Display current configuration and possibly change it (M)

h1p Process 1D proton spectra (M)

Description Processes non-arrayed 1D proton spectra using standard macros. `h1p` is called by `proc1d`, but can also be used directly. Fully automatic processing (up to a point where a spectrum could be plotted) is provided: Fourier transformation (using preset weighting functions), automatic phasing (`aphx` macro), select integral regions (`hregions` macro), adjust integral size (`integrate` macro), vertical scale adjustment (`vsadjc` macro), avoiding excessive noise (`noislm` macro), threshold adjustment (if required, `thadj` macro), and referencing to the TMS signal if present (`setref` macro, then `tmsref` macro).

See also *NMR Spectroscopy User Guide*

Related
aphx	Perform optimized automatic phasing (M)
h1	Automated proton acquisition (M)
hregions	Select integral regions for proton spectra (M)
integrate	Automatically integrate 1D spectrum (M)
noislm	Avoids excessive noise (M)
proc1d	Processing macro for simple (non-arrayed) spectra (M)
setref	Set frequency referencing for proton spectra (M)
thadj	Adjust threshold (M)
tmsref	Reference spectrum to TMS line (M)
vsadjh	Adjust vertical scale for proton spectra (M)

h2cal Calculate strength of the decoupler field (C)

Syntax `h2cal<(j1r,j2r<,j0>)><:gammah2,pw90,frequency>`

Description Calculates the strength of the decoupler field. It uses the results from two experiments: one with the decoupler off-resonance at a lower frequency and the other with the decoupler off-resonance at a higher frequency than the frequency of the peak being decoupled.

Arguments `j1r` is the frequency of the decoupler during these two experiments;. The default is that `h2cal` prompts for a value. If the parameter `dof` is arrayed and has two values, `h2cal` assumes these two values represent the decoupler frequencies; if `dof` is arrayed and has more than two values, `h2cal` prompts for the two decoupler frequencies.

`j2r` is the reduced coupling constants from the two experiments. The default is that `h2cal` prompts for a value

`j0` is the full coupling constant that results when no decoupling is done. The default is a value of 142 Hz, the constant for the standard sample dioxane, or 15 Hz for the methyl iodide sample.

`gammah2` is a return value set to the strength of the decoupler field.

`pw90` is a return value set to the pulse width of a 90° pulse from the decoupler. It is related to the value of parameter `dmf` through the equation `dmf=1/pw90`.

`frequency` is a return value set to the coalescence point (i.e., frequency at which single-frequency decoupling would collapse the dioxane to a singlet).

See also *NMR Spectroscopy User Guide*

Related `dmf` Decoupler modulation frequency for first decoupler (P)
`dof` Frequency offset for first decoupler (P)

halt Abort acquisition with no error (C)

Syntax `halt`

Description Aborts an experiment that has been submitted to acquisition. If the experiment is active, it is aborted immediately, all data is discarded, and the experiment is interpreted as complete. Any data collected from an earlier block size transfer is retained. If any `wexp` processing is defined, that processing then occurs, followed by any queued experiments. The login name, and the FID directory path in `file` are used as keys to find the proper experiment to abort.

Under some circumstances, there is a delay between the time `go` is entered and the acquisition is started. During this time, instructions based on the selected pulse sequence are being generated. This is signified by the letters “PSG” appearing in the upper left corner of the status window. A `halt` command issued under these circumstances reports that no acquisition is active but it instead stops the instruction generation process and displays “PSG aborted”.

See also *NMR Spectroscopy User Guide*

Related

aa	Abort acquisition with error (C)
file	File name of parameter set (P)
go	Submit experiment to acquisition (C)
wexp	Specify action when experiment completes (C)
wexp	When experiment completes (P)

hc Automated proton and carbon acquisition (M)

Syntax `hc<(solvent)>`

Description Combines the operation of the h1 and c13 macros. In non-automation mode, both spectra are acquired in the experiment in which the hc macro was entered. After the completion of the acquisition, rttmp can be used for further processing of the two spectra.

Arguments `solvent` is the solvent name In automation mode, the enter program supplies the value. In non-automation mode, the default is 'cdcl3'.

Examples
```
hc
hc('dmso')
```

See also *NMR Spectroscopy User Guide*

Related

c13	Automatic carbon acquisition (M)
enter	Enter sample information for automation run (M,U)
h1	Automated proton acquisition (M)
rttmp	Retrieve experiment data from experiment subfile (M)

hcapt Automated proton, carbon, and APT acquisition (M)

Syntax `hcapt<(solvent)>`

Description Combines the operation of the h1 and c13 macros and the APT experiment. In non-automation mode, all spectra are acquired in the experiment in which the hcapt macro was entered. After acquisition completes, rttmp can be used for further processing of the three spectra.

Arguments `solvent` is the solvent name. In automation mode, the enter program supplies the value. In non-automation mode, the default is 'cdcl3'.

Examples
```
hcapt
hcapt('dmso')
```

See also *NMR Spectroscopy User Guide*

Related

Apt	Set up parameters for APT experiment (M)
c13	Automatic carbon acquisition (M)
enter	Enter sample information for automation run (M,U)

h1	Automated proton acquisition (M)
rttmp	Retrieve experiment data from experiment subfile (M)

hcchtocsy Set up parameters for HCCHTOCSY pulse sequence (M)

Description Used for sidechain assignments in fully ^{13}C-enriched molecules.

See also *NMR Spectroscopy User Guide*

hccorr Automated proton, carbon, and HETCOR acquisition (M)

Syntax `hccorr<(solvent)>`

Description Combines the operation of the h1 and c13 macros and the HETCOR experiment. In non-automation mode, all spectra are acquired in the experiment in which hccorr is entered. After acquisition completes, rttmp can be used for further processing of the three spectra.

Arguments solvent is the solvent name. In automation mode, the enter program supplies the value. In non-automation mode, the default is 'cdcl3'.

Examples
```
hccorr
hccorr('dmso')
```

See also *NMR Spectroscopy User Guide*

Related

c13	Automated carbon acquisition (M)
enter	Enter sample information for automation run (M,U)
h1	Automated proton acquisition (M)
hetcor	Set up parameters for HETCOR experiment (M)
rttmp	Retrieve experiment data from experiment subfile (M)

hcdept Automated proton, carbon, and DEPT acquisition (M)

Syntax `hcdept<(solvent)>`

Description Combines the operation of the h1 and c13 macros and the DEPT experiment. In non-automation mode, all spectra are acquired in the experiment in which hcdept was entered. After the completion of the acquisition, rttmp can be used for further processing of the three spectra.

Arguments solvent is the solvent name. In automation mode, the enter program supplies the value. In non-automation mode, the default is 'cdcl3'.

Examples
```
hcdept
hcdept('dmso')
```

See also	*NMR Spectroscopy User Guide*	
Related	c13	Automatic carbon acquisition (M)
	Dept	Set up parameters for DEPT experiment (M)
	enter	Enter sample information for automation run (M,U)
	h1	Automated proton acquisition (M)
	rttmp	Retrieve experiment data from experiment subfile (M)

hcosy — Automated proton and COSY acquisition (M)

Syntax `hcosy<(solvent)>`

Description Combines the operation of the h1 macro and the COSY experiment. In non-automation mode, both spectra are acquired in the experiment in which `hcosy` is entered. After acquisition completes, rttmp can be used for further processing of the two spectra.

Arguments `solvent` is the solvent name. In automation mode, the enter program supplies the value. In non-automation mode, the default is `'cdcl3'`.

Examples
```
hcosy
hcosy('dmso')
```

See also *NMR Spectroscopy User Guide*

Related	enter	Enter sample information for automation run (C)
	h1	Automated proton acquisition (M)
	rttmp	Retrieve experiment data from experiment subfile (M)

hdmf — Modulation frequency for homonuclear decoupling (P)

Applicability VNMRS liquids, 400 MR

Syntax `hdmf=<value>`

Description Sets the modulation frequency for the band selective homonuclear decoupling. The parameter specifies `1/pw90` at the power value, `hdpwr`, used for homonuclear decoupling. The parameter is not used with single frequency homonuclear decoupling.

Related	dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
	hdof	Frequency offset for homodecoupling (P)
	hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
	hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
	hdres	Sets the tip angle resolution (P)
	hdseq	Sets the decoupler waveform filename (P)

homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

hcmult — Execute protocol actions of apptype hcmult (M)

Description This macro is used to execute the protocol actions of the hcmult apptype.

Examples `hcmult('setup')` – execute hcmult experimental setup
`hcmult('process')` – execute hcmult processing
`hcmult('plot')` – execute hcmult plotting

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

apptype	Application type (P)
execpars	Set up the exec parameters (M)

hdof — Frequency offset for homodecoupling (P)

Applicability VNMRS systems

Syntax `hodf=<value>`

Description Sets the irradiation frequency offset for homonuclear decoupling and similar to how `tof`, and `dof` determine the frequency. The parameter is not used if `hdseq` is set to a filename.

Values –100000 to 100000 Hz in steps of 0.1 Hz.

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	modulation frequency for the band selective homonuclear decoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Homodecoupling fine power (optional) (P)
hdres	Sets the tip angle resolution (P)
hdseq	Sets the decoupler waveform filename (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

hdpwr Power level for homodecoupling (P)

Applicability VNMRS systems, 400 MR

Syntax `hdpwr=<value>`

Description Sets the rf attenuator to control the power for homonuclear decoupling. The `dutyc` parameter must be accounted for when setting `hdpwr`.

Values -16 to 50 dB

CAUTION Homodecoupling power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate homodecoupling to avoid exceeding 2 watts. The maximum value for hdpwr is set to 49, corresponding to about 2 watts of power. The actual power delivered depends on the CW duty cycle. Before using close to the maximum value of power or duty cycle, ensure safe operation by measuring the output power.

Related

`dutyc`	The rf duty cycle fraction for homonuclear decoupling (P)
`hdmf`	modulation frequency for the band selective homonuclear decoupling (P)
`hdof`	Frequency offset for homodecoupling (P)
`hdpwrf`	Homodecoupling fine power (optional) (P)
`hdres`	Sets the tip angle resolution (P)
`hdseq`	Sets the decoupler waveform filename (P)
`homo`	Homodecoupling control for observe channel (P)
`homorof1`	Delay before turning on homo decoupling rf (P)
`homorof2`	Delay after blanking the amplifier and setting T/R switch to receive (P)
`homorof3`	Delay between setting T/R switch to receive gating on the receiver (P)
`tn`	Nucleus for observe transmitter (P)

hdpwrf Homodecoupling fine power (optional) (P)

Applicability VNMRS liquids systems

Syntax `hdpwrf=<value>`

Description Sets the rf linear modulator fine power for homonuclear decoupling. The default is 4095 if the variable does not exist. Attenuation is added to the attenuation set by `hdpwr`.

Values 0-4095

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdres	Sets the tip angle resolution (P)
hdseq	Sets the decoupler waveform filename (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

hdres Sets the tip angle resolution (P)

Applicability VNMRS liquids systems

Syntax `hdres=<value>`

Description Sets the tip angle resolution to be used for the band selective waveform mode of homonuclear decoupling. The parameter is not used with single frequency homonuclear decoupling.

Values 1 to 90 in units of degrees with 1 degree resolution

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
hdseq	Sets the decoupler waveform filename (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

hdseq — Waveform filename for band selective decoupling (P)

Applicability VNMRS liquids systems

Syntax `hdseq='filename'` — the file must have a `.DEC`. extension.

Description Sets the decoupler waveform filename (.DEC extension) for the band selective waveform mode. The irradiation frequency is determined by the transmitter offset last applied to the observe channel in the pulse sequence (typically `tof`) and any additional frequency offset from any phase modulation programmed implicitly into the waveform .DEC file.

Examples `hdseq=''` or does not exist — single frequency decoupling is used.

Related

dutyc	
hdmf	modulation frequency for the band selective homonuclear decoupling (P)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
hdres	Sets the tip angle resolution (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

hdwshim — Hardware shimming (P)

Applicability Systems with additional Z1 shimming hardware.

Description Allows go, su, au, etc., to turn on and off shimming hardware. Hardware shimming is automatically suspended during software autoshimming. Hardware shimming is only active during acquisition (`go`, `ga`, `au`). `hdwshim` is a global parameter, so it affects all experiments.

Values `'y'` turns hardware shimming on.

`'p'` turns hardware shimming on during presaturation pulse (power level change followed by pulse).

`'n'` turns shimming off.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (C)
go	Submit experiment to acquisition (C)
su	Submit a setup experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (M)

hdwshimlist List of shims for hardware shimming (P)

Description A global parameter that sets the shims to use during hardware shimming. If it does not exist, hardware shimming uses z1 by default. To create the parameter, use `create('hdwshimlist','string','global')`.

Values Any string composed of z1, z1c, z2, z2c, x1, y1. Commas and blank space are ignored. Shimming is done in the order z1, z2, x1, y1, regardless of the order in the string.

Examples
```
hdwshimlist='z1'
hdwshimlist='z1z2x1y1'
```

See also *NMR Spectroscopy User Guide*

Related		
	create	Create new parameter in a parameter tree (C)
	hdwshim	Hardware shimming (P)

help Display current help file

Syntax `help`

Applicability VnmrJ 3.1

Description This command displays help information that explains the functions of the buttons and current utility active. This information is displayed in the text window. The permanent help button on the Sun executes the help command.

The help information that is displayed is from a file located in directory /vnmr/help. The name of the file matches the name of the currently active menu.

HELP Help File for this Tool

Syntax `HELP`

Applicability VnmrJ 3.1

Description By default the help file for the current experiment defined by the pslabel parameter is shown. For all of the experiments that are found in the experiment selector items are listed under the Tab names. Thus, for example, there is a help file called J1CHTab which is general information for all of the supplied experiments located in that tab. Similarly there is an entry for Homo2DTab with general information for that group of experiments. For convenience the menu order of this help file arranged by the Tab names. If one opens Seq.Help the manual-select menu will be positioned so that all of the similar experiments for that group are nearby. All of help files are in alphabetical order.

There are two menus. One to select a specific experiment's help file and the other to conveniently view a help file describing all experiments in a given tab in the experiment selector. Both experiment and Tab help is available under the experiment menu.

het2dj — Set up parameters for HET2DJ pulse sequence (M)

Description	Sets up a HET2DJ (heteronuclear 2D-J) experiment.	
See also	*NMR Spectroscopy User Guide*	
Related	`foldj`	Fold J-resolved 2D spectrum about *f1=0* axis (C)

HETCOR — Change parameters for HETCOR experiment (M)

Description	Converts the current parameter set to a HETCOR experiment. This is a phase-sensitive, multiplicity-selected experiment.

hetcor — Set up parameters for HETCOR pulse sequence (M)

Syntax	`hetcor<(exp_number)>`	
Description	Sets up a HETCOR (heteronuclear chemical shift correlation) experiment.	
Arguments	`exp_number` is the number of the experiment, from 1 to 9, in which a proton spectrum of the sample already exists.	
See also	*NMR Spectroscopy User Guide*	
Related	`plhxcor`	Plot X,H-correlation 2D spectrum (M)
	`ppcal`	Proton decoupler pulse calibration (M)

hetcorcp1 — Set up parameters for solids HETCOR pulse sequence (M)

Applicability	Systems with the solids module.	
Description	Sets up a parameter set, obtained withXPOLAR1, for HETCORCP1, the solid-state heteronuclear correlation experiment.	
See also	*User Guide: Solid-State NMR*	
Related	`xpolar1`	Set up parameters for XPOLAR1 pulse sequence (M)

hetcorps Set up parameters for HETCORPS pulse sequence (M)

Description	Sets up parameters for a heteronuclear chemical shift correlation experiment (absolute value and phase sensitive).
See also	*NMR Spectroscopy User Guide*

hetero2d Execute protocol actions of apptype hetero2d (M)

Applicability	Liquids
Description	Perform the actions for Homonuclear 2D protocols to set up, process, and plot experiments.
Examples	`hetero2d('setup')` execute hetero2d experimental setup `hetero2d('process')` execute hetero2d processing `hetero2d('plot')` execute hetero2d plotting
See also	*NMR Spectroscopy User Guide, VnmrJ Walkup*
Related	`apptype` Application type (P) `execpars` Set up the exec parameters (M)

hidecommand Execute macro instead of command with same name (C)

Syntax	(1) `hidecommand(command_name)<:$new_name>` (2) `hidecommand('?')`
Description	Renames (or hides) a built-in VnmrJ command so that a macro with the same name as the built-in command is executed instead of the built-in command.
Arguments	`command_name` is the name of the command to be renamed. To reset the built-in command back to its original name, enter `hidecommand` with the hidden name as the argument. `$new_name` returns the new name of the built-in command. By using this new name, access is still available to the built-in command. `'?'` is a keyword to display a list of all of the renamed built-in commands and their original names.

Examples	`hidecommand('sys'):$newname`
	`hidecommand('Sys')`
	`hidecommand('?')`

See also	*System Administration; User Programming*
Related	`which` Display which macro or command is used (M)

`hipwrampenable` High Power Amplifier Enable (P)

Applicability: VNMRS solids and systems with high power amplifiers.

Description: This parameter controls the High/Low Power Relay. If the parameter does not exist low power is used. If the parameter exists and the field corresponding to the physical channel is `'n'` then low power is used. If the parameter exists and the field corresponding to the physical channel is `'y'` then high power is used. The parameter is created in the current tree as a flag with `create('hipwrampenable','flag')`.

Values:
`'y'` Enable high power
`'n'` Enable low power and disable high power

Examples: `hipwrampenable='yny'`

Physical channel 1 and 3 are high power enabled. Physical channel 2 is low power.

`Hmbc` Convert the parameter to a HMBC experiment (M)

Description: Convert the parameter to a HMBC experiment.

See also: *NMR Spectroscopy User Guide*

`Hmqc` Convert the parameter to a HMQC experiment (M)

Description: Convert the parameter to a HMQC experiment.

`HMQC15` Set up parameters for ^{15}N HMQC experiment (M)

Description: Converts the current parameter set to a HMQC experiment for ^{15}N.

`HMQC_d2` Set up parameters for ^{15}N HMQC experiment using dec. 2 (M)

Description: Converts the current parameter set to a HMQC experiment for ^{15}N with decoupler 2 as ^{15}N.

HMQC_d213 Set up parameters for ^{13}C HMQC experiment using dec. 2 (M)

Description Converts the current parameter set to a HMQC experiment for ^{13}C with decoupler 2 as ^{13}C.

hmqcr Set up parameters for HMQCR pulse sequence (M)

Applicability Not needed in current systems. Normally was used in systems with a ^{1}H only decoupler.

Description Sets up a HMQC (heteronuclear multiple-quantum coherence) experiment with “reverse” configuration.

See also *NMR Spectroscopy User Guide*

Hmqctoxy Convert the parameter to a HMQCTOXY experiment (M)

Description Convert the parameter to a HMQCTOXY experiment.

HMQCTOXY15 Set up parameters for ^{15}N HMQCTOXY experiment (M)

Description Converts the current parameter set to a HMQCTOXY experiment for ^{15}N.

HMQCTOXY_d2 Set up parameters for ^{15}N HMQCTOXY using decoupler 2 (M)

Description Converts the current parameter set to a HMQCTOXY experiment for ^{15}N with decoupler 2 as ^{15}N.

HMQCTOXY_d213 Set up parameters for ^{13}C HMQCTOXY using decoupler 2 (M)

Description Converts the current parameter set to a HMQCTOXY experiment for ^{13}C with decoupler 2 as ^{13}C.

hmqctoxy3d — Set up parameters for HMQC-TOCSY 3D pulse sequence (M)

Description: Sets up parameters for a HMQC-TOCSY 3D experiment with a presaturation option.

ho — Horizontal offset (P)

Description: Horizontal offset of the each spectrum in a “stacked display” with respect to the previous spectrum,. For 1D data sets, the parameter `vo` sets the vertical offset. For 2D data sets, the parameter `wc2` sets the vertical distance (in mm) between the first and last traces.

Values: Number, in mm, for offset size. For a “left-to-right” presentation, `ho` is typically negative; for “bottom-to-top” presentation, `vo` or `wc2` is positive.

hom2dj — Set up parameters for HOM2DJ pulse sequence (M)

Description: Sets up a HOM2DJ (homonuclear J-resolved 2D) experiment.

See also: *NMR Spectroscopy User Guide*

homo — Homodecoupling control for the observe channel (P)

Applicability: VNMRS liquids systems

Description: Homonuclear decoupling irradiates a single frequency if `hdseq = ''` (or if `hdseq` does not exist) or a frequency band if `hdseq ='filename'`. Pulse sequences do not require explicit homonuclear gating commands (`homo` function is similar to `dm`). A single RF channel, the observe channel, is used. The `homo='y'` setting cannot be used with pulse sequences containing explicit acquire commands.

Syntax: `homo=<'y' or 'n'>`

Values: `'y'` homonuclear decoupling rf and receiver gating is turned on during the acquisition time. Provides single frequency or band selective (`hdseq = 'filename'`) decoupling.

`'n'` homonuclear decoupling rf and receiver gating is turned off.

Related:

hdof	Frequency offset for homodecoupling (P)
hdpwr	Power level for homodecoupling (P)
hdpwrf	Homodecoupling fine power (P)
dutyc	Duty cycle for homodecoupling (optional) (P)
tn	Nucleus for observe transmitter (P)
homorof1	Delay before turning on homo decoupling rf (P)

homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)

HOMODEC — Change parameters for HOMODEC experiment (M)

Description Converts the current parameter set to a HOMODEC experiment. A 1D proton spectrum is displayed to do peak selection.

homo2d — Execute protocol actions of apptype homo2d (M)

Applicability Liquids

Description Perform the actions for Heteronuclear 2D protocols to set up, process, and plot experiments.

Examples

`homo2d('setup')`	execute homo2d experimental setup
`homo2d('process')`	execute homo2d processing
`homo2d('plot')`	execute homo2d plotting

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

apptype	Application type (P)
execpars	Set up the exec parameters (M)

homorof1 — Delay before turning on homo decoupling rf (P)

Applicability VNMRS liquids systems

Description Optional parameter for delay before turning on homonuclear decoupling after gating the receiver off. The amplifier is un-blanked and T/R switch set to transmit mode during homorof1 delay (in µsec. units). A default delay of 2 µsec. is used if the parameter does not exist.

Values 2 to 5 µsec. are typical.

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
hdseq	Sets the decoupler waveform filename (P)

hdres	Sets the tip angle resolution (P)
homo	Homodecoupling control for observe channel (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

homorof2 Delay after blanking the amp and setting T/R switch to recv (P)

Applicability VNMRS liquids systems

Description Optional parameter for delay after the transmitter is gated off, the amplifier is blanked, and before the T/R switch is set to receive. A default delay of 2 µsec. is used if the parameter does not exist.

Values 2 to 5 µsec. are typical.

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)
hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
hdseq	Sets the decoupler waveform filename (P)
hdres	Sets the tip angle resolution (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof3	Delay between setting T/R switch to receive gating on the receiver (P)
tn	Nucleus for observe transmitter (P)

homorof3 Delay between setting T/R to receive and gating the recvr on (P)

Applicability VNMRS liquids systems

Description Optional parameter for delay after the T/R switch is set to receive and before the receiver gate is gated on. A default delay of 2 µsec. is used if the parameter does not exist.

Values 2 to 5 µsec. are typical

Related

dutyc	The rf duty cycle fraction for homonuclear decoupling (P)
hdmf	Modulation frequency for the band selective homonuclear decoupling (P)

hdof	Frequency offset for homodecoupling (P)
hdpwr	Sets the rf attenuator to control the power for homonuclear decoupling (P)
hdpwrf	Sets the rf linear modulator fine power for homonuclear decoupling (P)
hdseq	Sets the decoupler waveform filename (P)
hdres	Sets the tip angle resolution (P)
homo	Homodecoupling control for observe channel (P)
homorof1	Delay before turning on homo decoupling rf (P)
homorof2	Delay after blanking the amplifier and setting T/R switch to receive (P)
tn	Nucleus for observe transmitter (P)

hoult Set parameters alfa and rof2 according to Hoult (M)

Description Sets the values of alfa and rof2 according to a prescription advanced by D. I. Hoult (*J. Magn. Reson.* **51**, 110 (1983)). These parameters set the times that follow the final pulse, which can be important where the flatness of the baseline is of concern.

See also *NMR Spectroscopy User Guide*

Related

alfa	Set alfa delay before acquisition (P)
calfa	Recalculate alfa so that first-order phase is zero (M)
rof2	Receiver gating time following pulse (P)

hpa Plot parameters on special preprinted chart paper (C)

Description Plots a predetermined list of parameters by “filling in the blanks” at the bottom of the preprinted chart paper available for Hewlett-Packard 7475- and 7550-series plotters.

See also *NMR Spectroscopy User Guide*

Related

apa	Plot parameters automatically (M)
x0	X-zero position of HP plotter or Postscript device (P)
y0	Y-zero position of HP plotter or Postscript device (P)

Hprescan Proton prescan (P))

Applicability *VnmrJ Walkup*

Description	This parameter is used to keep track of the type and status of the Proton prescan. It is used for Proton, Presat, Wet1d, and Minsw protocols.	
See also	*VnmrJ Walkup*	
Related	`xmHprescan`	Set up and process Proton prescans (M)

`hregions` Select integral regions in proton spectrum (M)

Description	Selects integral regions, a critical step in automatic processing of proton spectra. It is critical not only because of aesthetic reasons (some people like many small integrals, others prefer a few large regions), but also because other commands, such as `bc`, depend on the correct integration: `bc` can either fail or it can make broad, unintegrated lines disappear from the spectrum. `hregions` was specifically designed for proton spectra and should not be used for other types of spectra. The result of `hregions` also depends on the lineshape and the signal-to-noise ratio of a spectrum	
See also	*NMR Spectroscopy User Guide*	
Related	`bc`	1D and 2D baseline correction (C)
	`integrate`	Automatically integrate 1D spectrum (M)

`hs` Homospoil pulses (P)

Description	Turns on homospoil pulses at various times in different pulse sequences. Homospoil is a process by which the homogeneity is temporarily made very bad (“spoiled”) to cause any transverse magnetizations present at that time to decay rapidly to zero. `hst` controls the length of any homospoil pulse.	
Values	In a standard two-pulse sequence, homospoil pulses can be inserted during periods A and B (delays `d1` and `d2`): `hs='yn'` gives a homospoil pulse at the beginning of `d1`, `hs='ny'` gives a pulse during `d2`, and `hs='yy'` gives homospoil pulses during both `d1` and `d2`. The desired value is generally `hs='nn'`.	
See also	*NMR Spectroscopy User Guide*	
Related	`d1`	First delay (P)
	`d2`	Incremented delay in 1st indirectly detected dimension (P)
	`hst`	Homospoil time (P)

`Hsqc` Convert the parameter to a HSQC experiment (M)

Description	Convert the parameter to a HSQC experiment.

HSQC15 Set up parameters for ^{15}N HSQC experiment (M)

Description Converts the current parameter set to a HSQC experiment for ^{15}N.

HSQC_d2 Set up parameters for ^{15}N HSQC experiment using dec. 2 (M)

Description Converts the current parameter set to a HSQC experiment for ^{15}N with decoupler 2 as ^{15}N.

HSQC_d213 Set up parameters for ^{13}C HSQC experiment using dec. 2 (M)

Description Converts the current parameter set to a HSQC experiment for ^{13}C with decoupler 2 as ^{13}C.

hsqcHT Set up the hsqcHT experiment (M)

Description Sets up parameters for a Hadamard-encoded hsqc experiment.

See also *NMR Spectroscopy User Guide*

Related

htofs1	Hadamard frequency list in ni (P)
htfrq1	Hadamard offset in ni (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
ft2d	Fourier transform 2D data (C)
sethtfrq1	Set a Hadamard frequency list from a line list (M)
Hsqc	Set up parameters for HSQC experiment (M)

Hsqctoxy Convert parameters to a HSQCTOXY experiment (M)

Description Convert the parameter to a HSQCTOXY experiment.

HSQCTOXY15 Set up parameters for ^{15}N HSQCTOXY experiment (M)

Description Converts the current parameter set to a HSQCTOXY experiment for ^{15}N.

HSQCTOXY_d2 Set up parameters for ^{15}N HSQCTOXY using decoupler 2 (M)

Description Converts the current parameter set to a HSQCTOXY experiment for ^{15}N with decoupler 2 as ^{15}N.

HSQCTOXY_d213 Set up parameters for ^{13}C HSQCTOXY using decoupler 2 (M)

Description Converts the current parameter set to a HSQCTOXY experiment for ^{13}C with decoupler 2 as ^{13}C.

hsqctoxySE Set up parameters for HSQC-TOCSY 3D pulse sequence (M)

Description Sets up parameters for a HSQC -TOCSY 3D experiment.

hsrotor Display rotor speed for solids operation (P)

Applicability Systems equipped with the rotor synchronization module.

Description Controls display of rotor speed. Depending on whether the rotor synchronization module is present (set by the Rotor Synchronization label in the Spectrometer Configuration window, parameter `rotorsync` is set to 1 or 0. The `xpolar1` macro in turn uses this to create `hsrotor`, which is set to `'y'` if rotor synchronization is present. If the parameter `srate` exists, it is updated to the spin speed of the rotor at the end of the experiment. The interlock function specified by parameter `in` also changes. If `hsrotor='y'` and `in='y'`, the experiment is terminated if rotor speed deviates more than 100 Hz.

hst Homospoil time (P)

Description Controls pulse length if homospoil is activated by the `hs` parameter.

Values 0 to 20 ms (limited by hardware).

Values 'n' makes srate unmodified by acquisition and turns off the rotor speed display in Acqstat.

'y' makes the hardware information from the rotor synchronization board update srate and displays the rotor speed in the Acqstat status display.

See also *User Guide: Solid-State NMR*

Related

Acqstat	Bring up the acquisition status display (U)
config	Display current configuration and possibly change it (M)
in	Interlock (P)
rotorsync	Rotor synchronization (P)
srate	Spinning speed (P)
xpolar1	Set up parameters for XPOLAR1 pulse sequence (M)

ht — Setting up and processing Hadamard experiments.

Syntax `ht`

Applicability VnmrJ 3.1

Description To set up a Hadamard experiment, do the following.

1. First run a Proton, Carbon, or other 1D experiment.
2. When the acquisition is finished, process and phase the spectrum.
3. Run the editht macro to open the Edit HT Freq popup. Create a Hadamard frequency list for the nucleus of interest. Save the frequency list.
4. For a heteronuclear Hadamard experiment, run a Proton experiment, and adjust spectral width and decoupling as desired.
5. Load the desired Hadamard experiment. Check the Hadamard frequency list and other parameters.
6. Start the acquisition of the Hadamard experiment.
7. When acquisition is complete, process with proc1='ht' wft2da.

Parameters used:

htfrq1: Hadamard frequency list in indirect dimension.

ni: Number of increments in indirect dimension. Typically set to the size of htfrq1 plus htofs1.

htofs1: Offset in Hadamard processing (number of increments to skip).

proc1: Type of processing in indirect dimension. Set to 'ht'.

fn1: Fourier number in indirect dimension. It must be larger than the number of Hadamard frequencies, and larger than the minimum difference between Hadamard frequencies.

Hadamard Spectroscopy

Hadamard spectroscopy is a technique for acquiring multidimensional data sets using a small number of transients, and reconstructing the nD spectrum using a Hadamard transform. It is based on selective excitation of a predetermined set of frequencies.

Acquisition.

A list of frequencies to selectively excite is determined from a 1D spectrum, or other means. A series of shaped pulses is created from the frequency list, using a Hadamard matrix to selectively excite or invert the signals of interest. The matrix size must be greater than the number of frequencies in the list. A typical 8x8 Hadamard matrix is shown below.

++++++++

++++----

++--++--

++----++

+-+-+-+-

+-+--+-+

+--++--+

+--+-++-

Processing.

Hadamard processing in the indirect dimension is done by summing, adding or subtracting the acquired data increments in combinations according to the Hadamard matrix elements. Each sum gives a trace corresponding to one of the frequencies in the list, and is placed at the appropriate frequency in the indirect dimension. The areas between the frequencies in the list are zero filled. The direct dimension is Fourier transformed, giving the 2D spectrum.

See also
`editht`
`getht`
`HsqcHT`
`tocsyHT`
`sethtfrq1`
`ft2d`
`ft2da`

E. Kupce and R. Freeman, "Two-dimensional Hadamard spectroscopy," J. Magn. Reson. 162 (2003), pp. 300-310.

htbitrev Hadamard bit reversal flag (P)

Description A flag to enable or disable bit reversal of the Hadamard matrix. The flag should be the same for both acquisition and processing for the Hadamard transform to be successful.

Values 'y' enable Hadamard bit reversal
'n' disable Hadamard bit reversal
Default value is 'n'.

See also *NMR Spectroscopy User Guide*

Related htfrq1 Hadamard frequency list in ni (P)

htbw1 Hadamard pulse excitation bandwidth in ni (P)

Description The excitation bandwidth used to generate the frequencies contained in the shaped pulses used by the Hadamard matrix. If a single value is specified, the same bandwidth is used for all frequencies. If the parameter is arrayed, the bandwidth array element is used by the corresponding array element in htfrq1.

Values Default value is 20.0 if the parameter does not exist.

See also *NMR Spectroscopy User Guide*

Related htfrq1 Hadamard frequency list in ni (P)
ni Number of increments in 1st indirectly detected dimension (P)

htcal1 RF calibration flag for Hadamard waveforms in ni (P)

Description A flag to allow power optimization of Hadamard waveforms in the 1st indirect dimension.

Values 0 power optimization using htpwr1 is disallowed
>0 power optimization using htpwr1 is allowed
Default value is 0.

See also *NMR Spectroscopy User Guide*

Related htfrq1 Hadamard frequency list in ni (P)
htpwr1 Power level for rf calibration of Hadamard waveforms in ni (P)
ni Number of increments in 1st indirectly detected dimension (P)

htfrq1 Hadamard frequency list in ni (P)

Description A list of frequencies used in Hadamard spectroscopy, used for creating the Hadamard pulse shapes, and for placing the transformed traces at the correct frequencies in the indirect dimension.

Values Typical values are an arrayed set of frequencies between -sw1/2 and sw1/2.

See also	*NMR Spectroscopy User Guide*	
Related	htofs1	Hadamard offset in ni (P)
	fn1	Fourier number in 1st indirectly detected dimension (P)
	ni	Number of increments in 1st indirectly detected dimension (P)
	sethtfrq1	Set Hadamard frequency list from a line list (M)
	proc1	Type of processing on ni interferogram (P)
	sw1	Spectral width in 1st indirectly detected dimension (P)

htfrqdisp Read, write, and display Hadamard frequencies.

Syntax `htfrqdisp`

Applicability VnmrJ 3.1

Description The htfrqdisp macro is used by the Edit HT Freq dialog for setting and displaying Hadamard frequencies. It is not usually used from the command line.

See also editht
getht
ht
HsqcHT
tocsyHT
sethtfrq1
ft2d
ft2da

htofs1 Hadamard offset in ni (P)

Description The number of array elements to skip in ni when doing the Hadamard transform. The first element of the Hadamard matrix typically has all positive values (++++), and is usually not useful in constructing the Hadamard data.

Values Default value is 0. Typical values are 1 or 2.

See also	*NMR Spectroscopy User Guide*	
Related	htfrq1	Hadamard frequency list in ni (P)
	fn1	Fourier number in 1st indirectly detected dimension (P)
	ni	Number of increments in 1st indirectly detected dimension (P)
	ft2d	Fourier transform 2D data (C)
	proc1	Type of processing on ni interferogram (P)

htpwr1 — Power level for RF calibration of Hadamard waveforms in ni (P)

Description Power level for optimizing Hadamard waveforms in the 1st indirect dimension.

Values -16 to 63 dB in steps of 1 dB.

See also *NMR Spectroscopy User Guide*

Related

`htfrq1`	Hadamard frequency list in `ni` (P)
`htcal1`	RF calibration flag for Hadamard waveforms in ni (P)
`ni`	Number of increments in 1st indirectly detected dimension (P)

htss1 — Stepsize for Hadamard waveforms in `ni` (P)

Description Sets the stepsize during Hadamard waveform creation. Typically, this parameter is not needed, and a default stepsize is used.

Values Does not exist - default stepsize is used.
0 default stepsize is used.
>0 stepsize in microseconds.

See also *NMR Spectroscopy User Guide*

Related

`htfrq1`	Hadamard frequency list in `ni` (P)
`ni`	Number of increments in 1st indirectly detected dimension (P)

hzmm — Scaling factor for plots (P)

Description Contains the quotient of `wp` divided by `wc`, a scaling factor useful for plotting. `hzmm` applies to 1D only.

See also *NMR Spectroscopy User Guide*

Related

`wc`	Width of chart (P)
`wp`	Width of plot (P)

hztomm — Convert locations from Hz or ppm to plotter units (C)

Syntax
(1) `hztomm(x_position)<:xmm>`
(2) `hztomm(x_position,y_position)<:xmm,ymm>`
(3) `hztomm(<'box',><'plotter'|'graphics',>x_left, x_right,y_bottom,y_top)<:x1mm,x2mm,y1mm,y2mm>`

Description Converts locations from Hz, or ppm, to plotter units.

Arguments `x_position` in syntax 1 is a location along the 1D axis, in Hz or ppm, to be converted to plotter units using the current values of parameters `sp` and `wp`. Plotter units are mm on most plots and are scaled for graphics display. For ppm entries, use the `p` suffix following numerical values (see first example below).

`x_position,y_position` in syntax 2 is a coordinate, in Hz or ppm, on a 2D plot to be converted to plotter units, using the parameters `sp` and `wp` to convert the horizontal position and the parameters `sp1` and `wp1` to convert the vertical position.

`x_left,x_right,y_bottom,y_top` in syntax 3 are box edges, in Hz or ppm, on a 2D plot to be converted to plotter units, using the parameters `sp` and `wp` to convert the left and right edges, and parameters `sp1` and `wp1` to convert the top and bottom edges.

`'box'` is a keyword to draw a box and to make the first two return arguments, if supplied, give the location of the upper left corner of the box, in plotter units.

`'plotter'` is a keyword to select the plotter. The default is `'graphics'`.

`'graphics'` is a keyword to select the graphics screen. This is the default.

`x1mm,x2mm,y1mm,y2mm` are return arguments giving values in plotter units. If return arguments are not supplied, the results are displayed instead.

Examples
```
hztomm(20p)
hztomm(xpos,ypos):xmm,ymm
hztomm('box','plotter',20,50,10,30)
```

See also *NMR Spectroscopy User Guide*

Related

`box`	Draw a box on a plotter or graphics display (C)
`sp`	Start of plot in directly detected dimension (P)
`sp1`	Start of plot in 1st indirectly detected dimension (P)
`wp`	Width of plot in directly detected dimension (P)
`wp1`	Width of plot in 1st indirectly detected dimension (P)

I

i	Insert sample (M)
ihwinfo	Hardware status of console (U)
il	Interleave arrayed and 2D experiments (P)
ilfid	Interleave FIDs during data processing (C)
imagefile	Display an image file (M)
imagemath	Fit images to an specified function (M)
imageprint	Plot non interactive gray scale image (M)
imconi	Display 2D data in interactive grayscale mode (M)
import1Dspec	Import ASCII Spectrum into VnmrJ / VNMR (M)
in	Lock and spin interlock (P)
inadqt	Set up parameters for INADEQUATE pulse sequence (M)
index2	Projection or 3D plane index selected (P)
inept	Set up parameters for INEPT pulse sequence (M)
initialize_iterate	Set iterate string to contain relevant parameters (M)
input	Receive input from keyboard (C)
ins	Integral normalization scale (P)
ins2	2D volume value (P)
insref	Fourier number scaled value of an integral (P)
ins2ref	Fourier number scaled volume of a peak (P)
insert	Insert sample (M)
inset	Display an inset spectrum (C)
integ	Find largest integral in a specified region (C)
integrate	Automatically integrate 1D spectrum (M)
int_flg	determines whether dosy uses integrals or peak heights for DOSY fitting (P)
intmod	Integral display mode (P)

intvast	Produces a text file of integral regions (M)
intvast	Produce a text file containing the integral of the partial regions (M)
io	Integral offset (P)
is	Integral scale (P)
isadj	Automatic integral scale adjustment (M)
isadj2	Automatic integral scale adjustment by powers of two (M)
isreal	Utility macro to determine a parameter type (M)
isstring	Utility macro to determine a parameter type (M)
isvnmrj	Identifies the interface that is in use, either Vnmr or VnmrJ
iterate	Parameters to be iterated (P)

i — Insert sample (M)

Description: Turns off the eject air, waits for sample to slowly drop, and then turns off the slow drop air. The macro `insert` functions the same as `i`.

See also: *NMR Spectroscopy User Guide*

Related:
- `e` — Eject sample (M)
- `eject` — Eject sample (M)
- `insert` — Insert sample (M)

ihwinfo — Hardware status of console (U)

Syntax: (From UNIX) `ihwinfo('startup'|'abort')`

Description: Displays status of digital hardware in the console. The output is intended for service personnel and probably not meaningful to users.

Arguments: `'startup'` is a keyword to display the status at the conclusion of the last console startup (powerup, reboot, etc.).

`'abort'` is a keyword to display the status the last time an acquisition was aborted or the console rebooted from the host computer (`abortallacqs`). In this context, exiting from either the FID display or lock display of `acqi` counts as an abort. Only the status from the last abort can be displayed.

Examples:
```
ihwinfo('startup')
ihwinfo('abort')
```

See also *NMR Spectroscopy User Guide*

Related	abortallacqs	Reset acquisition computer in a drastic situation (C)
	showconsole	Show console configuration parameters (U)

il Interleave arrayed and 2D experiments (P)

Applicability Interleaving is not currently supported for the VNMRS or MR400 system.

ilfid Interleave FIDs during data processing (C)

Description Converts a multiple FID element into a single FID. It is possible to effectively extend the Nyquist frequency (i.e., increase the effective spectral width sw) by acquiring a number of FIDs with different *tau2* values and then reprocessing the data. ilfid does the necessary processing of time-domain data to achieve this extension, assuming that a pulse sequence (not supplied) has been written to generate the required data.

When invoked in an experiment of nf FIDs, each of np points, ilfid sorts the data into a single FID of np*nf points that can then be transformed. The interleaving takes the first complex point of each of the nf FIDs and places them in sequential order in the new FID. It then takes the second complex point from each of the nf FIDs and appends them sequentially to the new FID. This operation is repeated for all complex points. Although ilfid adjusts np and nf, it does not alter other parameters such as sw.

CAUTION Because ilfid alters the data irrevocably, it is strongly recommended that you save the FID before using ilfid.

Examples Illustrated below is the interleaving of an FID with nf=3 and np=4. Each point is represented by two digits. The first digit is the nf number and the second digit is the sequential point for that nf value. Data before the ilfid command:

```
11, 12, 13, 14; 21, 22, 23, 24; 31, 32, 33, 34
```

Data after the ilfid command:

```
11, 21, 31, 12, 22, 32, 13, 23, 33, 14, 24, 34
```

See also *NMR Spectroscopy User Guide*

Related	nf	Number of FIDs (P)
	np	Number of data points (P)
	sw	Spectral width in directly detected dimension (P)

imagefile Display an image file (M)

Applicability Imaging

Syntax `imagefile('output_option','imagefile'<,x,y,w,h,'mol'>)`

Description Display or plot an imagefile at default location and size or, optionally, at location and size specified by: `x` (x-position), `y` (y-position), `w` (width), `h` (`height`), and `mol` if it is an image file of a molecular structure. Display all, plot all, or clear all images for the current experiment.

Arguments `output_option` choices are:

`clear`, clear all images for the current experiment
`display`, display `imagefile`
`displayall`, displays all images for the current experiment
`plot`, plot `imagefile`
`plotall`, plot all images for the current experiment

`imagefile`, name of image file to display or plot

`x`, x position

`y`, y position

`w`, width

`h`, height

`mol` molecular structure image file

Examples `imagefile('clear')` clear all images for the current experiment.

`imagefile('displayall')` display all images for the current experiment.

imagemath Fit images to an specified function (M)

Applicability Imaging Systems

Syntax `imagemath(fit_type,fit_var,dir_flag)`

Description Calls standalone Linux program to fit data to the specified function (`fit_type`), either `T2`, or `DIFF` for a T2 map or diffusion calculation.

Data is fitted to a single exponential with the `ADC` or `T2` options. The output is given in two images:

A computed `S(0) image (filename S0)`

A map of either ADC or T2 (`filenameADC` or `filenameT2`).

The `diffcalc` linux program is invoked with the `DIFF` option. The output depends on the number of diffusion directions applied.

The argument `dir_flag` (if supplied) or the parameter `aipData` (if `dir_flag` is not supplied), determines where the program reads and writes data; if `aipData` or `dir_flag = 'saved'`, it uses the parameter `file` to determine the input directory (e.g., `sems_01.img`), and appends the name of the fit type to the directory name (e.g., `sems_01_ADC.img`) for the output directory; if `aipData` or `dir_flag`

= 'processed', it uses curexp/recon as the input directory and curexp/<fit_type> as the output directory. Calling imagemath from the Current viewport, using the current data, reads the data from/written to curexp.

See the *VnmrJ Imaging User's Guide* manual for information on the image math programs fdffit or diffcalc .

Arguments

fit_type 'ADC , 'T2', or 'DIFF'; default is 'ADC'

fit_var Name of the parameter that holds the independent variable.

Defaults to:
'bvalue' for ADC fit
'te' for T2 fit
blank string for DIFF fit

dir_flag optional string argument that mimics aipSave.
The macro imagemath looks at aipSave if no dirflag argument is given.

Examples

```
imagemath('ADC','bvalue','saved')
imagemath('T2','te')
imagemath('DIFF')
imagemath('DIFF','','saved')
```

See also *VnmrJ Imaging User's Guide*

imageprint Plot non interactive gray scale image (M)

Description Sends to the plotter a dcon color intensity map with linear instead of logarithmic increments and with grayscale instead of colors.

See also *NMR Spectroscopy User Guide*

Related dcon Display noninteractive color intensity map (C)

imconi Display 2D data in interactive grayscale mode (M)

Description Calls the dconi program with the arguments required for grayscale image display: dconi('dcon','gray','linear').

import1Dspec Import ASCII Spectrum into VnmrJ / VNMR (M)

Syntax import1Dspec('ascii_file'<,'av'>)

Applicability VnmrJ 3.1

Description "import1Dspec" imports a 1D ASCII spectrum into VnmrJ / VNMR. For the conversion, the ASCII file (Y .. Y or X,Y .. X,Y data,one Y value per line) is read in, and a UNIX utility with the same name ("bin/import1Dspec", a C program) is used to build the binary files "datdir/data" and "datdir/phasefile" in the current experiment. "vs" is set to 100.

Note that the imaginary part of the data set consists of zeroes only, i.e., the data cannot be phased - worse than that, setting "lp" to values other than 0 in "ph" mode would cause serious intensity distortions across the spectrum. To prevent such problems, "import1Dspec" offers two options:

- By default, the macro sets lp=0 rp=0 and sets the parameter protections bits such that these two parameters cannot be changed. Note that this may cause cause VnmrJ / VNMR to produce errors when the user (intentionally or inadvertently) attempts to change the phase within "ds".
- Alternatively, you can specify 'av' as second argument. In this case, the macro switches to "av" mode, and "dmg" is locked, such that the user cannot switch to "ph" mode. This has the disadvantage that negative parts of the spectrum are flipped to positive - but at least you will not have the usual problem with the line broadening (from the imaginary dispersion tails) that 'av' has with "normal" spectra. The (minor) advantage of this method is that "lp" and "rp" can be used to store the original "rp" and "lp" values.

If the current experiment contains nD or arrayed 1D data, "import1Dspec" first does a "setup('H1','CDCl3')", then imports the spectrum.

Limitation: Only works for simple, non-arrayed 1D spectra. The values of "rp" and "lp" are meaningless, the only meaningful parameter that is set from the imported data set is "fn". The other parameters (notably any applicable acquisition parameter, such as "sw", "tn", "sfrq", "nt", "ct", "np", "at", as well as possibly processing parameters such as weighting etc.) need to be set AFTER THE IMPORTING, along the following scheme:

```
$ct=ct  "remember 'ct'"
tn=.. sw=.. nt=.. at=.. ...
setvalue('ct',$ct)"restore 'ct'"
lb=.. sb=..
groupcopy('current','processed','acquisition')
groupcopy('current','processed','processing')
```

If you have a series of similar 1D ASCII spectra to process (e.g., a series of c13 spectra acquired with the same standard parameters) it is very helpful first to prepare such parameters according to the scheme above, then, for the importing, FIRST to retrieve these parameters, then to call "import1Dspec".

Arguments "**ascii_file**": Path to a pure ASCII file with either Y data (such as from "writetrace") or X,Y data (such as from "writexy"). The optional second argument 'av' selects and locks the spectrum in absolute value mode (see above); default is 'ph' mode.

Examples
import1Dspec('spectrum.xy')
import1Dspec('spectrum.xy','av')
import1Dspec('spectrum.txt')
import1Dspec('spectrum.txt','av')
import1Dspec('trace.1')
import1Dspec('trace.1','av')
import1Dspec('xytrace.1')
import1Dspec('xytrace.1','av')

Related

writetrace	Create ascii file from phasefile (f1 or f2) trace (M)
writexy	Create x,y ascii file from phasefile (f1 or f2) trace (M)
import1Dspec	Create phasefile and data from ASCII spectrum (U)

import1Dspec Create phasefile and data from ASCII spectrum (U)

Syntax `import1Dspec <-fn ##> <-vs #.#> ascii_file <phase_file <data>>`

Applicability VnmrJ 3.1

Description "import1Dspec" imports a 1D ASCII spectrum into VnmrJ / VNMR.

Arguments "**ascii_file**": Path to a pure ASCII file with either Y data (such as from "writetrace") or X,Y data (such as from "writexy"). "source/import1Dspec.c" is a C program that can be compiled with
cc -O -o /vnmr/bin/import1Dspec import1Dspec.c -m32
or (for a local installation)
cc -O -o ~/bin/import1Dspec import1Dspec.c -m32

"**phasefile**": Optional path to a binary "phasefile" that can afterwards be imported into VNMR / VnmrJ using "import1Dspec"; the default output file uses the same name as the ASCII file (with extensions ".txt" and ".xy" stripped off) but with ".phf" extension.

"**data**": Optional path to a binary "data" file that is required when importing "phasefile"; the default data file uses the same name as the phasefile, but with ".dat" extension.

"**-fn ##**": Optional, creates a phasefile with the specified number of points (fn/2 in VNMR!!!); should NOT be necessary, unless the ASCII file is somehow truncated; by default, "import1Dspec" will "zerofill" (add flat baseline at the high-field end) if the ASCII file does not contain a power of 2 in points; the argument following "-fn" MUST be numeric; if the specified number is NOT a power of 2, it will be rounded UP to the next higher power of 2. If the specified number or its next higher power of 2 are smaller than the number of points in the ASCII file, the spectrum is truncated at the high-field end.

"**-vs ##**": Optional, permits specifying a (down)scaling factor. When writing spectra in "ai" (absolute intensity) mode, "writetrace" writes out Y values in mm (spectrum multiplied by "vs"); specifying "-vs" with the value of "vs" from VNMR permits recreating the original ("ai") spectrum. Also here, the argument following "-vs" MUST be numeric and positive. The default "vs" value (downscaling factor) is 1.0.

Examples
import1Dspec spectrum.txt
import1Dspec spectrum.xy
import1Dspec spectrum.txt phasefile
import1Dspec spectrum.txt phasefile data
import1Dspec spectrum.xy phasefile data
import1Dspec -fn 64000 spectrum.xy datdir/phasefile datdir/data
import1Dspec -vs 327.54 spectrum.xy datdir/phasefile datdir/data
import1Dspec -vs 327.54 -fn 32000 spectrum.xy

Related
`import1Dspec` Import ASCII spectrum into VnmrJ / VNMR (M)
`writetrace` Create ascii file from phasefile (f1 or f2) trace (M)
`writexy` Create x,y ascii file from phasefile (f1 or f2) trace (M)

`in` Lock and spin interlock (P)

Description
Controls error handling based on lock level and spin speed, and specifies action based on lock level failure or spinner failure. The action can be to generate an error and halt acquisition, or to generate a warning and continue acquisition.

Values
Can be set to one or two characters:

- If set to two characters, the first character specifies the action for lock failure and the second character specifies the action for spinner failure.
- If set to only one character, that character specifies the same action for either lock or spinner failure.

`'n'` stops any system checking so that acquisition continues regardless of the lock level or spin speed.

`'w'` makes the system check the lock level and the spin speed. A warning message is added to the log file if the lock level falls below a preset hardware level (about 20 on the lock meter) or if `spin` is set to a particular value and the spin speed goes out of regulation; however, acquisition is not stopped.

`'y'` makes the system check the lock level and spin speed. Acquisition is halted if the lock level falls below a preset hardware level (about 20 on the lock meter) or if `spin` is set to a particular value and the spin speed goes out of regulation.

See also
NMR Spectroscopy User Guide

Related
`spin` Sample spin rate (P)

inadqt Set up parameters for INADEQUATE pulse sequence (M)

Description Sets up parameters for 2D INADEQUATE (Incredible Natural Abundance Double-Quantum Transfer Experiment).

See also *NMR Spectroscopy User Guide*

Related `foldcc` Fold INADEQUATE data about 2-quantum axis (C)

index2 Projection or 3D plane index selected (P)

Description Stores whether a projection or 3D plane index is selected. It shows the current status only and cannot be used to select a plane or projection. This parameter is also displayed in the Status window below "Index."

Values 0 indicates a projection is selected.

1 to the half the Fourier number of the normal axis indicates a 3D plane is selected; the number is the index of the 3D plane.

See also *NMR Spectroscopy User Guide*

Related
`dplane` Display a 3D plane (M)
`dproj` Display a 3D plane projection (M)
`nextpl` Display the next 3D plane (M)
`prevpl` Display the previous 3D plane (M)
`select` Select a spectrum or 2D plane without displaying it (C)

inept Set up parameters for INEPT pulse sequence (M)

Description Sets up parameters for the INEPT (Insensitive Nuclei Enhanced by Polarization Transfer) experiment.

See also *NMR Spectroscopy User Guide*

Related `ppcal` Proton decoupler pulse calibration (M)

initialize_iterate Set iterate string to contain relevant parameters (M)

Description Takes the current spin system (contained in `spinsys`) and derives from it relevant parameters. This can be used to control which parameters are iterated during a spin simulation iteration (e.g., for an ABC spin system, `iterate` is set to `'A,JAB,JAC,B,JBC,C'`).

See also *NMR Spectroscopy User Guide*

Related `iterate` Parameters to be iterated (P)

input — Receive input from keyboard (C)

Syntax `input<(<prompt><,delimiter>)>:var1,var2,...`

Description Receives fields of characters from the keyboard and stores them into one or more variables.

Arguments `prompt` is a string displayed on the command line.

`delimiter` is a character separating input fields. The default is a comma.

`var1,var2,...` are return values. `input` stores the values into as many of these arguments as given and ignores the rest of the input line.

Examples

```
input:$b
input('Enter pulse width:'):pw
input('x and y coordinates'):cr,cr1
input('Enter lastname:firstname',':'):$last,$first
```

See also *User Programming*

Related

`string`	Create a string variable (C)

ins — Integral normalization scale (P)

Description Sets the integral value, independent of `is` and `vs`. Reported integral values are scaled by `fn`; that is, the reported integral of a given region is independent of `fn`. The `insref` parameter is also used to determine a reference integral value. The `setint` macro sets integral value.

See also *NMR Spectroscopy User Guide*

Related

`dlni`	Display list of normalized integrals (M)
`fn`	Fourier number in directly detected dimension (P)
`is`	Integral scale (P)
`insref`	Fourier number scaled value of an integral (P)
`mark`	Determine intensity of spectrum at a point (C)
`setint`	Set value of an integral (M)
`vs`	Vertical scale (P)

ins2 — 2D volume value (P)

Description Adjusts the 2D volume value, independent of `is` and `vs`. The volume is scaled by Fourier numbers for the two dimensions.

See also *NMR Spectroscopy User Guide*

Related

`is`	Integral scale (P
`ins2ref`	Fourier number scaled volume of a peak (P)
`ll2d`	Automatic and interactive 2D peak peaking (C)
`vs`	Vertical scale (P)

insref — Fourier number scaled value of an integral (P)

Description: Set to the Fourier number scaled value of a selected integral. The reported integral values will be *(integral value)**ins/insref/fn. If insref is "not used", the sum of all integrals will be ins. The "not used" mode is the equivalent of the normalized integral mode. If insref is zero or not defined, the reported integrals will be *(integral value)**ins/fn.

See also: *NMR Spectroscopy User Guide*

Related:

fn	Fourier number in directly detected dimension (P)
ins	Integral normalization scale (P)
liamp	Amplitudes of integral reset points (P)
setint	Set value of an integral (M)

ins2ref — Fourier number scaled volume of a peak (P)

Description: Set to the Fourier number scaled volume of the selected peak. The reported volume is *volume**ins2/ins2ref/fn/fn1. If ins2ref is "not used", sum of all volumes is ins2. The "not used" mode is equivalent to a normalized volume mode. If ins2ref is zero or not defined, the reported volume is *volume**ins2/fn/fn1.

See also: *NMR Spectroscopy User Guide*

Related:

fn	Fourier number in directly detected dimension (P)
fn1	Fourier number in first indirectly detected dimension (P)
ins2	2D volume value (P)
ll2d	Automation and interactive 2D peak picking (C)

insert — Insert sample (M)

Description: Turns off the eject air, waits for the sample to slowly drop, and then turns off the slow drop air. The macro i is identical in function to insert.

See also: *NMR Spectroscopy User Guide*

Related:

e	Eject sample (M)
eject	Eject sample (M)
i	Insert sample (M)

inset Display an inset spectrum (C)

Description Displays the part of the spectrum between the two cursors as an inset. Before entering inset, run the ds command and display two cursors. The vertical position is shifted up about one-quarter of the height of the whole display canvas. The old spectrum remains on the screen, but the parameters shown at the bottom are relevant to the new display. If present, the integral trace is duplicated. The scale is also duplicated if it is present. After running inset, you can shift the displayed spectrum, expand it, or even contract it with the left and right mouse buttons.

See also *NMR Spectroscopy User Guide*

Related ds Display a spectrum FID (C)

integ Find largest integral in a specified region (C)

Syntax `integ<(highfield,lowfield)><:size,value>`

Description Finds the largest absolute-value integral in the specified region, or the total integral if no reset points are present between the specified limits.

Arguments highfield and lowfield are the limits of the region. The default values are the parameters sp and sp+wp, respectively.

size is a return value with the size of the largest integral. The size depends on the value of the parameter is and can be positive or negative.

value is a return argument with the value of the largest integral. This value depends on ins, insref, and fn, and is independent of is.

Examples
```
integ:r1,r2
integ(500,1000):$height
integ(100+sp,300+sp):$ht,$val
```

See also *User Programming*

Related

fn	Fourier number in directly detected dimension (P)
ins	Integral normalization scale (P)
insref	Fourier number scaled value of an integral (P)
is	Integral scale (P)
rp	Zero-order phase in directly detected dimension (P)
sp	Start of plot in directly detected dimension (P)
wp	Width of plot in directly detected dimension (P)

integrate Automatically integrate 1D spectrum (M)

Description A universal macro for selecting integral regions and adjusting the integrals in size and offset. Only if regions are not already selected, and if intmod is set to 'partial', will integrate call region to select integral regions. For proton spectra, the selection is done through the hregions macro; for ^{19}F and ^{31}P spectra (for wide spectral windows, multiplet spectra), region is called with optimized arguments, and for other nuclei (mostly decoupled, single-line spectra) other optimized parameters are used with region, such that lines consisting of a few data points only are recognized.

See also *NMR Spectroscopy User Guide*

Related	hregions	Select integral regions in proton spectrum (M)
	intmod	Integral display mode (P)
	isadj	Adjust integral scale (M)
	region	Automatically select integral regions (C)

int_flg

Syntax `int_flg`

Applicability VnmrJ 3.1

Description `int_flg` determines whether dosy uses integrals or peak heights for DOSY fitting. int_flg='y' requires that valid integral resets be defined.

Arguments in_flg='y' invokes fitting of peak integrals

in_flg='n' invokes fitting of peak heights

See also dosy

intmod Integral display mode (P)

Description Controls display and plotting of the spectral integral.

Values `'off'` indicates that no integrals are displayed or plotted.

`'full'` indicates that all integral regions are displayed or plotted.

`'partial'` indicates that every other integral region is plotted (typically used to display integrals of only peaks and not of the baseline region).

See also *NMR Spectroscopy User Guide*

Related	plc	Plot carbon spectrum (M)
	plh	Plot proton spectrum (M)
	plp	Plot phosphorus spectrum (M)

intvast Produce a text file of integral regions (M)

Applicability Systems with VAST accessory.

Syntax `intvast (last)`

Description `intvast` produces a text file, integ.out in the current experiment, containing the integrals of the partial regions of each spectra from wells 0 to `last`.

Arguments `last` is the number last sample well. The default is 96.

See also *NMR Spectroscopy User Guide*

Related `pintvast` Plot the integrals (M)

`intvast` Produce a text file containing the integral of the partial regions

Applicability VnmrJ 3.1

Description The intvast macro produce a text file containing the integral of the partial regions. The integral regions of the spectra need to be preset. The resulting file, called integ.out, is placed in the local experiment directory.

Examples intvast(<number of wells of data>)

See also
`dsvast`
`dsvast2d`
`plvast`
`plvast2d`
`intvast`
`pintvast`
`plateglue`
`vastglue`
`vastget`

`io` Integral offset (P)

Description Offset of the integral with respect to the spectrum.

Values 0 to 200, in mm.

See also *NMR Spectroscopy User Guide*

`is` Integral scale (P)

Description Multiplier that adjusts height of the displayed integral trace. Note that the `ins` parameter controls integral value, and that `is` has no effect on integral value.

Values 1 to 1e9

See also *NMR Spectroscopy User Guide*

Related	
ins	Integral normalization scale (P)
ins2	2D volume value (P)
insref	Fourier number scaled value of an integral (P)
integ	Find largest integral in a specified region (C)

isadj Automatic integral scale adjustment (M)

Syntax `isadj<(height<,neg_height>)>`

Description Adjusts the height of the integrals in a display to make the tallest integral fit the paper. Optionally, the height of the maximum integral can be specified by an argument. Negative integrals, if present, are given a limit of 10 mm if parameter io is less than 10; otherwise, they are set so they end 5 mm above the spectrum. Negative integrals can also be given a height. Whichever part of the integrals (positive or negative) runs into the given limit will be used to scale is.

Arguments `height` is the size, in mm, of the maximum integral on display. The default is the height that makes the tallest integral fit the paper.

`neg_height` is the desired height, in mm, of the largest negative integral. If io is less than 10, the default is 10; otherwise, the default height is 5 mm above the spectrum.

Examples
```
isadj
isadj(100)
isadj(100,100)
```

See also *NMR Spectroscopy User Guide*

Related	
io	Integral offset (P)
is	Integral scale (P)
isadj2	Automatic integral scale adjustment by powers of two (M)

isadj2 Automatic integral scale adjustment by powers of two (M)

Syntax `isadj2<(height<,neg_height>)>:scaling_factor`

Description Functionally the same as isadj except that `isadj2` adjusts the integral height by powers of two and returns the scaling factor to the calling macro.

Arguments `height` is the size, in mm, of the maximum integral on display.

`neg_height` is the desired height, in mm, of the maximum negative integral on display.

`scaling_factor` is a return value giving the ratio of the new integral size to the old value (`new_is/old_is`).

Examples
```
isadj2
isadj2(100)
```

```
isadj2(100,100)
isadj2(50):r1
```

See also *NMR Spectroscopy User Guide*

Related `is` Integral scale (P)
`isadj` Automatic integral scale adjustment (M)

isreal Utility macro to determine a parameter type (M)

Syntax `isreal(paramname<,tree>)`

Description Returns 1 if and only if `paramname` is a real type. It returns 0 if paramname is a string type. If there is an error, the error is reported and the macro also returns 0. The value of tree is `'current'`, `'global'`, `'processed'` or `'systemglobal'` and the default is `'current'`.

There is some unfortunate ambiguity and vagueness in regard to vnmr parameters and their types. The meaning of `real` and `string` vary slightly depending upon context. There are seven types altogether. The macro `gettype` returns a unique integer value when operating on the parameter. Of the seven types, two can be broadly categorized as string, and five can be broadly categorized as real. Since one of the string category types is '`string`' and one of the real category types is '`real`', this is where the ambiguity arises. The return values for `gettype` are:

category	*type*	*gettype returns*
string	`'string'`	2
	`'flag'`	4
real	`'real'`	1
	`'delay'`	3
	`'frequency'`	5
	`'pulse'`	6
	`'integer'`	7

The `isreal` function returns 0 for the string category and 1 for the real category. This function is consistent with the `typeof()` operator. The `typeof()` operator is primarily intended to ascertain the type of the input argument to a macro, so using it for other purposes is not recommended. Also, it does not take a tree argument. Note that `typeof()` returns 0 for reals and 1 for strings, the opposite of this macro, but it should be clear from the name what is intended. A sister macro `isstring` returns the same value as `typeof()`.

Related `isstring` Utility macro to determine a parameter type (M)
`typeof` Return identifier for argument type (O)

isstring — Utility macro to determine a parameter type (M)

Syntax `isstring(paramname<,tree>)`

Description Returns 1 if and only if paramname is a string type. It returns 0 if paramname is a real type. If there is an error, the error is reported and the macro also returns 0. The value of tree is `'current'`, `'global'`, `'processed'` or `'systemglobal'` and the default is `'current'`.

There is some unfortunate ambiguity and vagueness in regard to vnmr parameters and their types. The meaning of `real` and `string` vary slightly depending upon context. There are seven types altogether. The macro `gettype` returns a unique integer value when operating on the parameter. Of the seven types, two can be broadly categorized as string, and five can be broadly categorized as real. Since one of the string category types is '`string`' and one of the real category types is '`real`', this is where the ambiguity arises. The return values for `gettype` are:

category	*type*	*gettype returns*
string	`'string'`	2
	`'flag'`	4
real	`'real'`	1
	`'delay'`	3
	`'frequency'`	5
	`'pulse'`	6
	`'integer'`	7

The function `isstring` returns 0 for the real category and 1 for the string category. This function is consistent with the `typeof()` operator. The `typeof()` operator is primarily intended to ascertain the type of the input argument to a macro, so using it for other purposes is not recommended. Also, it does not take a tree argument. Note that `typeof()` returns 0 for reals and 1 for strings, the opposite of this macro, but it should be clear from the name what is intended. A sister macro `isstring` returns the same value as `typeof()`.

Related

`isreal`	Utility macro to determine a parameter type (M)
`typeof`	Return identifier for argument type (O)

isvnmrj — Identifies the interface is use, either Vnmr or VnmrJ

Syntax `isvnmrj:$val`

Applicability VnmrJ 3.1

Description The `isvnmrj` command identifies which interface is in use, either vnmr or vnmrj. This command would typically only be used in macros.

Arguments The command returns a 1 if the interface is vnmrj, otherwise it returns a 0.

Examples isvnmrj:$ans

`iterate` Parameters to be iterated (P)

Description Contains parameters to be iterated during iterative spin simulations. If the Set Params button is used in setting up spin simulation parameters, `iterate` is initialized to a string containing all parameters appropriate to the current spin system.

Values List of parameters, separated by commas (e.g., `iterate='A,B,JAB'`).

See also *NMR Spectroscopy User Guide*

Related `initialize_iterate` Set `iterate` string to contain relevant parameters (M)

J

jaddsub	Join the add/subtract experiment
jcurwin	Work space numbers of all viewports (P)
jdesign	Start Plot Designer Program (M)
jexp	Join existing experiment (C)
jexp1–jexp9999	Join existing experiment and display new parameters (M)
jexpn	Join experiment n, where n is a number between 1 and 9
jnewexp	Experiment-Specific Runtime Macro
jplot	Plot from Plot Designer program (C)
jplotscale	Scale plot parameters (M)
jplotunscale	Restore current experiment parameters (M)
jprint	Prints the selected images to a printer or file (M)
jpublish	Macro to archive and/or copy to system a local protocol (M)
jumpret	Set up parameters for JUMPRET pulse sequence (M)
jviewport	Work space numbers of the current viewports (P)
jviewportlabel	Work space labels for all viewport buttons (P)
jviewports	Viewport layout (P)
jwin	Activate and record activity in current window (M)

jaddsub Join the add/subtract experiment

Applicability VnmrJ 3.1

Description jaddsub joins the add-subtract experiment, as defined by the global addsubexp parameter. jaddsub will create this parameter if it does not exist, and set it to a default value of 5. jaddsub with an argument, as in jaddsub('silent') will not clear the graphics, text window, or menu system. It does not matter what the argument is.
The last displayed or selected FID is added to ("add") or subtracted from ("sub") the current contents of the add/subtract experiment. Am

optional argument allows the FID to be first multiplied by a 'multiplier'. The FID data are divided by the number of time averages of the data, reflected in the parameter ct. To get unscaled data, use a multiplier of ct. The parameters lsfid and phfid may be used to shift or phase rotate the selected FID before it is combined with the data in the add/subtract experiment.

A multi-fid add/subtract experiment may be created with the `add` or `sub` command. The optional argument 'new' will create a new FID element in the add/subtract experiment. For example, the commands `clradd` `select(1)` add from some experiment will create the add/subtract experiment with a single FID in it. If the next commands typed are `select(2)add`, then a single FID which is the sum of the original FIDs one and two will be made in the add/subtract experiment. If, on the other hand, the commands `select(2)add ('new')` were typed, then the add/subtract experiment will contain an array of two FIDs corresponding to the original FIDs one and two, espectively. One detail is that the arraydim parameter may need to be updated after constructing a multi-fid add/subtract experiment. The recipe for doing this is to join the add/subtract experiment (`jaddsub`) and enter setvalue('`arraydim`',<num>,'processed') where <num> is replaced by the number of FIDs in that experiment. For example, if twelve FIDs were put into the add/subtract experiment, one would enter setvalue(`'arraydim',12,'processed'`) Individual FIDs in a multi-fid add/subtract experiment may subsequently be added to and subtracted from. The `add` and `sub` command without a 'trace' argument will add or subtract from the first FID in the add/subtract experiment. Adding the '`trace`' argument followed by a required index number will select another FID to be the target of the add/subtract. For example, select(4) `add`('trace',6) will take the fourth FID from the current experiment and add it to the sixth FID in the add/subtract experiment. When using the 'trace' argument, that FID must already exist in the add/subtract experiment by using an appropriate number of `add('new')` or `sub('new')` commands.

Arguments silent
new

jcurwin — Work space numbers of all viewports (P)

Description An arrayed global parameter, set to the work space numbers used by all viewports.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related		
	`curwin`	Current window (P)
	`jviewport`	Work space numbers of the current viewports (P)
	`jviewportlabel`	Work space labels for all viewport buttons (P)

jdesign Start Plot Designer Program (M)

Syntax `jdesign`

Description Opens the Plot Designer program, which provides mechanisms for positioning spectra, parameters, axes, and other plot output on a page. Text annotation and drawing features are available.

See also *NMR Spectroscopy User Guide*

Related `jplot` Plot from Plot Designer program (C)

jexp Join existing experiment (C)

Syntax (1) `jexp(exp_number)`
(2) `jexp:$current_exp_number,$current_exp_name`

Description Joins an existing experiment (syntax 1) or returns the current experiment number and experiment name (syntax 2). After entering this command, until another "join experiment" command or macro is entered, all actions (including changes of parameters, acquisition of data, and display of data) apply to the parameters and data of the experiment joined.

The `jexp` command does not refresh the display or display new experiment parameters. Use one of the macros `jexp1`, `jexp2`, etc. to join an experiment and have the screen refreshed and new parameters displayed.

Arguments `exp_number` is a number from 1 to 9999 for existing experiment to be joined.

`$current_exp_number` is a return value with the current experiment number.

`$current_exp_name` is a return value with the current experiment name.

Examples
```
jexp(3)
jexp:$expp
jexp:r1,n1
```

See also *NMR Spectroscopy User Guide; VnmrJ Walkup*

Related

`cexp`	Create an experiment (M)
`delexp`	Delete an experiment (M)
`jexp1–jexp9`	Join existing experiment and display new parameters (M)
`unlock`	Remove inactive lock and join experiment (C)

jexp1–jexp9999 Join existing experiment and display new parameters (M)

Syntax `jexp1`, `jexp2`, `jexp3`, ...,`jexp9999`

Description Joins an existing experiment, refreshes the screen, and displays the main menu and the new experiment parameters. After entering this macro, until another "join experiment" command or macro is entered, all actions (including changes of parameters, acquisition of data, and display of data) apply to the parameters and data of the experiment joined.

To join an experiment without refreshing the screen and displaying new parameters, use the `jexp` command.

Examples `jexp8`
`jexp354`

See also *NMR Spectroscopy User Guide*

Related	`cexp`	Create an experiment (M)
	`delexp`	Delete an experiment (M)
	`jexp`	Join existing experiment (C)
	`unlock`	Remove inactive lock and join experiment (C)

jexpn Join experiment n, where n is a number between 1 and 9

Syntax `jexpn`

Applicability VnmrJ 3.1

Description Join experiment `n`, where n is a number from 1 to 9 describing an existing experiment. After this command, all actions including changes of parameters, acquisition of data, display of data, etc. will apply to the parameter and data of experiment `n`, until the next "`jexp`" command is executed.
Without an argument, `jexp:$num` returns the current experiment number in the variable `$num` and `jexp:$num,$name` returns both the experiment number and experiment name to the variables `$num` and `$name`.

jnewexp Creates and Joins a New Experiment

Description Creates and joins a new experiment.

Syntax `jnewexp`

jplot — Plot from Plot Designer program (C)

Syntax `jplot<(<'-setup'><,template)>`

Description Starts plotting from the Plot Designer program to the current plotter.

Arguments `'-setup'` is a keyword to start `jdesign`, the Plot Designer program, to allow interactive design and plotting.

`template` is the name of a file that will be used to make a plot of the current experiment. The default is a saved file chosen by the user.

Examples
```
jplot
jplot('t1')
```

See also *NMR Spectroscopy User Guide*

Related

`jdesign`	Start Plot Designer program (M)
`jplotscale`	Scale plot parameters (M)
`jplotunscale`	Restore current experiment parameters (M)

jplotscale — Scale plot parameters (M)

Applicability Plot Designer program

Description Scales parameters of plotting area and an imported plot. When a region is drawn in Plot Designer, `jplotscale` automatically changes the plotting area parameters `wcmax` and `wc2max`. The parameters `io`, `is`, `vs`, `wc`, and `wc2` of a plot imported into a region are adjusted according to `wcmax` and `wc2max`.

See also *NMR Spectroscopy User Guide*

Related

`jplot`	Plot from Plot Designer program (C)
`jplotunscale`	Restore current experiment parameters (M)

jplotunscale — Restore current experiment parameters (M)

Applicability Plot Designer program

Description Restores the current experiment parameters (`io`, `is`, `vs`, `wc`, and `wc2`) to a plot within a region that was created in Plot Designer. For example, entering `jplotunscale jexp2 jplotscale` restores the parameters of experiment 2 to a plot and then `jplotscale` applies the adjusted parameters to the plot.

See also *NMR Spectroscopy User Guide*

Related

`jplot`	Plot from Plot Designer program (C)
`jplotscale`	Scale plot parameters (M)

jprint Prints the selected images to a printer or file (M)

Description The jprint macro takes the value of the parameters printregion, printsend, printfile, printlayout, printformat, printsize.

jpublish Macro to archive and/or copy to system a local protocol (M)

Syntax jpublish

Applicability VnmrJ 3.1

Description Assembles all elements of the experiment protocol in the current experiment into a archive protocolname_proto.tar.Z into a User_Protocols directory of the local vnmrsys directory. If the user is the VnmrJ admin, the protocol is installed into the VnmrJ system for all users. Note that the file protocolListWalkup.xml in /vnmr/adm/users would need to be edited to add the new protocol to the experiment chooser in a walkup-style account. The user is prompted for all needed information and the addition of one additional setup support macro is allowed. By default, jpublish will assign the local protocol name and seqfil as being associated. This is almost always the case.

jumpret Set up parameters for JUMPRET pulse sequence (M)

Description Sets up parameters for a jump-and-return water suppression sequence.

See also *NMR Spectroscopy User Guide*

jviewport Work space numbers of the current viewports (P)

Description A global parameter, set to the work space number that the current viewport is joined to. The parameter is set when the viewport starts. Each viewport may be joined to a different work space.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

curwin	Current window (P)
jcurwin	Work space numbers of all viewports (P)
jviewports	Viewport layout (P)
jviewportlabel	Work space labels for all viewport buttons (P)

`jviewportlabel` Work space labels for all viewport buttons (P)

Description	An arrayed global parameter, set to the labels on the toolbar buttons used to switch viewports. It is used by the viewport editor under **Edit -> Viewports**.
See also	*NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

`jviewport`	Work space numbers of the current viewports (P)
`jviewports`	Viewport layout (P)
`vpaction`	Set initial state for multiple viewports (M)

`jviewports` Viewport layout (P)

Description	An arrayed global parameter, used to keep track of the viewport layout. It is used by the viewport editor under **Edit -> Viewports** to change the viewport layout.

Related

`jcurwin`	Work space numbers of all viewports (P)
`jviewport`	Work space numbers of the current viewports (P)
`jviewportlabel`	Work space labels for all viewport buttons (P)
`vpaction`	Set initial state for multiple viewports (M)
`vpset3def`	Set the viewport state to three default viewports (M)
`vpsetup`	Set new viewports (M)

`jwin` Activate and record activity in current window (M)

Syntax	`jwin(pane_number)`
Description	Activates and records the activity in a specific window pane, created by `setgrid`, in the VnmrJ graphics window. `jwin` is executed when you double-click the left mouse button in a multiple-paned graphics window.
Arguments	`pane_number` is the number of the pane to join.
Examples	`jwin(2)`
See also	*NMR Spectroscopy User Guide*

Related

`curwin`	Current window (P)
`fontselect`	Open FontSelect window (C)
`mapwin`	List of experiment numbers (P)
`setgrid`	Activate selected window (M)
`setwin`	Activate selected window (C)

K

killft3d	Terminate any ft3d process started in an experiment (M,U)
killplot	Stop plot jobs and remove from plot queue (M)
killprint	Stop print jobs and remove from print queue (M)
kind	Kinetics analysis, decreasing intensity (M)
kinds	Kinetics analysis, decreasing intensity, short form (M)
kini	Kinetics analysis, increasing intensity (M)
kinis	Kinetics analysis, increasing intensity, short form (M)

killft3d Terminate any ft3d process started in an experiment (M,U)

Syntax `killft3d(exp_number)`

Description Terminates any ft3d program that has been started in the specified VnmrJ experiment. `killft3d` can be executed from any experiment. For each ft3d process terminated, the relevant 3D data subdirectory is also deleted. Remote ft3d processes, denoted by the call name `ftr3d` in the process table (displayed by the UNIX command `ps -azx`), are not directly terminated by `killft3d` but die of their own accord due to the deletion of the 3D data subdirectory.

The `killft3d` command can also be run as a shellscript from UNIX. Its function is analogous to the associated VnmrJ macro.

Arguments `exp_number` is a number from 1 to 9 that identifies the experiment that started the ft3d program.

Examples `killft3d(4)`

See also *NMR Spectroscopy User Guide*

Related ft3d Perform a 3D Fourier transform (M,U)

killplot Stop plot jobs and remove from plot queue (M)

Description Kills all current plot jobs in the plot queue for the active plotter in VnmrJ, then removes the jobs from the plot queue. Unless the user executing `killplot` is `root`, only that user's plot jobs are deleted from the plot queue. To kill a plot that is in progress (i.e., a plot in which you have not entered `page`), use the `page('clear')` command.

The plotter may have to be reinitialized after `killplot` is executed. To reinitialize the plotter, turn it off and then back on after a few seconds. Hewlett-Packard (HP) pen plotters appear to be more susceptible to this problem than the other HP output devices supported by VnmrJ.

If one port is configured to be both a printer and a plotter, `killplot` can cause both plot *and* print jobs to that port to be deleted. For example, if `printer='LaserJet_300'`, `plotter='LaserJet_300R'`, and a plot command `pl pscale page` is followed by a print command `ptext(vnmruser+'/psglib/noesy.c')`, entering `killplot` deletes both jobs.

See also *NMR Spectroscopy User Guide*

Related

`killprint`	Stop print jobs and remove from print queue (M)
`page`	Move plotter forward one or more pages (C)
`pl`	Plot spectra (C)
`pscale`	Plot scale below spectrum or FID (C)
`ptext`	Print out a text file (M)
`showplotq`	Display plot jobs in plot queue (M)

killprint Stop print jobs and remove from print queue (M)

Description Kills all current print jobs in the print queue for the active printer in VnmrJ, then removes the jobs from the print queue. Unless the user executing `killprint` is `root`, only that user's print job is deleted from the print queue. It is slightly possible that the printer may have to be reinitialized after the execution of this macro. To reinitalize the printer, turn it off, wait a few seconds, and then turn it back on.

If one port is configured to be both a printer and a plotter, `killprint` can cause both print *and* plot jobs to that port to be deleted. For example, if `printer='LaserJet_300'`, `plotter='LaserJet_300R'`, and a plot command `pl pscale page` is followed by a print command `ptext(vnmruser+'/psglib/noesy.c')`, entering `killprint` deletes both jobs.

See also *NMR Spectroscopy User Guide*

Related

`killplot`	Stop plot jobs and remove from plot queue (M)

ptext	Print out a text file (M)
showprintq	Display print jobs in print queue (M)

kind Kinetics analysis, decreasing intensity (M)

Description If the signal decreases exponentially toward a limit, the output is matched by *I = A1 * EXP(–T/TAU) + A3*. This macro supplies the necessary keywords to the analyze command, which uses the output of fp (i.e., the file fp.out) as input. The results can be displayed with expl.

See also *NMR Spectroscopy User Guide*

Related

analyze	Generalized curve fitting (C)
expl	Display exponential/polynomial curves (C)
fp	Find peak heights (C)
kinds	Kinetic analysis, decreasing intensity, short form (M)
kini	Kinetics analysis, increasing intensity (M)
kinis	Kinetic analysis, increasing intensity, short form (M)

kinds Kinetics analysis, decreasing intensity, short form (M)

Description Produces a summary of the results from kind.

See also *NMR Spectroscopy User Guide*

Related

kind	Kinetics analysis, decreasing intensity (M)

kini Kinetics analysis, increasing intensity (M)

Description If the signal increases exponentially toward a limit, the output is matched by
*I = –A1 * EXP(–T/TAU) + A3 – A1*. This macro supplies the necessary keywords to the analyze command, which uses the output of fp (i.e., the file fp.out) as input. The results can be displayed with expl.

See also *NMR Spectroscopy User Guide*

Related

kind	Kinetics analysis, decreasing intensity (M)
kinis	Kinetic analysis, increasing intensity, short form (M)

kinis Kinetics analysis, increasing intensity, short form (M)

Description Produces a summary of the results from kini.

See also *NMR Spectroscopy User Guide*

Related kind Kinetics analysis, decreasing intensity (M)
kini Kinetics analysis, increasing intensity (M)

L

laser	SVS adiabatic localization
lastlk	Last lock solvent used (P)
lastmenu	Menu to display when Return button is selected (P)
latch	Frequency synthesizer latching (P)
lb	Line broadening in directly detected dimension (P)
lb1	Line broadening in 1st indirectly detected dimension (P)
lb2	Line broadening in 2nd indirectly detected dimension (P)
lc1d	Pulse sequence for LC-NMR (M)
lcdatast	An LC-NMR plotting and display macro (M)
lcpar2d	Create 2D LC-NMR acquisition parameters (M)
lcpeak	Peak number (P)
lcplot	Plot LC-NMR data (M)
lcscale	An LC-NMR plotting macro (M)
lcpsgset	Set up parameters for various LC-NMR pulse sequences (M)
lcscale	An LC-NMR plotting macro
lcset2d	General setup for 2D LC-NMR experiments (M)
left	Set display limits to left half of screen (C)
legrelay	Independent control of magnet leg relay (P)
length	Determine length of a string (C)
lf	List files in directory (C)
lgcp	X Lee-Goldburg cross polarization (CP) between protons and X with a choice of SPINAL or TPPM decoupling
liamp	Amplitudes of integral reset points (P)
lifrq	Frequencies of integral reset points (P)
liqbear	Liquids Bearing Air Level (P)
listenoff	Disable receipt of messages from send2Vnmr (M)

listenon	Enable receipt of messages from send2Vnmr (M)
listparam	List parameters in simple format (UNIX)
lkof	Track changes in lock frequency (P)
ll2d	Automatic and interactive 2D peak picking (C)
ll2dbackup	Copy current ll2d peak file to another file (M)
ll2dmode	Control display of peaks picked by ll2d (P)
llamp	List of line amplitudes (P)
llfrq	List of line frequencies (P)
ln	Find natural logarithm of a number (C)
load	Load status of displayed shims (P)
loadcolors	Load colors for graphics window and plotters (M)
loaduserprefs	Load Operator Preferences
loc	Location of sample in tray (P)
locaction	Locator action (M)
lock	Submit an Autolock experiment to acquisition (C)
lockacqtc	Lock loop time constant during acquisition (P)
lockfreq	Lock frequency (P)
lockgain	Lock gain (P)
lockphase	Lock phase (P)
lockpower	Lock power (P)
locktc	Lock time constant (P)
log	
logate	Transmitter local oscillator gate (P)
lookup	Look up words and lines from a text file (C)
locprotoexec	Execute a protocol from the locator (M)
lp	First-order phase in directly detected dimension (P)
lp1	First-order phase in 1st indirectly detected dimension (P)
lp2	First-order phase in 2nd indirectly detected dimension (P)
lpalg	LP algorithm in np dimension (P)
lpalg1	LP algorithm in ni dimension (P)
lpalg2	LP algorithm in ni2 dimension (P)
lpext	LP data extension in np dimension (P)

lpext1	LP data extension in ni dimension (P)
lpext2	LP data extension in ni2 dimension (P)
lpfilt	LP coefficients to calculate in np dimension (P)
lpfilt1	LP coefficients to calculate in ni dimension (P)
lpfilt2	LP coefficients to calculate in ni2 dimension (P)
lpnupts	LP number of data points in np dimension (P)
lpnupts1	LP number of data points in ni dimension (P)
lpnupts2	LP number of data points in ni2 dimension (P)
lpopt	LP algorithm data extension in np dimension (P)
lpopt1	LP algorithm data extension in ni dimension (P)
lpopt2	LP algorithm data extension in ni2 dimension (P)
lpprint	LP print output for np dimension (P)
lpprint1	LP print output for ni dimension (P)
lpprint2	LP print output for ni2 dimension (P)
lptrace	LP output spectrum in np dimension (P)
lptrace1	LP output spectrum in ni dimension (P)
lptrace2	LP output spectrum in ni2 dimension (P)
lres	Used to plot lineshape values (M)
ls	List files in directory (C)
lsfid	Number of complex points to left-shift the np FID (P)
lsfid1	Number of complex points to left-shift ni interferogram (P)
lsfid2	Number of complex points to left-shift ni2 interferogram (P)
lsfrq	Frequency shift of the fn spectrum (P)
lsfrq1	Frequency shift of the fn1 spectrum (P)
lsfrq2	Frequency shift of the fn2 spectrum (P)
lvl	Zero-order baseline correction (P)
lvltlt	Control sensitivity of lvl and tlt adjustments (P)

laser SVS adiabatic localization

Syntax

Applicability VnmrJ 3.1

Description To set all frequencies: Click on the "`Set All Freq (Hz)` " Button

Phase cycle: `autoph='n'`, only phase cycles up to `nt`

`autoph='y'` goes through nt*array

`pcflag='n'` - turns off the phase cycle entirely

Central (Base) Frequency=resto-restol (or H1offset-restol)

Restol (Local offset) is a small offset 0 to ca. 20 Hz from the global frequency

of the reference. If after clicking on "`Set All Freq (Hz)`", Local offset appears to be large, then H1offset is not calibrated correctly.

For water suppression optimization:

`sglarray=1;`

`sglpower=0;`

For RF pulse `tpwr` array:

`sglpower=1`

`lastlk` Last lock solvent used (P)

Description Contains the name of the last lock solvent. Intended for use with the optional sample changer, this parameter is a user global variable (stored in the user's `global` file) and is not accessible to multiple users simultaneously. On a multiuser automation run, you should preferably access the last lock solvent from the file `/vnmr/acqqueue/lastlk`.

Values String containing the name of the solvent.

See also *NMR Spectroscopy User Guide*

Related `solvent` Lock solvent (P)

`lastmenu` Menu to display when Return button is selected (P)

Description Contains the name of the menu to display when the Return button is clicked on certain menus. For example, if the Phase F2 button in the 2D Processing menu (controlled by the file `process_2D`) is clicked, `lastmenu` is set to `'process_2D'`, the `ft` and `aph` commands are executed, the `ds` window is opened, and the Interactive 1D Spectrum Display menu (`ds_1` file) is displayed. Appearing in this menu is a Return button. Because `lastmenu` is still set to `'process_2D'`, clicking on the Return button redisplays the 2D Processing menu. `lastmenu` is stored in the `$vnmrsys/global` file.

Values String containing the name of a menu (e.g., `'process_2D'`).

See also *User Programming*

Related `menu` Change status of menu system (C)
`newmenu` Select a menu without immediate activation (C)

latch — Frequency synthesizer latching (P)

Description: Configuration parameter for whether the PTS frequency synthesizer has latching capabilities (all digits of the frequency value are sent to the synthesizer at once). The value for each channel is by the Latching label in the Spectrometer Configuration window.

Values: `'n'` indicates the synthesizers do not have latching capabilities (Not Present choice from the Spectrometer Configuration window).
`'y'` indicates the synthesizers have latching capabilities (Present choice from the Spectrometer Configuration window).

See also: *VnmrJ Installation and Administration*

Related:

`config`	Display current configuration and possibly change it (M)

lb — Line broadening in directly detected dimension (P)

Description: Sets line broadening and exponential weighting along the directly detected dimension. This dimension is often referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc.

Values: A positive value gives the desired line broadening, in Hz, which is then used to calculate a decaying exponential function of the form `exp(-t*π*lb)`.

A negative value gives a resolution enhancement function (increasing exponential) of the form `exp(-t*π*lb)`.

`'n'` turns off line broadening and exponential weighting.

See also: *NMR Spectroscopy User Guide*

Related:

`exp`	Find exponential value of a number (C)
`lb1`	Line broadening in 1st indirectly detected dimension (P)
`lb2`	Line broadening in 2nd indirectly detected dimension (P)

lb1 — Line broadening in 1st indirectly detected dimension (P)

Description: Sets line broadening and exponential weighting along the first indirectly detected dimension. This dimension is often referred to as the f_1 dimension in multidimensional data sets. `lb1` works analogously to the parameter `lb`. The "conventional" parameters (`lb`, `gf`, etc.) operate on the detected FIDs, while this "2D" parameter is used during processing of the interferograms.

Values: A positive value gives the desired line broadening, in Hz, which is then used to calculate a decaying exponential function of the form `exp(-t*π*lb1)`. A typical value is between 0.0001 to 1000 Hz.

A negative value gives a resolution enhancement function (increasing exponential) of the form exp(-t*p*lb1).

'n' turns off line broadening and exponential weighting.

See also *NMR Spectroscopy User Guide*

Related	exp	Find exponential value of a number (C)
	lb	Line broadening in directly detected dimension (P)
	lb2	Line broadening in 2nd indirectly detected dimension (P)

lb2 Line broadening in 2nd indirectly detected dimension (P)

Description Sets line broadening and exponential weighting along the second indirectly detected dimension. This dimension is often referred to as the f_2 dimension in multidimensional data sets. lb2 works analogously to the parameter lb. lb2 can be set with wti on the 2D interferogram data.

Values A positive value gives the desired line broadening, in Hz, which is then used to calculate a decaying exponential function of the form exp(-t*π*lb2).

A negative value gives a resolution enhancement function (increasing exponential) of the form exp(-t*π*lb2).

'n' turns off line broadening and exponential weighting.

See also *NMR Spectroscopy User Guide*

Related	exp	Find exponential value of a number (C)
	lb	Line broadening in directly detected dimension (P)
	wti	Interactive weighting (C)

lc1d Pulse sequence for LC-NMR (M)

Applicability Systems with LC-NMR accessory.

Description Creates parameters to set up a pulse sequence that can be used to start an LC-NMR run, including triggering the injection of a sample, and can be used also to obtain multiple solvent-suppressed spectra using multi frequency Shifted Laminar Pulses (SLP) and gradients. The sequence is coded without a d2 variable, thus allowing ni to be used to obtain a series of spectra without resulting in any delay in the sequence being incremented.

The sequence requires a phase table, lc1d, to be found in the tablib directory. Phases of the selective pulses, the observe pulse, and the receiver and separately controlled by phase variables.

Note that the lc1d sequence uses power scaling of shaped pulses, which is supported starting in VnmrJ 5.2. Because of this feature, this sequence *will not run* in earlier versions of VnmrJ.

lcdatast An LC-NMR plotting and display macro

Applicability VnmrJ 3.1

Description The engine for display and plotting of LC detector data. The default mode is a horizontal display of detector 1. Stop codes are marked if encountered in the LC data.

Arguments The following arguments are recognized and any number can be entered in any order.
plot sends output to plotter.
side activates vertical display on the side of the NMR data. In this mode the LC data are positioned between wc and wcmax and scaled appropriately to fit. In the stopped -flow mode, the side option also places the stopped-flow NMR data at a position so that it is time-aligned with the relevant LC peak.
det2 displays or plots the outputs of detectors one and two.
full The detector data is displayed vertically at `sc`.

Examples `lcdatast`(<arguments>)

`lcpar2d` Create 2D LC-NMR acquisition parameters (M)

Applicability Systems with LC-NMR accessory.

Description Creates the acquisition parameters `ni`, `sw1`, and `phase`, which can be used to acquire a 2D LC-NMR data set. `lcpar2d` is functionally the same as `addpar('2d')`.

Related
`addpar` Add selected parameters to current experiment (M)
`lcset2d` General setup for 2D LC-NMR experiments (M)

`lcpeak` Peak number (P)

Applicability Systems with LC-NMR accessory.

Description Contains the number of the peak being sensed or the loop being flushed.

`lcplot` Plot LC-NMR data (M)

Applicability Systems with LC-NMR accessory.

Syntax `lcplot`

Description Plots LC-NMR data. This macro is executed with the Plot LC-NMR button on the Spare pane when LC-NMR is active.

lcpsgset — Set up parameters for various LC-NMR pulse sequences (M)

Applicability: Systems with LC-NMR accessory.

Syntax: `lcpsgset(file,parameter1,parameter2,...,parameterN)`

Description: Sets up parameters for various LC-NMR pulse sequences using information in a `parlib` file. Rather than returning the entire parameter file, `lcpsgset` returns the parameters listed. `lcpsgset`, in general, is never entered from the keyboard but is used as part of experiment setup macros.

Arguments: `file` is the file from the user or system `parlib` that provides information on setting up parameters listed. The parameters `seqfil` and `pslabel` are set to the supplied file name.

`parameter1,parameter1,...,parameterN` are 1 to 11 parameters to be returned from the `parlib` file.

Examples: `lcpsgset('lccosy','ds','ap','ss','d1','axis','phase')`

lcscale — An LC-NMR plotting macro (M)

Applicability: VnmrJ 3.1

Description: A macro for drawing the LC time axis that is called by other macros.

lcset2d — General setup for 2D LC-NMR experiments (M)

Applicability: Systems with LC-NMR accessory.

Syntax: `lcset2d(experiment<,F2_dig_res<,F1_dig_res>>)`

Description: Runs the macro `lcpar2d` to create new parameters needed for 2D LC-NMR experiments, then selects starting values for a number of parameters. The `lcset2d` macro is “internal” and not normally entered directly by the user.

Arguments: `experiment` is the name of a 2D LC-NMR experiment.

`F2_dig_res` is the f_2 digital resolution desired, in Hz/pt.

`F1_dig_res` is the f_1 digital resolution desired, in Hz/pt.

Examples: `lcset2d('lcnoesy')`

left — Set display limits to left half of screen (C)

Description: Sets the horizontal control parameters sc and wc to produce a display (and subsequent plot) in the left half of a screen (and page). For 2D data, space is left for the scales.

Related:

center	Set display limits for center of screen (C)
full	Set display limits for a full screen (C)
fullt	Set display limits for full screen with room for traces (C)
right	Set display limits for right half of screen (C)

legrelay — Independent control of magnet leg relay (P)

Description: Gives override capability over the magnetic leg high and low (broad) band rf signal routing. This parameter does not normally exist but can be created by the user with the command `create('legrelay','string')`.

The `legrelay` override is operational only on standard systems shipped starting in November 1990 and on certain special systems shipped before that date. A system includes the override capability if it uses N-type connectors instead by BNC connectors on the magnet leg.

Values: `'n'` indicates normal logic is used to set the leg relay.

`'h'` indicates the leg relay is set to the high band.

`'l'` indicates the leg relay is set to the low (broad) band.

Any other value results in an error message and an abort of pulse sequence generation.

See also: *User Programming*

Related:

create	Create new parameter in a parameter tree (C)

length — Determine length of a string (C)

Syntax: `length(string):$string_length`

Description: Returns the length in characters of a specified string.

Arguments: `string` is zero or more characters enclosed in single quotes.

`string_length` is the number of characters (a real number) in `string`.

Examples:

```
length('abc'):r1
length(solvent):$len
```

See also: *User Programming*

Related:

substr	Select a substring from a string (C)

lf List files in directory (C)

Syntax lf<(directory)>

Description Lists the files in a directory, with output on the text output window. Directories are suffixed by "/", executable files by "*", and links by "@".

Arguments directory is the name of a directory. The default is the current working directory. lf is equivalent to the UNIX command ls -F and uses the same options (e.g., –l for a long listing such as lf('-l *.fid')).

Examples
```
lf
lf('data'))
lf('-l *.fid')
```

See also *NMR Spectroscopy User Guide*

Related

dir	List files in directory (C)
ls	List files in directory (C)

lgcp

Applicability VnmrJ 3.1

Description X Lee-Goldburg cross polarization (CP) between protons and X with a choice of SPINAL or TPPM decoupling. Used for selective CP with suppression of homonuclear dipolar interactions and for setup of Lee-Goldburg HETCOR.

Setup:

Load a calibrated data set and select the protocol Lgcp. For a new nucleus calibrate CP with Onepul and Tancpx and then select Lgcp.

Select the desired decoupling method, TPPM or SPINAL. The manual file onepul describes calibration of decoupling.

Before running Lee-Goldburg CP use Tancpx to calibrate aHhx with a known field strength. This can be done by calibrating CP with aH90 = aHhx and then determining pwH90. The proton field strength is ?B1H = 1/(4.0*pwH90). Match the Hartmann Hahn condition by varying aXhx as needed. It is helpful to array aXhx and note the positions of the intensity maxima.

Select the protocol Lgcp. Set ofHX = ?B1H/sqrt(2) and continue to use aHhx from the previous step. Note that of HX = - ?B1H/sqrt(2) is incorrect relative to the phase cycles of pwH90 and pwHtilt.

Recalibrate the proton excitation pulse pwH90 to the value usually used for proton excitation if desired.

After recalibration of pwH90 set pwHtilt = pwH90*35.3/90.0. Set Shape = 'const' and Channel = 'from'. Match the Hartmann-Hahn condition by varying aXhx as needed. It is helpful to array aXhx and note the positions of the intensity maxima. You will note that the pattern is shifted to higher amplitude because of the offset. Choose one of the maxima for CP.

Note: the CP module only allows an offset on the channel selected in Channel. Since this must be 'from' or protons, a ramped amplitude cannot be applied to X. It is possible to use a ramp on protons, though that is not a usual practice for Lee-Goldburg CP.

Array the contact time tXhx for the Lee-Goldburg CP. You will find that non-protonated X nuclei cross polarize weakly for all contact times and that protonated nuclei polarize to their fullest extent with a short contact time.

Lee-Goldburg CP cross polarization can only occur through an X-H dipolar mechanism and X-H-H three spin CP is suppressed. A long contact time however can increase the importance of long range X-H interactions. This distinction is important when using Lee Goldburg CP for Lee-Goldburg HETCOR. Spin diffusion is suppressed and long distance X-H correlations can be recognized.

Note that signal to noise of `Lgcp` is about 50% of that of `Tancpx`.

Parameter Groups:

90H: Module: no

Sequence: `tancpx.c`

Description: Provides a 90-degree pulse on dec that can be used as a preparation pulse. Also used to store the dec calibration.

Parameters: Channels Page

aH90 - the amplitude of the pulse.

pwH90 - the pulse length.

cpHX: Module: yes

Sequence: tancpx.c

Description: Implements constant, linear or tangent-ramped cross polarization from dec to obs.

Parameters: Sequence Page

shHX - 'c', constant, 'l', linear and 't', tangent shape on the channel designated with chHX.

chHX - the variable-amplitude channel, '`fr`' from or 'to', to.

aHhx - median amplitude of the dec channel.

aXhx - median amplitude of the obs channel.

bHX - +/- tangent curvature (>0 to 1.0e5).

dHX - amplitude width of the ramp. A negative width sets a ramp that starts high and decreases in amplitude.

tHX - contact time

ofHX - overall offset on the variable channel

frHX = 'dec' - channel with the initial polarization (must be set - not shown)

toHX = 'obs' - channel with the final polarization (must be set - not shown).

Implementation: CP hx = getcp("HX",0.0,0.0,0,1);

Underscore functions: _cp_(hx,phHhx,phXhx);

Hseq: Module: yes

Sequence: tancpx.c

Description: Chooses SPINAL or TPPM decoupling on the dec channel during acquisition.

Parameters: Sequence Page - the Hspinal and Htppm groups overlap.

Hseq - chooses the decoupling sequence, TPPM or SPINAL.

Implementation: DSEQ dec = getdseq("H"); The router implements getspinal() or gettppm().

Underscore functions: _dseqon(dec); runs _tppm(); or _spinal(); _dseqoff(dec); runs decprgoff();

Hspinal: Module: yes

Sequence: tancpx.c

Description: Implements SPINAL decoupling on the dec channel during acquisition.

Parameters: Sequence Page

`aHspinal` - amplitude of the dec channel.

`pwHspinal` - approximate 180-degree flip angle on resonance.

phHspinal - +/- small angle phase. SPINAL64 is implemented with phases = +/- 1.0, +/- 1.5 and +/-2.0 times phHspinal.

chHspinal = 'dec' must be set (not shown).

Implementation: SPINAL dec = getspinal("H"); or DSEQ dec = getdseq("H");

Underscore functions: _spinal(dec); and decprgoff(); or _dseqon(dec); and _dseqoff(dec);

Htppm: Module: yes

Sequence: tancpx.c

Description: Implements TPPM decoupling on the dec channel during acquisition.

Parameters: Sequence Page

`aHtppm` - amplitude of the dec channel.

`pwHtppm` - approximate 180-degree flip angle on resonance.

phHtppm - +/- small angle phase. TPPM is implemented with phases = +/- 1.0 times phHtppm for alternating pulses.

`chHtppm` = 'dec' must be set (not shown).

Implementation: TPPM dec = gettppm("H"); or DSEQ dec = getdseq("H");

Underscore functions: _tppmon(dec); and decprgoff(); or _dseqon(dec); and _dseqoff(dec);

liamp Amplitudes of integral reset points (P)

Description Stores the integral amplitudes at the integral reset points for a list of integrals. To display the values of liamp, enter display('liamp'). Values of liamp can also be accessed in MAGICAL macros using, for example, liamp[$i]. Values are stored as absolute numbers (summations of data point values) and, as such, are a function of the parameter fn. The values displayed by the dli, pir, and dpir programs are related to liamp values by the relationship:

Displayed or plotted integral =
liamp[i]*is/(fn/128)*ins)

See also *NMR Spectroscopy User Guide*

Related

display	Display parameters and their attributes (C)
dli	Display list of integrals (C)
dpir	Display integral amplitudes below spectrum (C)
fn	Fourier number in directly detected dimension (P)
lifrq	Frequencies of integral reset points (P)
pir	Plot integral amplitudes below spectrum (C)

lifrq Frequencies of integral reset points (P)

Description Stores the frequencies of integral reset points for a list of integrals. The frequencies are stored in Hz and are *not* adjusted by the reference parameters rfl and rfp.

See also *NMR Spectroscopy User Guide*

Related

liamp	Amplitudes of integral reset points (P)
rfl	Ref. peak position in directly detected dimension (P)
rfp	Ref. peak frequency in directly detected dimension (P)

liqbear Liquids Bearing Air Level (P)

Description This global parameter is the DAC value used when the liquids spinner bearing air is turned on. If the parameter does not exist the value defaults to 0xc000.
To create the parameter:

```
create('liqbear','integer','global')
setlimit('liqbear',65535,0,1,'global')
```

Values 0 - 65535

lscale

Syntax

Applicability VnmrJ 3.1

limnet

Syntax

Applicability VnmrJ 3.1

listenoff Disable receipt of messages from send2Vnmr (M)

Description Deletes the file `$vnmruser/.talk`, thereby disallowing `send2Vnmr` to send commands to VnmrJ

See also *User Programming*

Related `listenon` Enable receipt of messages from `send2Vnmr` (M)
`send2vnmr` Send a command to VnmrJ (U)

listenon Enable receipt of messages from send2Vnmr (M)

Description Writes files with the VnmrJ port number that `/vnmr/bin/send2Vnmr` needs to talk to VnmrJ. The command then to send commands to VnmrJ is
`/vnmr/bin/send2Vnmr $vnmruser/.talk command.`

See also *User Programming*

Related `listenoff` Disable receipt of messages from `send2Vnmr` (M)
`send2vnmr` Send a command to VnmrJ (U)

listparam List parameters in simple format (UNIX)

Syntax `listparam filename <parametergroup>`

Applicability VnmrJ 3.1

Description Lists parameters from a VNMR paramater file in a simple format using one line per parameter value. One application of `listparam` is for comparison purposes, in which case one would typically sort the output using the '`sort`' command, see below. `listparam` can also be used togenerate JCAMP-DX style parameter output.

Arguments filename is a VNMR parameter file, like
`$HOME/vnmrsys/exp1/procpar`
`$HOME/vnmrsys/exp1/curpar`
`$HOME/vnmrsys/global`
`/vnmr/conpar`
`xyz.fid/procpar`

parametergroup is an optional argument that permits specifying the parameter type. By default, only acquisition parameters are dumped. The following options exist (only the first two characters are relevant):

- acquisition - list acquisition parameters (default)
- processing - list processing parameters only
- display - list display parameters only
- spsim - list spin simulation parameters only
- sample - list sample parameters only
- all - list ALL parameters (output indicates group for each parameter)
- JCAMP - list acquisition parameters in JCAMP-DX format.
- Inactive parameters are suppressed, for FID saving
- JS - list acquisition, sample & processing parameters in JCAMP-DX format (for saving with spectra)
- JP - list acquisition, sample & processing parameters, plus parameters without Ggroup assignment in JCAMP-DX format (for saving with parameters)

Examples Using `listparam` on single files:

listparam vnmrsys/exp1/procpar | sort

listparam vnmrsys/exp1/curpar all

listparam xyz.fid/procpar JCAMP | sort

Using `listparam` to compare parameters:

listparam xyz.fid/procpar | sort > xyz.pars

listparam abc.fid/procpar | sort > abc.pars

diff xyz.pars abc.pars

Related
`diffparam`	report differences between parameter sets (UNIX)
vnmr2jcamp	create JCAMP parameters from VNMR parameters (UNIX)
`svfj`	save FID in JCAMP-DX format

`lkof` Track changes in lock frequency (P)

Description Tracks changes in the lock frequency resulting from changes in the solvent, and minor changes caused by the magnet drifting. The frequency units for `lkof` are in Hz, analogous to `sfrq` and `tof`, or `dfrq` and `dof`. `lkof` affects two components of the system: autolock on the console and `acqi` on the host computer. If `lkof` exists, it offsets the current value of the `lockfreq` parameter.

See also *NMR Spectroscopy User Guide*

Related `lockfreq` Lock frequency (P)

`ll2d` Automatic and interactive 2D peak picking (C)

Syntax (1) `ll2d<(options)><:$num>`

(2) `ll2d('info'<,#>):$peak_number,$f1,$f2,$amplitude, $volume,$label,$comment,$FWHH1,$FWHH2,$f1_min, $f1_max,$f2_min,$f2_max`

Description Automatically finds and integrates peaks that are above the threshold `th` in a 2D spectrum or a 2D plane of a 3D spectrum, and writes the peak location, volume, full-width at half-height (FWHH), volume, and the boundaries of the integrated region to a file in the `ll2d` subdirectory of the current experiment directory. For 2D spectra, the file name is `peaks.bin`, and for 2D planes of 3D spectra, the file name is `peaks_f#f#_#.bin`, where `f#f#` gives the plane direction (e.g., `f1f3`) and the final # gives the number of the plane. For easy import and export of peak data, `ll2d` also allows insertion and deletion of peaks interactively as well as reading and writing of text peak files.

Two-dimensional volumes are scaled in a manner analogous to 1D integrals, using the parameters `ins2` and `ins2ref`. The `ins2ref` parameter is the Fourier number scaled value of a selected volume. The reported value of a peak volume is (*unscaled volume*) × `ins2`/`ins2ref`/`fn`/`fn1`. The unscaled volume of a peak can be obtained from the command `ll2d('info',peak#)`. `ins2ref` can be set to the unscaled value divided by `fn` and `fn1`. The report volume for that peak is then the value of `ins2`.

Arguments `options` (syntax 1) are any of the following (`dconi` is not necessarily active):

- `'adjust'` is a keyword to adjust the bounds of all peaks in the displayed area so that no boundaries overlap, and then to recalculate peak volumes.
- `'draw'` is a keyword to draw the peaks, boxes, numbers, and labels on the spectrum based on the value of the parameter `ll2dmode`.
- `'info','total'` displays the total number of peaks in the current peak table. If a single return value is requested, printing is suppressed and the total number of peaks is returned.
- `'peak'` is a keyword to find all peaks in the displayed area above a threshold `th`. If `dconi` is active and in the box mode, `ll2d` finds peaks only in the area defined by the cursors. The `'peak'` option is the default if no arguments are entered.
- `'pos'` or `'neg'` keywords can be used in addition to `'peak'` or `'clear'` to operate only on positive or negative peaks.

- `'read'<,file >` reads in a binary peak file, where `file` is the name of the peak file. If a full path is not specified, the file is searched for first in the current working directory and then in the `ll2d` subdirectory of the current experiment directory.
- `'readtext'<,file>` reads in a text peak file, where `file` is the name of the peak file. If a full path is not specified, the file is searched for first in the current working directory and then in the `ll2d` subdirectory of the current experiment directory.
- `'reset'` is a keyword to delete all peaks in the peak table.
- `'volume'` is a keyword to find the bounds of each peak in the displayed area and integrate this area.
- `'writetext'<,file>` writes a peak file to a text file, where `file` is the name of the text file written. If a full path is not specified, the file is written in the current working directory.

  `options` (syntax 1) can also be any of the following (`dconi` must be active):
- `'clear'` is a keyword to delete all peaks in the displayed region if in the `dconi` cursor mode, or to delete all peaks within the cursors if in the `dconi` box mode.
- `'combine'` is a keyword to combine all peaks within the area defined by the cursors into a single peak (in `dconi` box mode only). The center of the new peak is at the average of all combined peaks' centers, and the bounds of this peak contains the maximum extents of the combined peaks' bounds. If all combined peaks have the same label, this label is assigned to the new peak.

CAUTION

All individual peaks to be combined are deleted prior to the creation of the new combination peak, and there is no automatic way to restore the original peaks. Therefore, it is recommended that you make a backup copy of the peak file prior to using this option.

- `'comment'` is a keyword to prompt for an 80-character comment. The comment is assigned to the nearest peak in the `dconi` cursor mode or to all peaks within the cursors in the `dconi` box mode.
- `'comment',text` executes the `'comment'` option using the string entered for `text` instead of prompting for a comment.
- `'label'` is a keyword to prompt for a 15-character label. The label is assigned to the nearest peak in `dconi` cursor mode or assigned to all peaks within the cursors in `dconi` box mode. To erase an existing label, enter a label consisting of one or more spaces.
- `'label',text` executes the `'label'` option using the string entered for `text` instead of prompting for a label.

- 'mark' is a keyword to insert a peak at the current cursor position if in the dconi cursor mode. If in the dconi box mode, 'mark' is a keyword to integrate the area within the cursors and assign that area to all peaks within the cursors that do not have their bounds already defined. If there are no peaks within the area defined by the cursors, using 'mark' finds the highest point within this area, marks that as a peak, integrates the area within the cursors, and assigns that area to the peak. The displayed values of the volume integrals are scaled by ins2 and ins2ref and the Fourier number of the 2D experiment.
- 'unmark' is a keyword to delete the nearest peak if in dconi cursor mode. If in the dconi box mode, 'unmark' deletes all peak bounds that are completely within the area defined by the cursors. Peaks are not deleted in the box mode.

options (syntax 1) also can be any of the following (dconi does not have to be active because ll2d is executed on a peak number):

- 'combine',#1,#2,... executes the 'combine' option on the list of peak numbers that follow the 'combine' keyword. If a single return value is requested, the peak number of the new combination peak is returned.
- 'comment',text,# executes the 'comment' option on peak # using the string entered for text instead of prompting for a comment.
- 'label',text,# executes the 'label' option on peak # using the string entered for text instead of prompting for a label.
- 'unmark',# deletes peak number #.

$num (syntax 1) is a return value set to the total number of peaks that have been picked unless the arguments 'combine',#1,#2,... are used, in which case $num is the number of the newly created combination peak.

Syntax 2 arguments are the following:

- 'info'<,#> displays information in the text window about peak number #. If no peak number is included, dconi must be active and the default is the peak nearest to the cursor. If return values are requested, the display is suppressed.
- $peak_number is a return value set to the number of the peak, either the second argument # or, if no value is given for #, the peak nearest to the cursor in dconi.
- $f1 and $f2 are return values set to the peak frequencies in f_1 and f_2 of peak $peak_number.
- $amp is a return value set to the amplitude of peak $peak_number.
- $vol is a return value set to the unscaled volume of $peak_number. peak. This value can be used to set the ins2ref parameter.
- $label is a return value set to the label of peak $peak_number.
- $comment is a return value set to the comment about $peak_number.
- $FWHH1 and $FWHH2 are return values set to full-width at half-height of $peak_number.

- $f1_min,$f1_max,$f2_min,$f2_max are return values set to the bounds of $peak_number.

Examples
```
ll2d
ll2d:$npeaks
ll2d('volume')
ll2d('read','peaklist.inp')
ll2d('mark')
ll2d('label','Peak 1')
ll2d('info','total'):$npeaks
ll2d('combine',3,4,5,6):$cpn
ll2d('info',3):$num,$f1,$f2,$amp,$vol,$label
```

See also *NMR Spectroscopy User Guide*

Related

dconi	Interactive 2D contour display (C)
ins2	2D volume value (P)
ins2ref	Fourier number scaled volume of a peak (P)
ll2dbackup	Copy current ll2d peak file to another file (M)
ll2dmode	Control display of peaks picked by ll2d (P)
parll2d	Create parameters for 2D peak picking (M)
pll2d	Plot results of 2D peak picking (C)
th	Threshold (P)
th2d	Threshold for integrating peaks in 2D spectra (P)
xdiag	Threshold for excluding diagonal peaks when peak picking (P)

ll2dbackup Copy current ll2d peak file to another file (M)

Syntax ll2dbackup<(file)>

Description Backs up the current ll2d peak file by copying it to a file with a different file name. The default ll2d peak file is peaks.bin for 2D data.

Arguments file is the name to be given to the backup file. If a full path is not specified, the file is written to the current working directory. If no argument is provided, the system prompts for a file name. If no file name is specified at the prompt, the default ll2d peak file name with .bck appended is used.

See also *NMR Spectroscopy User Guide*

Related

ll2d	Automatic and interactive 2D peak picking (C)

ll2dmode Control display of peaks picked by ll2d (P)

Description Sets the display attributes of peaks picked by the ll2d command

Values A string variable composed of 4 characters, with each character taking the value 'y' (display the peak attribute) or 'n' (do not display the attribute). The first character determines if a “+” is drawn on the screen in dconi displays to mark peaks, the second character controls the drawing of the peak number, the third character controls drawing of the peak bounds box, and the last character controls drawing of the peak label.

See also *NMR Spectroscopy User Guide*

Related	ll2d	Automatic and interactive 2D peak picking (C)

llamp List of line amplitudes (P)

Description Stores a list of line amplitudes above the threshold set by th.

See also *NMR Spectroscopy User Guide*

Related	dll	Display listed line frequencies and intensities (C
	llfrq	List of line frequencies (P)
	th	Threshold (P)

llfrq List of line frequencies (P)

Description Stores a list of line frequencies above the threshold set by th. Frequencies are stored in Hz and are *not* adjusted by reference parameters rfl and rfp.

See also *NMR Spectroscopy User Guide*

Related	llamp	List of line amplitudes (P)
	rfl	Ref. peak position in directly detected dimension (P)
	rfp	Ref. peak frequency in directly detected dimension (P)
	th	Threshold (P)

ln Find natural logarithm of a number (C)

Syntax `ln(value)<:n>`

Description Finds the natural logarithm (base e) of a number. To convert the value to base 10, use $log_{10}x$ = 0.43429**ln(x)*.

Arguments `value` is a number.

`n` is the return value giving the logarithm of `value`. The default is to display the logarithmic value in the status window.

Examples
```
ln(.5)
ln(val):ln_val
```

See also *User Programming*

Related

atan	Find arc tangent of a number (C)
cos	Find cosine value of an angle (C)
exp	Find exponential value of a number (C)
sin	Find sine value of an angle (C)
tan	Find tangent value of an angle (C)

load — Load status of displayed shims (P)

Description Sets whether shim values are used. load is automatically set to 'y' by the rts and is automatically set to 'n' by su, go, au, and shim. Shim DAC values are automatically loaded after the console is rebooted (the last values returned before the console was rebooted).

Values 'y' begins any noninteractive shimming process or data acquisition after loading the shim DACs with the shim values from the current experiment. It also prevents acqi from delivering shim values to that experiment.

'n' begins any noninteractive shimming process or data acquisition with the current values stored in the shim DACs. Shim values in the current experiment are ignored.

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display process (C)
au	Submit experiment to acquisition and process data (C)
go	Submit experiment to acquisition (C)
rts	Retrieve shim coil settings (C)
shim	Submit an autoshim experiment to acquisition (C)
su	Submit a setup experiment to acquisition (M)

loadcolors — Load colors for graphics window and plotters (M)

Syntax `loadcolors<(color_file)>`

Description Loads the color table for VnmrJ graphics window and plotters. loadcolors is generated by the color program and includes a series of setcolor commands. On bootup, the bootup macro calls loadcolors to set the graphics and plotter colors.

The loadcolors macro checks the value of maxpen to decide if the plotter supports colors. If maxpen is greater than 1, a color printer is configured.

Arguments color_file is the name of the file to load. loadcolors first searches for this file in the directory $vnmruser/templates/ directory. If not found there, loadcolors then searches the user_templates/vnmr directory. The default is a color table with the same name as the value of the plotter parameter that `loadcolors` searches for in the same two directories.

Examples `loadcolors`
`loadcolors('mycolortable')`

See also *VnmrJ Imaging NMR*

Related		
	bootup	Macro executed automatically when VnmrJ activated (M)
	color	Select plotting colors from a graphic interface (M)
	maxpen	Maximum number of pens to use (P)
	setcolor	Set colors for graphics window and for plotters (C)

loaduserprefs Load Operator Preferences

See also At operator login, this macro loads the operator-specific parameter values set in the Preferences/UserPrefs panel.

loc Location of sample in tray (P)

Description Indicates whether a sample changer is present and enabled, present but disabled, or not present. If the changer is present and enabled, the value of `loc` sets the location in the tray of the sample in use or to be used. The `loc` parameter is stored in the global tree. When an acquisition is started, certain global parameters, including `loc`, are saved with the experiment parameters. The `saveglobal` parameter specifies which global parameters are saved.

The `auto_au` macro controls most of the automation features, including setting the value of `loc`.

Values A number between 1 and `traymax` indicates the sample location.

0 indicates the changer is not present or disabled.

See also *NMR Spectroscopy User Guide; VnmrJ Walkup*

Related		
	auto_au	Controlling macro for automation (M)
	saveglobal	Save selected parameters from global tree (P)
	traymax	Sample changer tray size (P)

locaction Locator action (M)

Description Perform an action on an object in the locator database. The action depends on the type of object selected, the action performed, and the target selected for the action.

Related

dndfid	Retrieve and process fid data from the locator (M)
dndjoin	Join a work space from the locator (M)
dndpar	Retrieve a parameter set from the locator (M)
dndshims	Retrieve a shimset set from the locator (M)
locprotoexec	Execute a protocol from the locator (M)
xmmakenode	Make a new study queue node (M)

lock Submit an Autolock experiment to acquisition (C)

Description Performs an automatic locking operation using the acquisition computer, optimizing lock power, phase, and gain. If necessary, `lock` obtains lock through a software-controlled search. `lock` is the only method to automatically adjust lock phase (usually needed only after probe change or lock channel tuning). `lock` also sets the rf frequencies, decoupler status, and temperature.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (C)
change	Submit a change sample experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
sample	Submit change sample, autoshim experiment to acquisition (M)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
su	Submit a setup experiment to acquisition (M)

lockacqtc Lock loop time constant during acquisition (P)

Description Controls time constant of lock loop during acquisition (i.e., time constant by which the lock feedback corrects disturbances of the magnetic field).

Values 1, 2, 3, or 4 (where 1 sets 1.2 seconds, 2 sets 4.7 seconds, 3 sets 12 seconds, and 4 sets 48 seconds).
If `lockacqtc` does not exist, it is set to 48 seconds. All systems are designed to work well with the default settings, and there should rarely be a reason to alter the lock time constant. However, to experiment with other values, create `lockacqtc` and set a new value:

```
create('lockacqtc','integer','global')
setlimit('lockacqtc',4,1,1,'global') lockacqtc=n
```

where *n* is the new value.

See also *NMR Spectroscopy User Guide*

Related		
	create	Create new parameter in a parameter tree (C)
	locktc	Lock time constant (P)
	setlimit	Set limits of a parameter in a tree (C)

lockfreq Lock frequency (P)

Description Sets system lock frequency. The value is entered using the Lock Frequency label in Spectrometer Configuration window. **The value of lockfreq must be set correctly in order to observe NMR signals.**

`lockfreq` can find the lock signal or resonance. Traditionally, Varian spectrometers have used the parameter `z0` for this purpose; however, using `lockfreq` can require less shimming when switching solvents and less adjustment to the lock phase. To use `lockfreq`, set `z0='n'`.

Values 1 to 160 (in MHz), `'n'`

Use the true ^{2}H frequency. Typical values of `lockfreq` are shown in the chart below.

^{1}H Frequency		
200	30.710	30.6976
300	46.044	46.0625
400	61.395	61.471
500	76.729	...
600	92.095	...
750	115.250	...

Refer to the manual *VnmrJ Installation and Administration* for details on finding the correct lock frequency.

The commands; `go`, `lock`, `shim`, and `su` reset the lock frequency in the console to the current value of `lockfreq`. Lock frequency in the console can be set with the `sethw` command.

`lockfreq` is offset by the value of `lkof`, if that parameter exists, but `sethw` directly uses its numeric argument, without any offset by `lkof`.

See also *VnmrJ Installation and Administration*; *NMR Spectroscopy User Guide*

Related		
	config	Display current configuration and possibly change it (M)
	go	Submit experiment to acquisition (M)
	lkof	Track changes in lock frequency (P)
	lock	Submit an Autolock experiment to acquisition (C)
	sethw	Set values for hardware in acquisition system (C)
	setlockfreq	Set lock frequency (C)
	shim	Submit an Autoshim experiment to acquisition (C)

su	Submit a setup experiment to acquisition (M)
z0	Z0 field position (P)

lockgain Lock gain (P)

Description Contains the current lock gain value as set by computer control. The value is stored in `vnmrsys/global` and can be examined by typing `lockgain?`.

Values 0 to 48 dB, in 1-dB steps.

See also *NMR Spectroscopy User Guide*

lockphase Lock phase (P)

Description Contains the current lock phase. The value is stored in `vnmrsys/global` and can be examined by typing `lockphase?`.

Values 0 to 360, in degrees, in 1.4-degree steps.

See also *NMR Spectroscopy User Guide*

lockpower Lock power (P)

Description Contains the current lock power value as set by computer control. The value is stored in `vnmrsys/global` and can be examined by typing `lockpower?`.

Values 0 to 68 dB, in 1-dB steps, 68 is full power.

See also *NMR Spectroscopy User Guide*

locktc Lock time constant (P)

Description Controls lock loop time constant when system is not performing acquisition (idle, lock display, shim display, FID display, autoshim, autolock, etc.).

Values 1, 2, 3, or 4 (where 1 corresponds to 1.2 seconds, 2 to 4.7 seconds, 3 to 12 seconds, and 4 to 48 seconds). If `locktc` does not exist, the system uses a value of 1, the fastest value. To experiment with other value, create `locktc` and set a value (e.g.,
`create('locktc','integer','global')`
`setlimit('locktc',4,1,'global') locktc=2`).

See also *NMR Spectroscopy User Guide*

Related `create` Create new parameter in a parameter tree (C)
`lockacqtc` Lock acquisition time constant (P)
`setlimit` Set limits of a parameter in a tree (C)

log

Syntax `log(base 10)(x) = 0.43429 * ln(x)`

Applicability VnmrJ 3.1

Description `sin` `sin(angle)<:n>, radians,` n is destination parameter
`cos` `cos(angle)<:n>, radians,` n is destination parameter
`tan` `tan(angle)<:n>, radians,` n is destination parameter
`asin` `asin(angle)<:n> radians,` n is destination parameter
`acos` `acos(angle)<:n>, radians,` n is destination parameter
`atan` `atan(value)<:n>, pi/2 to -pi/2n,` n is destination parameter
`atan2` `atan2(x,y)<:n>, y/x is pi/2 to -pi/2n,` n is destination parameter
`exp` `exp(value)<:n>,` n is destination parameter
`ln` `ln(value)<:n>,` n is destination parameter
`sqrt` `sqrt(value)<:n>,` n is destination parameter
`abs` `abs(value)<:n>,` n is destination parameter

logate Transmitter local oscillator gate (P)

Description Specifies whether the transmitter local oscillator (L.O.) is gated with the transmitter rf output or with the transmitter I.F. (intermediate frequency).

The `logate` parameter does not exist in most parameter sets; the system internally sets it to `'l'`. To use the value `'s'`, create `logate` and change the value by entering: `create('logate','string')` `setenumeral('logate',2,'l','s') logate='s'`.

Values `'l'` makes the transmitter L.O. gate with the rf output, producing better signal-to-noise, usually most important in liquids NMR.

`'s'` makes the transmitter L.O. gate with the I.F. signal, producing sharper pulses, especially important in solid-state NMR.

See also *User Guide: Solid-State NMR*

Related `create` Create new parameter in a parameter tree (C)
`setenumeral` Set values of a string variable in a tree (C)

`lookup` Look up words and lines from a text file (C)

Applicability VnmrJ

Syntax `lookup('codeword',argument<,'codeword',argument<,...>>):$n1<$n2<,...>>`

Description Search a text file or files for a word or any string of characters delimited by white space characters (space character, a tab, a new line, a carriage return, or a comma) or codeword and return to the user subsequent words or lines.

The white space characters may be specified. Punctuation marks, unless they are defined as white space as the comma is by default, also form words or are part of a word. A line is any string of characters from the current word to the next carriage return. A line will include all "white space" characters except the carriage return. Depending on the codeword, word searches and word counts can be case insensitive or case sensitive.

The codewords `mfile` and `filekey` implement multiple text file lookup and `lookup` reads the contents of the specified files.

The `mfile` and `file` keywords are used together to keep track of various locations within a single file to restart the search from that location.

The first time a file is selected, or the search is restarted at the beginning of the file, use the name of the file instead of the `filekey`. Subsequent calls to `lookup` on this file use the value returned by the `filekey` codeword as the argument following the `mfile` codeword. The `mfile` codeword resets the white space to the default values.

Arguments Default white space characters: space character, tab, new line, carriage return, or comma.

`file` codeword specifies that the next supplied argument is the name of the active text file. This codeword must be the first argument and the file name must be the second argument passed to `lookup`. The search through a text file is a top to bottom search. The `file` codeword resets the search to start from the top of the text file. Subsequent searches through a previously accessed text file will continue from where the previous search stopped provided the `file` codeword is not used. The `file` codeword resets the white space characters to their default values.

`mfile` codeword specifies that the next supplied argument is the `filekey` to select one of multiple text files to access. This codeword must be first argument and the `filekey` must be the second argument passed to `lookup` if `mfile` is used.

`seek` this codeword causes the `lookup` program to search the text file for words which match those supplied as arguments following the `seek` codeword. An implicit `seek` is initially assumed for each call to `lookup`. The `lookup` program maintains a pointer to the word following the last successful `seek`. The first argument following an explicit `seek` codeword is interpreted as a word to search for and not a codeword. The second or later argument following an explicit `seek` is interpreted as a codeword if it matches one of the nine cases. Therefore, for example, one can search for the word `file` without having it interpreted as a codeword by having it immediately follow the `seek` codeword in the argument list. This seek is case insensitive.

`seekcs` this codeword is the case sensitive equivalent to the `seek` codeword and follows the same rules as `seek`. Alternate case sensitive and case insensitive searches are allowed.

`skip` increments the word pointer to the next word in the text file. This codeword may optionally be followed by a number which will specify how many words to skip.

`read` returns to the user the word currently being pointed to and increments the pointer to the next word in the text file. This codeword may optionally be followed by a number which will specify how many words to return to the user.

`readline` returns to the user the word currently being pointed to and all following words until the end of the current line. The pointer is moves to the first word of the next line in the text file. This codeword may optionally be followed by a number which will specify how many lines to return to the user.

`count` returns to the user the number of times words in the text file match the subsequent argument. The count starts at the current word pointer and proceeds to the end of the text file. The word `count` is not case sensitive.

`countcs` this codeword is the case sensitive equivalent to the count codeword. In all other respects, it is the same as `count`.

`delimiter` this codeword specifies that the next supplied argument is a list of characters which are used to identify the white space used to identify words.

Characters are specified by the following:
`\n` — new line
`\t` — tab
`\r` — carriage return
`\\` — backslash
`\'` — single quote.

The two arguments `delimiter`,`' \t\n\r'`, reselect the default white space. The `file` codeword will also reselect the default white space. The distinction is that the `file` codeword restarts the search from the beginning of the file while the `delimiter` codeword continues from the current search position. An implicit `seek` is applied following the `'delimiter'` codeword and argument.

`filekey` returns the current location within the file being accessed. Combined with the `mfile` codeword, a subsequent call to `lookup` starts the search at the location within the file specified by the value of `filekey`. The `filekey` serves both as a pointer to the file and as the character offset within that file.

Examples `lookup('file',systemdir + '/manual/lookup')`
Select this file for the search.

`lookup('user','skip',2,'read',2,'readline'):$n1,$n2,$n3,$ret`

Seek is assumed with the call to `lookup`. Finding the word `user` the next instruction,`'skip',2`, causes the pointer to jump two words. The codeword `read` causes the word to be put into `$n1`. The argument 2 specifies two words to be read into `$n2.` The word pointer now points to the next. The codeword `readline` causes the remaining characters up to the next carriage return to be placed in `$n3`. The pointer now points to the first word in the next line. The variable `$ret` is set to the number of arguments successfully returned from the text file and is used to determine if the end of the text file has been reached.

`lookup('skip',8,'read','skip',3,'read',2,'seek','comma'):$n3,$n4,$n5`

`'Skip',8` causes the pointer to jump eight words. The `'read'` sets `$n3` equal to word where the pointer is now located. `'Skip',3` jumps the next three words. `'Read',2` reads two consecutive words and sets `$n4` to the first word and `$n5` equal the second word. The `seek` argument searches for the word `'comma'`. If the word `'comma' is` at the end of a sentence it will not be found because the period is treated (by default) as part of the word. Define the period as a white space and occurrences comma at the end of sentences are also found. The word pointer now points to the next word.

```
lookup('delimiter',' ,\'.\n\t"','seek','file',
'skip',6,'read'):n6
```

The `delimiter` with the argument `' ,\'.\n\t"'` sets white space to space, comma, single quote, period, new line, tab, and double quote. Setting single quotes to white space causes the explicit `seek` to select the next argument `file` as a search word not a codeword. The search for the word `must` matches both MUST and must because `seek` is not case sensitive. `'Skip',6` jumps six words. `Read` sets `$n6` equal to word found between the next set of single quotes because single quotes are defined as white space.

`lookup('seekcs','Test','read'):$n7`

`seekcs` is the case sensitive form of `seek` and searches for the word that is an exact match to the case of `Test` (the argument following the codeword `seekcs`). Finding the word `'Test'`, `read` sets `$n7` to `search`. Any occurrence of the word `test` is skipped.

See also *User Programming*

Related	`dialog`	Display a dialog box from a macro (C)
	`systemdir`	VnmrJ system directory (P)

locprotoexec Execute a protocol from the locator (M)

Description When a protocol is dragged from the locator and dropped onto the graphics canvas, this macro adds the protocol to the end of the study queue, and executes the macro associated with the protocol.

Related		
	dndfid	Retrieve and process fid data from the locator (M)
	dndjoin	Join a work space from the locator (M)
	dndpar	Retrieve a parameter set from the locator (M)
	dndshims	Retrieve a shimset set from the locator (M)
	locaction	Locator action (M)
	xmmakenode	Make a new study queue node (M)

lp First-order phase in directly detected dimension (P)

Description Specifies the first-order phase-correction angles along the directly detected dimension according to the formula

absorption spectrum(ω) =
real channel(ω) * *cos*θ + *imaginary channel*(ω) * *sin* θ

where the phase angle θ is a function of frequency, i.e.
θ = rp + $(\omega - \omega_o)$/sw *lp

ω_o is defined to be the right end of the spectrum (i.e., lp has zero effect at the right edge of the spectrum and a linearly increasing effect going to the left). In multidimensional data sets, lp controls the phase of the directly detected dimension: f_2 dimension in 2D data sets, f_3 dimension in 3D data sets, etc.

Values –3600 to +3600, in degrees. Typical values are between 0 and –180.

See also *NMR Spectroscopy User Guide*

Related		
	aph	Automatic phase adjustment of spectra (C)
	lp1	First-order phase in 1st indirectly detected dimension (P)
	lp2	First-order phase in 2nd indirectly detected dimension (P)
	rp	Zero-order phase in directly detected dimension (P)
	setlp0	Set parameters for zero linear phase (M)

lp1 First-order phase in 1st indirectly detected dimension (P)

Description Controls the first-order phase constant along the first indirectly detected dimension during the process of phase-sensitive 2D transformation. The first indirectly detected dimension is often referred to as the f_1 dimension of a multidimensional data set.

See also *NMR Spectroscopy User Guide*

Related
lp First-order phase in directly detected dimension (P)
lp2 First-order phase in 2nd indirectly detected dimension (P)
rp1 Zero-order phase in 1st indirectly detected dimension (P)

lp2 First-order phase in 2nd indirectly detected dimension (P)

Description Controls the first-order phase constant along the second indirectly detected dimension during a ds, dconi, or equivalent display operation on the 2D data or a 1D trace therein. The second indirectly detected dimension is often referred to as the f_2 dimension of a 3D (or higher dimensionality) data set.

See also *NMR Spectroscopy User Guide*

Related
dconi Interactive 2D contour display (C)
ds Display a spectrum (C)
lp First-order phase in directly detected dimension (P)
rp2 Zero-order phase in 2nd indirectly detected dimension (P)

lpalg LP algorithm in np dimension (P)

Description Specifies the linear prediction (LP) algorithm to use in the np dimension. The resulting LP coefficients are used to appropriately extend the complex time-domain data prior to a normal Fourier transform. The LP algorithms work both on complex t_2 FIDs and on hypercomplex or complex t_1 interferograms. Enter addpar('lp') to create lpalg and other np dimension LP parameters in the current experiment

Values 'lpfft' does a least-squares calculation of lpfilt complex LP coefficients using lpnupts complex time-domain data points. Eigenvalue decomposition of the least-squares matrix is done using Householder tridiagonalization followed by the QL method with implicit shifts.

'lparfft' does a non-least-squares calculation of lpfilt complex LP coefficients using (lpfilt+1) complex, autoregressive (AR) matrix elements. These AR matrix elements are calculated from the raw, complex time-domain data using lpnupts points.

Note that the 'lpfft' algorithm is preferred by far. While 'lparfft' can model broad lines and can extend data sets when mostly noise exists, it cannot model narrow lines.

See also *NMR Spectroscopy User Guide*

Related
addpar Add selected parameters to the current experiment (M)
lpalg1 LP algorithm in ni dimension (P)

lpalg2	LP algorithm in ni2 dimension (P)
lpext	LP data extension in np dimension (P)
lpfilt	LP coefficients to calculate in np dimension (P)
lpnupts	LP number of data points in np dimension (P)
lpopt	LP algorithm data extension in np dimension (P)
lpprint	LP print output in np dimension (P)
lptrace	LP output spectrum in np dimension (P)
np	Number of data points (P)
proc	Type of processing on np FID (P)
strtlp	Starting point for LP calculation in np dimension (P)
strtext	Starting point for LP data extension in np dimension (P)

lpalg1 — LP algorithm in ni dimension (P)

Description Specifies the LP (linear prediction) algorithm to use in the ni dimension. lpalg1 functions analogously to lpalg. Enter addpar('lp',1) to create lpalg1 and other ni dimension LP parameters in the current experiment.

Values 'lpfft' or 'lparfft'

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpalg	LP algorithm in np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lpalg2 — LP algorithm in ni2 dimension (P)

Description Specifies the LP (linear prediction) algorithm to use in the ni2 dimension. lpalg2 functions analogously to lpalg. Enter addpar('lp',2) to create lpalg2 and other ni2 dimension LP parameters in the current experiment.

Values 'lpfft' or 'lparfft'

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M
lpalg	LP algorithm in np dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

lpext — LP data extension in np dimension (P)

Description: Specifies number of complex time-domain data points for LP (linear prediction) in the np dimension by which the original data is to be extended (or altered) in either the forward or backward direction. lpext is constrained by (strtext-lpext)>= ≥0 for lpopt='b' and by (strtext+lpext-1)<=fn/2 for lpopt='f'. In the np direction, if (strtext-lpext)=0 and lpopt='b' (backwards linear prediction with calculation of the first point), fpmult defaults to the theoretical value of 0.5 instead of 1.0. Enter addpar('lp') to create lpext and other np dimension LP parameters in the current experiment.

Related:

addpar	Add selected parameters to the current experiment (M)
lpalg	LP algorithm in np dimension (P)
lpext1	LP data extension in ni dimension (P)
lpext2	LP data extension in ni2 dimension (P)
lpopt	LP algorithm data extension in np dimension (P)
np	Number of data points (P)
strtext	Starting point for LP data extension in np dimension (P)

lpext1 — LP data extension in ni dimension (P)

Description: Specifies number of complex time-domain data points for LP (linear prediction) in the ni dimension by which the original data is to be extended (or altered) in either the forward or backward direction. lpext1 functions analogously to lpext. Enter addpar('lp',1) to create lpext1 and other ni dimension LP parameters in the current experiment.

Related:

addpar	Add selected parameters to the current experiment (M)
lpext	LP data extension in np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lpext2 — LP data extension in ni2 dimension (P)

Description: Specifies number of complex time-domain data points for LP (linear prediction) in the ni2 dimension by which the original data is to be extended (or altered) in either the forward or backward direction. lpext2 functions analogously to lpext. Enter addpar('lp',2) to create lpext2 and other ni2 dimension LP parameters in the current experiment.

Related:

addpar	Add selected parameters to the current experiment (M)

lpext	LP data extension in np dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

lpfilt LP coefficients to calculate in np dimension (P)

Description Specifies number of complex LP (linear prediction) coefficients in the np dimension to be calculated from a specified region of the time-domain data. lpfilt should be greater than nsignals, where nsignals is the number of sinusoidal signals contained in that FID (or interferogram). Enter addpar('lp') to create lpfilt and other np dimension LP parameters in the current experiment.

Related

addpar	Add selected parameters to the current experiment (M)
lpalg	LP algorithm in np dimension (P)
lpfilt1	LP coefficients to calculate in ni dimension (P)
lpfilt2	LP coefficients to calculate in ni2 dimension (P)
np	Number of data points (P)

lpfilt1 LP coefficients to calculate in ni dimension (P)

Description Specifies number of complex LP (linear prediction) coefficients in the ni dimension to be calculated from a specified region of the time-domain data. lpfilt1 functions analogously to lpfilt. Enter addpar('lp',1) to create lpfilt1 and other ni dimension LP parameters in the current experiment.

Related

addpar	Add selected parameters to the current experiment (M)
lpfilt	LP coefficients to calculate in np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lpfilt2 LP coefficients to calculate in ni2 dimension (P)

Description Specifies number of complex LP (linear prediction) coefficients in the ni2 dimension to be calculated from a specified region of the time-domain data. lpfilt2 functions analogously to lpfilt. Enter addpar('lp',2) to create lpfilt1 and other ni2 dimension LP parameters in the current experiment.

Related

addpar	Add selected parameters to the current experiment (M)
lpfilt	LP coefficients to calculate in np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lpnupts LP number of data points in np dimension (P)

Description Specifies number of complex time-domain data points in the np dimension to be used in constructing the autoregressive (lpalg='lparfft') or least- squares (lpalg='lpnefft') matrix from which the complex LP (linear prediction) coefficients are calculated. Note that lpnupts greater than or equal to 2*lpfilt is required for both algorithms. Enter addpar('lp') to create lpnupts and other np dimension LP parameters in the current experiment.

Related	addpar	Add selected parameters to the current experiment (M)
	lpalg	LP algorithm in np dimension (P)
	lpfilt	LP coefficients to calculate in np dimension (P)
	lpnupts1	LP number of data points in ni dimension (P)
	lpnupts2	LP number of data points in ni2 dimension (P)
	np	Number of data points (P)

lpnupts1 LP number of data points in ni dimension (P)

Description Specifies number of complex time-domain data points in the ni dimension to be used in constructing the autoregressive (lpalg1='lparfft') or least- squares (lpalg1='lpnefft') matrix from which the complex LP (linear prediction) coefficients are calculated. lpnupts1 functions analogously to lpnupts. Enter addpar('lp',1) to create lpnupts1 and other ni dimension LP parameters in the current experiment.

Related	addpar	Add selected parameters to the current experiment (M)
	lpalg1	LP algorithm in ni dimension (P)
	lpnupts	LP number of data points in np dimension (P)
	ni	Number of increments in 1st indirectly detected dimension (P)

lpnupts2 LP number of data points in ni2 dimension (P)

Description Specifies number of complex time-domain data points in the ni2 dimension to be used in constructing the autoregressive (lpalg2='lparfft') or least- squares (lpalg2='lpnefft') matrix from which the complex LP (linear prediction) coefficients are calculated. lpnupts2 functions analogously to lpnupts. Enter addpar('lp',2) to create lpnupts2 and other ni2 dimension LP parameters in the current experiment.

Related	addpar	Add selected parameters to the current experiment (M)
	lpalg2	LP algorithm in ni2 dimension (P)

lpnupts	LP number of data points in np dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

lpopt — LP algorithm data extension in np dimension (P)

Description Specifies how the specific LP (linear prediction) algorithm is to extend (or alter) forward or backward the time-domain data in the np dimension. Enter addpar('lp') to create lpopt and other np dimension LP parameters in the current experiment.

Multiple LP operations, extended forward or backward, can be performed on each FID or interferogram. This is accomplished by arraying the LP processing parameters (e.g., lpopt='b','f','b'). The number of LP operations is determined by the LP processing parameter with the largest array size. LP parameters having a smaller array size are padded out with their last value. The most common use for this capability is to back-calculate the first 1 to 2 points in an FID or interferogram and subsequently to extend the length of the time-domain data by LP.

A printout can be obtained for each LP operation on an individually definable FID or interferogram. For example, if lpprint=30,30 and lptrace=1,2, the text file lpanalyz.out.1 contains the LP printout for the first LP operation on FID 1 and lpanalyz.out.2 contains the LP printout for the second LP operation on FID 2.

Values 'b' indicates the LP coefficients are to be used in the back-calculation of a specified number of time-domain data points.

'f' indicates the LP coefficients are to be used in the forward extension of the time-domain data by a specified number of points. The characteristic polynomial in z space, derived from the complex LP coefficients, is set up and rooted. Any root found to lie outside the unit circle is reflected back into the unit circle. New complex LP coefficients are then calculated from these adjusted complex roots.

Related

addpar	Add selected parameters to the current experiment (M)
lpalg	LP algorithm in np dimension (P)
lpopt1	LP algorithm data extension for ni dimension (P)
lpopt2	LP algorithm data extension for ni2 dimension (P)
lpprint	LP print output for np dimension (P)
lptrace	LP output spectrum for np dimension (P)
np	Number of data points (P)

lpopt1 — LP algorithm data extension in ni dimension (P)

Description Specifies how the specific LP (linear prediction) algorithm is to extend (or alter) forward or backward the time-domain data in the ni dimension. lpopt1 functions analogously to lpopt. Enter addpar('lp',1) to create lpopt1 and other ni dimension LP parameters in the current experiment.

Related	addpar	Add selected parameters to the current experiment (M)
	lpopt	LP algorithm data extension for np dimension (P)
	ni	Number of increments in 1st indirectly detected dimension (P)

lpopt2 LP algorithm data extension in ni2 dimension (P)

Description Specifies how the specific LP (linear prediction) algorithm is to extend (or alter) forward or backward the time-domain data in the ni2 dimension. lpopt2 functions analogously to lpopt. Enter addpar('lp',2) to create lpopt2 and other ni2 dimension LP parameters in the current experiment.

Related	addpar	Add selected parameters to the current experiment (M)
	lpopt	LP algorithm data extension for np dimension (P)
	ni2	Number of increments in 2nd indirectly detected dimension (P)

lpprint LP print output for np dimension (P)

Description Controls LP (linear prediction) print output for the np dimension and creates an output file in the current experiment directory (curexp) with the name lpanalyz.out.1. Enter addpar('lp') to create lpprint and other np dimension LP parameters in the current experiment.

Values Comprised of sum of decimal values of the following bit fields, in which each bit field controls an independent output option:

- Bit 0 (decimal value 1) writes out the LP matrix and Y vector from which the LP coefficients are calculated.
- Bit 1 (decimal value 2) writes out the LP coefficients that have been obtained using either of the two supported algorithms.
- Bit 2 (decimal value 4) writes out the LP roots obtained from the characteristic polynomial derived from the LP coefficients; this only applies for lpalg='lpfft' and lpopt='f'.
- Bit 3 (decimal value 8) writes out the original and recalculated values for each LP extended (or altered) complex time-domain data point.
- Bit 4 (decimal value 16) writes out the internal LP parameter structure.

For example, lpprint=12 and lptrace=1 yields the following information in the file curexp/lpanalyz.out.1 for spectrum 1 along f_2: the values for all lpfilt complex LP coefficients and the original and recalculated values for each of the lpext LP extended (or altered) complex time-domain data points.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
curexp	Current experiment directory (P)
lpalg	LP algorithm in np dimension (P)
lpext	LP data extension in np dimension (P)
lpfilt	LP coefficients to calculate in np dimension (P)
lpopt	LP algorithm data extension for np dimension (P)
lpprint1	LP print output for ni dimension (P)
lpprint2	LP print output for ni2 dimension (P)
lptrace	LP output spectrum in np dimension (P)
np	Number of data points (P)

lpprint1 LP print output for ni dimension (P)

Description Controls LP (linear prediction) print output for the ni dimension and creates an output file in the current experiment directory (curexp) with the name lpanalyz1.out.1. lpprint1 functions analogously to lpprint. Enter addpar('lp',1) to create lpprint1 and other ni dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpprint	LP print output for np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lpprint2 LP print output for ni2 dimension (P)

Description Controls LP (linear prediction) print output for the ni2 dimension and creates an output file in the current experiment directory (curexp) with the name lpanalyz2.out.1. lpprint2 functions analogously to lpprint. Enter addpar('lp',2) to create lpprint2 and other ni2 dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpprint	LP print output for np dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

lptrace LP output spectrum in np dimension (P)

Description Specifies for which spectrum LP (linear prediction) output in the np dimension is produced in accordance with the parameter lpprint. Enter addpar('lp') to create lptrace and other np dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpalg	LP algorithm in np dimension (P)
lpprint	LP print output in np dimension (P)
lptrace1	LP output spectrum in ni dimension (P)
lptrace2	LP output spectrum in ni2 dimension (P)
np	Number of data points (P)

lptrace1 LP output spectrum in ni dimension (P)

Description Specifies for which spectrum or trace LP (linear prediction) output in the ni dimension is produced in accordance with the parameter lpprint1. lptrace1 functions analogously to lptrace. Enter addpar('lp',1) to create t lpprint2 and other ni dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpprint1	LP print output in ni dimension (P)
lptrace	LP output spectrum in np dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)

lptrace2 LP output spectrum in ni2 dimension (P)

Description Specifies for which spectrum or trace LP (linear prediction) output in the ni2 dimension is produced in accordance with the parameter lpprint2. lptrace2 functions analogously to lptrace. Enter addpar('lp',2) to create lptrace2 and other ni2 dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpprint2	LP print output in ni2 dimension (P)
lptrace	LP output spectrum in np dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

lres — Used to plot lineshape values (M)

Related	res	List files in directory (C)

ls — List files in directory (C)

Syntax `ls<(directory)>`

Description Lists the names of files in a directory on the text output window. `ls` is identical to `dir` and `lf`.

Arguments `directory` is the name of a directory. The default is the current working directory. `ls` is equivalent to the UNIX command `ls` and uses the same options (e.g., `–l` for a long listing such as `ls('-l *.fid')`).

Examples

```
ls
ls('data')
ls('-l *.fid')
```

Related	dir	List files in directory (C)
	lf	List files in directory (C)

lsfid — Number of complex points to left-shift the np FID (P)

Description Specifies number of complex points (not real points) that the `np` FID is to be either left-shifted (`lsfid>0`) or right-shifted (`lsfid<0`). A right shift adds zeros to the front of the FID. `lsfid` (and related parameters `phfid` and `lsfrq`) operate on complex `np` FID data, referred to as the t_2 dimension in a 2D experiment or as the t_3 dimension in a 3D experiment. `lsfid` is in the processing group and is properly handled by a `wti` operation (display).

Values `–fn/2` to `np/2` (or `–fn/2` to `fn/2` if `fn<np`), `'n'`

Related	dfid	Display a single FID (C)
	ds	Display a spectrum FID (C)
	fn	Fourier number in directly detected dimension (P)
	ft	Fourier transform 1D data (C)
	ft1d	Fourier transform along f_2 dimension (C)
	ft2d	Fourier transform 2D data (C)
	lsfid1	Number of complex points to left-shift `ni` interferogram(P)
	lsfid2	Number of complex points to left-shift `ni2` interferogram (P)
	lsfrq	Frequency shift of the `fn` spectrum in Hz (P)
	np	Number of data points (P)
	phfid	Zero-order phasing constant for the `np` FID (P)
	wft	Weight and Fourier transform 1D data (C)
	wft1d	Weight and Fourier transform f_2 of 2D data (C)

wft2d	Weight and Fourier transform 2D data (C)
wti	Interactive weighting (C)

lsfid1 Number of complex points to left-shift ni interferogram (P)

Description Specifies number of hypercomplex (for hypercomplex interferogram data) or complex (for complex interferogram data) points that the ni interferogram is to be either left-shifted (lsfid1>0) or right-shifted (lsfid1<0). A right shift adds zeros to the front of the FID. lsfid1 (and related parameters phfid1 and lsfrq1) operate on ni interferogram data, both hypercomplex and complex. ni interferogram data are referred to as the t_1 dimension in both a 2D and a 3D experiment. lsfid1 is in the processing group and is properly handled by a wti operation (display); that is, a wti operation on an ni interferogram applies the parameters phfid1, lsfid1, and lsfrq1, if selected, to the time-domain data prior to the Fourier transformation.

Values -fn1/2 to ni (or -fn1/2 to fn1/2 if fn1<2*ni), 'n'

Related

fn1	Fourier number in 1st indirectly detected dimension (P)
lsfid	Number of complex points to left-shift np FID (P)
lsfid2	Number of complex points to left-shift ni2 interferogram (P)
lsfrq1	Frequency shift of the fn1 spectrum in Hz (P)
ni	Number of increments in 1st indirectly detected dimension (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
wti	Interactive weighting (C)

lsfid2 Number of complex points to left-shift ni2 interferogram (P)

Description Specifies the number of hypercomplex (for hypercomplex interferogram data) or complex (for complex interferogram data) points that the ni2 interferogram is to be either left-shifted (lsfid2>0) or right-shifted (lsfid2<0). A right shift adds zeros to the front of the FID. lsfid2 (and related parameters phfid2 and lsfrq2) operate on ni2 interferogram data, both hypercomplex and complex. ni2 interferogram data are referred to as the t_2 dimension in a 3D experiment. lsfid2 is in the processing group and is properly handled by a wti operation (display).

Values -fn2/2 to ni2 (or -fn2/2 to fn2/2 if fn2<2*ni2), 'n'

Related

fn2	Fourier number in 2nd indirectly detected dimension (P)
lsfid	Number of complex points to left-shift np FID (P)

lsfid1	Number of complex points to left-shift ni interferogram(P)
lsfrq2	Frequency shift of the fn2 spectrum in Hz (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
wti	Interactive weighting (C)

lsfrq — Frequency shift of the fn spectrum (P)

Description Sets a frequency shift of spectral data, in Hz. lsfrq is the time-domain equivalent of lp within VnmrJ. lsfrq (and related parameters phfid and lsfid) operate on complex np FID data, referred to as the t_2 dimension in a 2D experiment or as the t_3 dimension in a 3D experiment. lsfrq is in the processing group and is properly handled by a wti operation (display).

Values A positive value results in peaks being shifted downfield (to the left). A negative value results in peaks being shifted upfield (to the right).

Related

dfid	Display a single FID (C)
ds	Display a spectrum FID (C)
fn	Fourier number in directly detected dimension (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
lp	First-order phase in directly detected dimension (P)
lsfid	Number of complex points to left-shift np FID (P)
lsfrq1	Frequency shift of the fn1 spectrum in Hz (P)
lsfrq2	Frequency shift of the fn2 spectrum in Hz (P)
phfid	Zero-order phasing constant for np FID (P)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 of 2D data (C)
wft2d	Weight and Fourier transform 2D data (C)
wti	Interactive weighting (C)

lsfrq1 — Frequency shift of the fn1 spectrum (P)

Description Sets a frequency shift of spectral data, in Hz. lsfrq1 is the time-domain equivalent of lp1 within VnmrJ. lsfrq1 (and related parameters phfid1 and lsfid1) operate on ni interferogram data, both hypercomplex and complex. ni interferogram data are referred to as the t_1 dimension in both a 2D and a 3D experiment. lsfrq1 is in the processing group and is properly handled by a wti operation (display); that is, a wti operation on an ni interferogram applies the parameters phfid1, lsfid1, and lsfrq1, if selected, to the time-domain data prior to the Fourier transformation.

Values A positive value results in peaks being shifted downfield (to the left). A negative value results in peaks being shifted upfield (to the right).

Related

fn1	Fourier number in 1st indirectly detected dimension (P)
lp1	First-order phase in 1st indirectly detected dimension (P)
lsfid1	Number of complex points to left-shift ni interferogram(P)
lsfrq	Frequency shift of the fn spectrum in Hz (P)
lsfrq2	Frequency shift of the fn2 spectrum in Hz (P)
ni	Number of increments in 1st indirectly detected dimension (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
wti	Interactive weighting (C)

lsfrq2 Frequency shift of the fn2 spectrum (P)

Description Sets a frequency shift of spectral data in Hz. lsfrq2 is the time-domain equivalent of lp2 within VnmrJ. lsfrq2 (and related parameters phfid2 and lsfid2) operate on ni2 interferogram data, both hypercomplex and complex. ni2 interferogram data is referred to as the t_2 dimension in a 3D experiment. lsfrq2 is in the processing group and is properly handled by a wti operation (display).

Values A positive value results in peaks being shifted downfield (to the left). A negative value results in peaks being shifted upfield (to the right).

Related

fn2	Fourier number in 2nd indirectly detected dimension (P)
lp2	First-order phase in 2nd indirectly detected dimension (P)
lsfid1	Number of complex points to left-shift ni interferogram (P)
lsfid2	Number of complex points to left-shift ni2 interferogram (P)
lsfrq	Frequency shift of the fn spectrum in Hz (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
wti	Interactive weighting (C)

lvl Zero-order baseline correction (P)

Description When spectral display is active, the command dc turns on a linear drift correction (baseline correction). The result of this operation includes calculating a zero-order baseline correction parameter lvl. This is done by averaging of a small number of points at either end of the display and drawing a straight line baseline between them.

Related	cdc	Cancel drift correction (C)
	lvltlt	Control sensitivity of lvl and tlt adjustments (P)
	tlt	First-order baseline correction (P)

lvltlt Control sensitivity of lvl and tlt adjustments (P)

Description Controls the sensitivity of the interactive lvl and tlt adjustments. lvltlt is in the “current” parameter set and is basically a multiplier for the sensitivity. If this parameter does not exist, it can be created by commands create('lvltlt') setgroup('lvltlt','display').

Values The default value is 1.0. Larger values make the adjustments larger. Smaller values make the adjustments smaller.

Related	create	Create new parameter in a parameter tree (C)
	ds	Display a spectrum (C)
	lvl	Zero-order baseline correction (P)

M

macro	Macro name (P)
macrocat	Display a user macro file in text window (C)
macrocp	Copy a user macro file (C)
macrodir	List user macro files (C)
macroedit	Edit a macro with user-selectable editor (M)
macrold	Load a macro into memory (C)
macrorm	Remove a user macro (C)
macrosyscat	Display a system macro file in text window (C)
macrosyscp	Copy a system macro to become a user macro (C)
macrosysdir	List system macros (C)
macrosysrm	Remove a system macro (C)
macrovi	Edit a user macro with the vi text editor (M)
make3dcoef	Make a 3D coefficients file from 2D coefficients (M)
makedosyparams	Create parameters for DOSY processing (M)
makefid	Make a FID element using numeric text input (C)
makeeccglobals	Create global parameters for ECC control (M)
makeslice	Synthesize 2D projection of 3D DOSY experiment (C)
makeStudy	Create and manage Study Clones. (M)
makeuser	Add a new Vnmr user account or update an existing Vnmr user account (U)
makeuserpsg	Compiles the user PSG sources and constructs the user PSG object library
man	Display online description of command or macro (M)
managedb	Update user files (U)
manualpath	Path to user's manual directory (P)
manvi	Edit online description of a command or macro (M)
mapwin	List of experiment numbers (P)

mark	Determine intensity of spectrum at a point (C)
masvt	Type of variable temperature system (P)
maxattench1-4	Maximum limit for attenuator setting for rf channel 1-4 (P)
maxpen	Maximum number of pens to use (P)
md	Move display parameters between experiments (C)
menu	Change status of menu system (C)
menuvi	Edit a menu with vi text editor (M)
method	Autoshim method (P)
mf	Move FIDs between experiments (C)
mfblk	Copy FID block (C)
mfclose	Close memory map FID (C)
mfdata	Move FID data (C)
mfopen	Memory map open FID file (C)
mftrace	Move FID trace (C)
mht	Move Hadamard parameters from one workspace to another
minsw	Reduce spectral width to minimum required (M)
mkCPprotocol	Make Protocol
mkdir	Create new directory (C)
mlabel	Menu label (P)
move	Move to an absolute location to start a line (C)
movedssw	Set down sampling parameters for selected spectral region (M)
moveossw	Set over sampling parameters for selected spectral region (M)
movesw	Move spectral window according to cursors (M)
movetof	Move transmitter offset (M)
mp	Move parameters between experiments (C)
mparval	Moves a Paramter Value Between Experiments
mqcosy	Set up parameters for MQCOSY pulse sequence (M)
mrev8	Set up parameters for MREV8 pulse sequence (M)
mrfb	Set the filter bandwidths for multiple receivers (P)
mref	Set referencing based on a existing spectrum of the sample (M)
mrgain	Set the gain for multiple receivers (P)
mstat	Display memory usage statistics (C)

mstring	Menu string (P)
mtune	Tune probe using swept-tune graphical display (M)
mv	Move and/or rename a file (C)
mvsampglobal	Moves sample global parameters
mxconst	Maximum scaling constant (P)
mz	Move Integral Reset Points to specified experiment

macro Macro name (P)

Description A string parameter, available in each experiment, similar to the n1, n2, and n3 parameters. Certain macros, such as h1p, need to know which macro invoked them. This parameter is used to pass that information.

See also *User Programming*

Related h1p Process simple proton spectra from h1 macro (M)
n1,n2,n3 Name storage for macros (P)

macrocat Display a user macro file in text window (C)

Syntax `macrocat(file1<,file2><,...>)`

Description Displays one or more user macro files in the text window.

Arguments `file1, file2, ...` are the names of macros in the user macro library.

Examples
```
macrocat('build')
macrocat('dan','george')
```

See also *User Programming*

Related macrodir List user macros (C)
macrosyscat Display a system macro file in text window (C)

macrocp Copy a user macro file (C)

Syntax `macrocp(from_file,to_file)`

Description Makes a copy of the existing user macro file and places the copy in the user's macro library. Using `macrocp` to make a backup copy is the recommended procedure to modify a macro but still be able to revert to the previous version if you are unsure about the modification. `macrocp` can also be useful for writing a new macro that is very similar to an existing macro.

Arguments `from_file` is the name of an existing user macro file to be copied. The file must be in the user's macro library.

`to_file` is the file name to be given to the copy. This name must be different from the name of the original macro.

Examples `macrocp('dan','dan.old')`

See also *User Programming*

Related	`macrocat`	Display a user macro file in text window (C)
	`macrodir`	List user macros (C)
	`macrosyscp`	Copy a system macro to become a user macro (C)

`macrodir` List user macro files (C)

Description Lists the names of user macro files in the user's macro library.

See also *User Programming*

Related `macrosysdir` Lists system macros (C)

`macroedit` Edit a macro with user-selectable editor (M)

Syntax `macroedit(file)`

Description Opens a MAGICAL macro file from a user's personal macro library for editing (if you want to edit a system macro, copy it to a personal library and then use `macroedit`).

The default editor is `vi`. To select another editor, first set UNIX environmental variable `vnmreditor` to the name of the editor; that is, in the `.login` file, change the line

`setenv vnmreditor old_ed`

to become

`setenv vnmreditor new_ed` (e.g., `setenv vnmreditor emacs`).

Second, make sure a script with the prefix `vnmr_` followed by the name of the editor is placed in the `bin` subdirectory of the VnmrJ system directory (e.g., `vnmr_emacs`).

The script file makes adjustments for the type of graphic interface in use. Scripts provided in the software include `vnmr_vi` and `vnmr_textedit.` To create other scripts, refer to the `vnmr_vi` script for non-window editor interfaces or refer to `vnmr_textedit` for window-based editor interfaces.

Arguments `file` is the name of the macro file you wish to edit.

Examples `macroedit('pa')`

See also *User Programming*

Related

paramedit	Edit a parameter and its attributes with user-selected editor (C)
paramvi	Edit a parameter and its attributes with *vi* editor (M)
edit	Edit a file with user-selectable editor (C)
macrovi	Edit a user macro with vi editor (M)
menuvi	Edit a menu with the vi editor (M)
textvi	Edit text file of current experiment with *vi* editor (M)

macrold Load a macro into memory (C)

Syntax
```
macrold(file)<:dummy>
macrold('macrofile')
macrold('_sw'):$ret
macrold('_sw'):$ret,$msg
```

Description Loads a macro, user or system, into memory. If the macro already exists in memory, it is overwritten by the new macro. Loading a macro into memory increases the execution speed of the macro. The trade-off is that the macro uses memory. The mstat command displays macros that have been loaded into memory. One or more individual macros, or all the macros loaded in memory, can be removed from memory with the purge command.

A return value from macrold will be set to 1 if the load succeeded and it will be set to 0 if the load failed. Requesting a return value from macrold will also suppress the message that the macro is loaded. If the macro failed to load, a message about the failure will be displayed. If a second return argument is requested, possible failure messages will also be suppressed. The suppressed messages will be put into the second return argument.

If a macro already loaded into memory is edited using macrovi or macroedit, the changed macro automatically is loaded by those macros. This overwrites the previous macro. However, if a macro is edited or created some other way (with macrocp perhaps), the changed version is not automatically loaded. If the macro already exists in memory, the previous version executes unless the user runs macrold.

Arguments file is the name of the macro file to be loaded into memory. For loading macros, the same search path is used as when deciding which macro to execute. That is, the user's private maclib directory is searched first and finally the system maclib. If an absolute path is supplied as the file argument, that macro is loaded. This allows macros not in a maclib to be loaded and executed from VnmrJ.

dummy is any throwaway variable. Requesting a return value suppresses the message in the status window (line 3) that the macro is loaded.

Examples
```
macrold('pa')
macrold('_sw'):$noline3
```

See also	*User Programming*	
Related	macrocp	Copy a user macro file (C)
	macroedit	Edit a macro with user-selectable editor (M)
	macrovi	Edit a user macro with the `vi` text editor (M)
	mstat	Display memory usage statistics (C)
	purge	Remove macros from memory (C)

macrorm Remove a user macro (C)

Syntax	`macrorm(file)`
Description	Removes a user macro from the user's macro directory. If the macro has already been loaded in memory, it remains in memory until a new macro of the same name is loaded or the program exits.
Arguments	`file` is the name of the user macro to be removed.
Examples	`macrorm('pa')`
See also	*User Programming*

Related	delcom	Delete a user macro (M)
	macrodir	List user macros (C)
	macrosysrm	Remove a system macro (C)
	purge	Remove all macros from memory (C)

macrosyscat Display a system macro file in text window (C)

Syntax	`macrosyscat(file1<,file2><,...>)`
Description	Displays one or more system macro files in the text window.
Arguments	`file1, file2, ...` are names of macros in the system macro library.
Examples	`macrosyscat('build')` `macrosyscat('dan','george')`
See also	*User Programming*

Related	macrocat	Display a user macro file in text window (C)
	macrosysdir	Lists system macros (C)

macrosyscp Copy a system macro to become a user macro (C)

Syntax	`macrosyscp(from_file,to_file)`
Description	Makes a copy of the existing system macro file and places the copy in the user's macro library. This is the recommended way to modify a system macro for personal use.

Arguments `from_file` is the name of an existing system macro file to be copied. The file must be in the system macro library.

`to_file` is the file name to be given to the copy. In this case, the name of the copied macro can be the same as the original macro. In many cases, it is the same, allowing the user to have a personal macro of the same name as the system macro but which will override the system macro.

Examples `macrosyscp('pa','pa')`
`macrosyscp('pa','mypa')`

See also *User Programming*

Related
macrocp	Copy a user macro file (C)
macrosyscat	Display a system macro file in text window (C)
macrosysdir	Lists system macros (C)

macrosysdir List system macros (C)

Description Lists the names of system macros in the system macro library.

See also *User Programming*

Related
macrodir	List user macros (C)

macrosysrm Remove a system macro (C)

Syntax `macrosysrm(file)`

Description Removes a system macro file from the system macro directory. If the macro has already been loaded in memory, it remains in memory until a new macro of the same name is loaded or the program exits.

Arguments `file` is the name of the system macro file to be removed.

Examples `macrosysrm('pa')`

See also *User Programming*

Related
macrorm	Remove a user macro (C)
macrosysdir	Lists system macros (C)
purge	Remove all macros from memory (C)

macrovi Edit a user macro with the vi text editor (M)

Syntax `macrovi(file)`

Description Initiates creating a new user macro or modifying an existing user macro using the UNIX `vi` text editor. On the Sun workstation, a pop-up window contains the edit. On the GraphOn, the edit is done on the entire terminal. To edit a system macro, first copy the macro to a personal library and then edit it using macroedit or macrovi.

Arguments file is the name of an existing user's macro to be edited or the name of a new user's macro to be created.

Examples macrovi('pa')

See also *User Programming*

Related

macroedit	Edit a macro with a user-selectable editor (C)
vi	Edit text file with vi text editor (C)

make3dcoef Make a 3D coefficients file from 2D coefficients (M)

Syntax make3dcoef<('t1t2'|'t2t1')>

Description Makes a 3D coefficients file from 2D coefficients and writes the file in the path stored by curexp. 2D coefficients are supplied as strings in the parameters f2coef and f1coef. This macro is capable of handling 3D data collected with any number of data sets (e.g., TPPI, Hypercomplex, Rance SE, Kay SE, and phase-sensitive gradient in one or both dimensions). make3dcoef is called by the ft3d macro.

The 2D coefficients are supplied as strings in f1coef and f2coef. These coefficients are the same as found by processing with wft2d(2dcoefs). Note that wft2da (for States-Hypercomplex method) is equivalent to wft2d(1,0,0,0,0,0,-1,0), and that wft2d (for absolute-value mode) is equivalent to wft2d(1,0,0,-1).

Coefficients are separated by spaces and not commas. For example, if a 3D data set collected by the States-Hypercomplex method in both ni and ni2 dimensions, f1coef='1 0 0 0 0 0 -1 0' and f2coef='1 0 0 0 0 0 -1 0'. And if a 3D data set collected in absolute-value mode in both ni and ni2 dimensions, f1coef='1 0 0 -1' and f2coef='1 0 0 -1'.

The f1coef and f2coef parameters are created by the par3d macro. Execution of make3dcoef when f1coef and f2coef have no value or inconsistent values causes the macro to abort, which enables the user to enter these values and reexecute the macro. For example, the value of f1coef when the F1 dimension can be processed with wft2da is '1 0 0 0 0 0 -1 0'. The value of f2coef when the F2 dimension can be processed with wft2d(1,0,1,0,0,-1,0,1) is '1 0 1 0 0 -1 0 1'.

The parameters f1coef and f2coef must be 2D coefficients that give proper ni and ni2 first planes with the same rp (assuming lp is 0 by using calfa) values. For example, processing the phase-sensitive gradient dimension should not be done with 1 0 0 1 0 1 1 0 and applying 45° phase shifts to rp, but with 1 0 1 0 0 1 0 -1, or its variant, that gives the same rp value as the other dimension. This also applies to Rance-type or Kay-type sensitivity-enhanced dimensions.

Note that sensitivity-enhanced sequences (gradient or otherwise) can be processed two different ways to give “orthogonal” data sets. The coefficients must be picked so that they have the same rp as the other dimension.

This macro can also handle coefficients that are not 1s or 0s. For example, if processing requires that a data set contributes to the interferogram after a 30°phase shift, cos(30) and sin(30) can be selected as the real and imaginary contributions, respectively, during the construction of the interferogram.

Arguments 't1t2' means array='phase,phase2' in simple hypercomplex data sets. It means array='t1related','t2related' with multiple sets in general.

't2t1' means array='phase2,phase' in simple hypercomplex data sets. It means array='t2related','t1related' with multiple sets in general.

If no argument is used and if array='phase,phase2' or array='phase2,phase, the macro automatically decides on 't1t2' or 't2t1', respectively.

See also *NMR Spectroscopy User Guide*

Related

array	Parameter order and precedence (P)
calfa	Recalculate alfa so that first-order phase is zero (M)
curexp	Current experiment directory (P)
f1coef	Coefficient to construct F1 interferogram (P)
f2coef	Coefficient to construct F2 interferogram (P)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M)
lp	First-order phase in directly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
ntype3d	Specify whether f_1 or f_2 display expected to be N-type (P)
rp	Zero-order phase in directly detected dimension (P)
wft2d	Weight and Fourier transform 2D data (C)
wft2da	Weight and Fourier transform phase-sensitive data (M)

makedosyparams Create parameters for DOSY processing (M)

Syntax `makedosyparams(dosytimecubed,dosyfrq)`

Description This macro is automatically called by the Dbppste, DgcsteSL, Doneshot, Dbppsteinept, Dgcstecosy, and Dgcstehmqc sequences to create the parameters dosyfrq, dosygamma, and dosytimecubed, which are necessary for the dosy analysis. Do not manually run makedosyparams.

See also *NMR Spectroscopy User Guide*

Related	
dosy	Process DOSY experiments (M)
dosyfrq	Larmor frequency of phase encoded nucleus in DOSY (P)
dosygamma	Gyromagnetic constant of phase encoded nucleus in DOSY (P)
dosytimecubed	Gyromagnetic constant of phase encoded nucleus in DOSY (P)

makefid — Make a FID element using numeric text input (C)

Syntax `makefid(file<,element_number<,format>)`

Description Creates FID files that can be used to introduce computed data into an experiment. The number of points comes from the number of numeric values read from the input file. If the current experiment already contains a FID, you will not be able to change either the format or the number of points from that present in the FID file. Use `rm(curexp+'/acqfil/fid')` to remove the FID.

The `makefid` command does not look at parameter values when establishing the format of the data or the number of points in an element. Thus, if the FID file is not present, it is possible for `makefid` to write a FID file with a header that does not match the value of `dp` or `np`. Because the active value is in the processed tree, you need to use the `setvalue` command if any changes are required.

Arguments `file` is the name of the input file. It contains numeric values, two per line. The first value is assigned to the X (or real) channel; the second value on the line is assigned to the Y (or imaginary) channel.

`element_number` is the number of the element or FID and is any integer larger than 0. The default is the first element or FID. If the FID element already exists in the FID file, the program overwrites the old data.

`format` is a character string with the precision of the resulting FID file and can be specified by one of the following strings:

'dp=n'	single-precision (16-bit) data
'dp=y'	double-precision (32-bit) data
'16-bit'	single-precision (16-bit) data
'32-bit'	double-precision (32-bit) data

If an FID file exists, `makefid` uses the same `format` string for precision; otherwise, the default is double-precision (32-bit) data.

`element_number` and `format` arguments can be entered in any order.

Examples `makfid('fid.in',2,'32-bit')`

See also *NMR Spectroscopy User Guide*; *User Programming*

Related	
cp	Copy a file (C)
curexp	Current experiment directory

dp	Double precision (P)
mv	Move and/or rename a file (C)
np	Number of data points (P)
rm	Delete file (C)
setvalue	Set value of any parameter in a tree (C)
writefid	Write numeric text file using a FID element (C)

makeeccglobals Create global parameters for ECC control (M)

Applicability Systems with Varian, Inc. Cold Probes

Description Creates the following nine global parameters required for ECC control by PSG:
`tc1z`, `tc2z`, `tc3z`, `tc4z`, `amp1z`, `amp2z`, `amp3z`, `amp4z`, and `chiliConf`

Related chiliConf

makeslice Synthesize 2D projection of 3D DOSY experiment (C)

Syntax `makeslice(<option>,lowerlimit,upperlimit)`

Arguments `option` is either `'i'` or `'s'`.

`'i'` includes the “tails” of diffusion peaks that lie outside the range between `lowerlimit` and `upperlimit`. The default is `'i'`.

`'s'` only includes the integration peaks whose diffusion coefficient lies between the specified limits.

`lowerlimit` is the lower diffusion limit (in units of 10^{-10} m^2/s) to be displayed.

`upperlimit` is the upper diffusion limit (in units of 10^{-10} m^2/s) to be displayed.

Description Synthesizes an integral projection between specified diffusion limits of a 3D DOSY spectrum onto the frequency-frequency plane. `makeslice` requires the first 2D increment of the 3D DOSY data to have been transformed.

See also *NMR Spectroscopy User Guide*

Related

dosy	Process DOSY experiments (M)
showoriginal	Restore first 2D spectrum in 3D DOSY spectrum (M)

makeStudy Create and manage Study Clones.

Syntax

Applicability VnmrJ 3.1

Description Do not use this macro from the command line.

See also User Guide: Automation-Clone a New Study

makeuser Add a new Vnmr user account or update an existing Vnmr user account (U)

Syntax `makeuser`

Applicability VnmrJ 3.1

Description The `makeuser` command is provided to create a new user account with permission to access the VNMR files and programs. The `makeuser` command will also install the necessary files and directories into the user's home directory.

The `makeuser` command can be run by the system administrator's root account or by any current user with appropriate permissions. In order to add a new user to the system, `makeuser` must be run by `root`.

When `root` executes `makeuser`, the location of the VNMR system directory, if not available from the vnmrsystem environmental parameter, will be requested. The most likely location, which is provided as the default, is /vnmr. When executed as root, the user name whose account is to be added or updated may be supplied as an argument to makeuser. If no name is supplied, one is requested. The makeuser script checks to see if the user is already defined in the system. If not, the /etc/passwd file will be updated with the new user and the user will be added to the `nmr` group in the `/etc/group` file. A home directory will also by made. The location of the home directory is encoded in the makeuser script and may be altered if desired. By convention, on Sun systems, the home directory is made in the /home directory. Also by convention, on IBM systems, the home directory is in the /u directory. Note that the makeuser script updates the /etc/passwd and /etc/group files on the local machine. If Network Information Services (formerly known as Sun Yellow Pages) is running, this may not be the correct thing to do. In this case, the script could be executed on the host which is the password server. Additional steps may be required to make the new account available over the network. Refer to Sun documentation for this information.

The above operations require root privileges to execute. After finishing these tasks, the makeuser script gives root an opportunity to exit from the script.

A user other than root can run the makeuser script. In this case, the makeuser script will only update the current user's files. If one tries to update another account, an error will be reported. If a user runs the makeuser script (or the root account decided not to exit when given the above opportunity), the following question will appear

Automatically configure the 'user' account (y or n) where user is replaced by the user's name. If this is answered with a y, then the necessary UNIX files and Vnmr files will be added to the account and various Vnmr subdirectories will be made. The next time the user logs out and then logs in, Vnmr will automatically start. After logging out and then logging in, whenever vnmr is typed, VNMR will start. If the above question is answered with an n, meaning one does not want all the files automatically updated, then two additional questions are asked.

- Automatically configure UNIX environment (.files) (y or n)
- Automatically configure Vnmr directories and global parameters (y or n)

Answering n to one or both of these questions invokes an interactive mode where one is asked

OK to update Unix_file (y or n)

for the various UNIX dotfiles (for example, .login, .cshrc, .Xdefaults, etc).

For the Vnmr related files and subdirectories, one is asked

- Create subdirectory of your VNMR user directory (y or n)
- Update your VNMR global parameters (y or n)

If one decides not to do the automatic configuration of the UNIX dotfiles, then samples of what is required for proper functioning of Vnmr is defined in the file .xlogin in the user_templates subdirectory of the Vnmr system directory. The actual UNIX dotfiles which are used for the automatic update are also stored in this same user_templates directory. he Vnmr global file is also stored there.

makeuserpsg Compiles the user PSG sources and constructs the user PSG object library

Syntax

Applicability VnmrJ 3.1

Description MAKEUSERPSG is a UNIX makefile which is invoked by the shellscript PSGGEN. MAKEUSERPSG has the following attributes:

- All compilation and library construction is performed in the user PSG directory;
- Any additional source (*.c) and header files (*.h and *.p) and the makefile itself, unless already present, are linked from the system PSG directory into the user PSG directory via soft links;
- The three possible names for the user PSG object library are LIBPSGLIB.A, LIBPSGLIB_FPC.A, and LIBPSGLIB_FPA.A. The first name is used only for Sun 4 systems. The last two names are used for Sun 3 systems with SEQGEN_OPTION set to f68881 or ffpa, respectively.

MAKEUSERPSG currently has no error recovery. Therefore, if an error occurs, the user PSG directory will not be cleaned up, i.e., the soft links to files in the system PSG directory will remain in this directory along with any object file previously created by the make-file.

man — Display online description of command or macro (M)

Syntax man('file')<:$return>

Displays a description of commands and macros from files in the applications directory. The manual file is displayed in the text window when it is retrieved by the man macro. The man macro aborts if a name is not supplied as an argument.

Arguments file — name of a command or macro in one of the applications directories.

:$res — supply a return argument to suppress messages if the manual page does not exist.

Examples man('mark')

man('notAcommand'):$res

See also *NMR Spectroscopy User Guide; User Programming*

Related

manvi	Edit online description of a command or macro (M)
manualpath	Path to user's manual directory (P)

managedb — Update user files (U)

Syntax managedb update

Description Updates VnmrJ database for the Locator.

See also *NMR Spectroscopy User Guide*

manualpath — Path to user's manual directory (P)

Description Contains the absolute path to a user's directory of VnmrJ manual entries. If manualpath exists for a user, it must be defined in the user's global parameter file. Enter create('manualpath','string','global') to create the manualpath parameter.

See also *User Programming*

Related

man	Display online description of a command or macro (M)

manvi Edit online description of a command or macro (M)

Syntax `manvi('file')`

Description Enables editing or creating an online description of commands and macros stored in any of the applications directories for which the user has write permission.

Arguments `file` is the name of a command macro.

Examples `manvi('mark')`

See also *User Programming*

Related `man` Display online description of command or macro (M)

mapwin List of experiment numbers (P)

Description Arrayed global parameter that maintains a list of experiment numbers for the window panes in the VnmrJ graphics window.

Related
`curwin` Current window (P)
`fontselect` Open FontSelect window (C)
`jwin` Activate current window (M)
`setgrid` Activate selected window (M)
`setwin` Activate selected window (C)

mark Determine intensity of spectrum at a point (C)

Syntax
(1) `mark<(f1_position)><:intensity>`
(2) `mark<(left_edge,region_width)><:intensity, integral>`
(3) `mark<(f1_position,f2_position)><:intensity>`
(4) `mark<(f1_start,f1_end,f2_start,f2_end)> <:intensity,integral,c1,c2>`
(5) `mark<('trace',<options>)><:intensity,integral, c1,c2>`
(6) `mark('reset')`

Description Find the intensity of a spectrum at a point. Either 1D or 2D operations can be performed in the cursor or box mode for a total of four separate functions: 1D operations in cursor mode (syntax 1), 1D operations in box mode (syntax 2), 2D operations in cursor mode (syntax 3) and 2D operations in box mode (syntax 4).

In the *cursor mode*, the intensity at a particular point is found. In the *box mode*, the integral over a region is calculated. The displayed integral is scaled in the same way as output from `dli` is scaled; that is, by the ins and insref parameters. For 2D operations, this is the volume integral and the volume is scaled by ins2 and ins2ref. In addition, the mark command in the box mode finds the maximum intensity and the coordinate(s) of the maximum intensity.

The mark command requires that transformed data be present in the current experiment. If required, it recomputes the phase file from the complex data (i.e., it rephases the data if required); however, the mark command requires parameters from the command line if no data is displayed (i.e., if ds or dconi has not been executed).

Note that 2D operations require that 2D data be present. This not only means that ni must be larger than 1, but also that the data was transformed using ft1d, ft2d or an equivalent (and not ft or its equivalents).

The mark command, as well as the MARK button of ds, writes output to a file in the current experiment. For 1D operations, the file is named mark1d.out; for 2D operations, it is mark2d.out. If this file already exits, VnmrJ appends output from the current mark operation to the end of the file. (Older versions of VnmrJ used ds.out and dconi.out as files for output from the MARK button). Either file can be read by other programs at any time between operations.

The following criteria establish the exact function. The command checks them in the following order until it determines the exact function:

1. Number of numeric parameters.
2. Number of return values called out.
3. Which display command (ds or dconi) was last used.
4. Nature of the data in the experiment.

The first two criteria only serve to distinguish between box mode and cursor mode. The nature of the data in the experiment and the last display command entered determines whether a 1D or a 2D operation is selected.

Arguments

f1_position defines the position, in Hz, along the f_1 axis in the 1D and 2D cursor modes. The default is cr (1D) or cr1 (2D).

left_edge defines the position of the left edge of the region, in Hz, to be integrated in 1D box mode. The default is cr.

region_width defines the width, in Hz, of the region, which extends to the right of left_edge, in 1D box mode. The default is delta.

f2_position defines the position, in Hz, along the f_2 axis in the 2D cursor mode. The default is delta1.

f1_start and f1_end define region along the f_1 axis in the 2D box mode.

f2_start and f2_end define region along the f_2 axis in the 2D box mode.

'trace' is a keyword to select a 1D operation if 2D data is present. It must be either the first or the last argument (e.g., mark('trace',400) determines the intensity at 400 Hz in the current trace).

`'reset'` is a keyword to erase the output files from the `mark` command. No other argument can be used with this keyword. Use `rename` to rename the current `mark` output files (e.g., `rename(curexp+'/mark1d.out', curexp+'/mark.16.01.89')`)

`intensity` is a return value set to the intensity of the spectrum at the point for either 1D or 2D operations (the maximum if cursor mode was selected).

`integral` is a return value set to the integral of the spectrum at the point. `integral` is not returned in the cursor mode.

`c1,c2` are return values set to the coordinates where the maximum intensity was found in 2D mode. `c1` and `c2` are not returned in the cursor mode.

Examples 1D data sets:

```
mark(cr)                        cursor mode for 1D data
mark(cr,delta)                  box mode for 1D data
```

2D data sets (2D mode): In this mode, the order of the arguments to mark is independent of the `trace` parameter.

```
mark(cr1,cr)                    cursor mode for 2D data
mark(cr1,delta1,cr,delta)       box mode for 2D data
```

2D data sets (1D mode): In this mode, the selection of the arguments to mark is dependent on the `trace` parameter. If `trace='f2'`, then `cr`, `delta`, `sp`, or `wp` are appropriate. If `trace='f1'`, then `cr1`, `delta1`, `sp1`, and `wp1` are appropriate.

```
mark('trace',cr)                cursor mode for selected 2D trace
mark('trace',cr1,delta1)        box mode for selected 2D trace
```

Alternate: MARK button in the `ds` program.

See also *NMR Spectroscopy User Guide; User Programming*

Related

`cr`	Cursor position in directly detected dimension (P)
`cr1`	Cursor position in 1st indirectly detected dimension (P)
`curexp`	Current experiment directory (P)
`dconi`	Interactive 2D contour display (C)
`delta`	Difference of two frequency cursors (P)
`dli`	Display list of integrals (C)
`ds`	Display a spectrum (C)
`ft1d`	Fourier transform along f_2 dimension (C)
`ft2d`	Fourier transform 2D data (C)
`ins`	Integral normalization scale (P)
`ins2`	2D volume value (P)
`insref`	Fourier number scaled value of an integral (P)
`ins2ref`	Fourier number scaled volume of a peak (P)
`mv`	Move and/or rename a file (C)
`ni`	Number of increments in 1st indirectly detected dimension (P)

masvt Type of variable temperature system (P)

Description Identifies the type of VT system in use: the standard Oxford VT controller or the Oxford-Sorenson or solids VT controller system (used with the Varian VT CP/MAS probe). masvt is a global parameter that is active on all of each user's experiments on a per user account basis. The current value of the parameter can be displayed by typing masvt?.

Note that the VT Controller option displayed by config must be set to Present for either VT controller system to be active. If masvt does not exist, it can be created with the command create('masvt','string','global').

The new Highland VT controller is autosensing, making masvt superfluous for systems with this controller.

Values 'y' indicates the solids VT system is in use.

'n', any other value but 'n' and 'y', or if masvt does not exist, indicate that the Oxford Varian VT controller, if present, is in use.

See also *VnmrJ Installation and Administration*

Related

config	Display current configuration and possibly change values (M)
create	Create a new parameter in a parameter tree (C)
vttype	Variable temperature controller present (P)

maxattench1-4 Maximum limit for attenuator setting for rf channel 1-4 (P)

Description maxattench1, maxattench2, maxattench3, and maxattench4, are optional global parameters for the limiting the maximum attenuator settings for rf channel 1, channel 2, channel 3, and channel 4 (respectively) from pulse sequence statements and through tpwr/dpwr/... settings on go command. If maxattench2 is present, the attenuator setting check will be carried out by SpinCAD and C psg. If the attenuator setting exceeds the limit set in maxattench2, psg aborts with error message. This command is only applicable for check during the go command.

See also *SpinCAD*

maxpen Maximum number of pens to use (P)

Description Controls the maximum number of pens that will be used.

Values 1 to the number of pens in the system plotter. If maxpen=x and the software attempts to use pen x+y, it uses pen y instead.

See also *NMR Spectroscopy User Guide*

Related	pen	Select a pen or color for drawing (C)
	setpen	Set maximum number of HP plotter pens (M)

md — Move display parameters between experiments (C)

Syntax `md(<from_exp,>to_exp)`

Description Moves the saved display parameters from one experiment to another. These parameters must have been saved with the s command (e.g., s2).

Arguments `from_exp` specifies the number of the experiment, 1 through 9, from which the parameters are to be taken. The default is that the parameters are moved from the currently active experiment.

`to_exp` specifies to which experiment the parameters are to be moved.

Examples
```
md(4)
md(2,3)
```

See also *NMR Spectroscopy User Guide*

Related	mf	Move FIDs between experiments (C)
	mp	Move parameters between experiments (C)
	s	Save display parameters as a set (M)

menu — Change status of menu system (C)

Syntax (1) `menu(menu_name)`
(2) `menu<('off')>`

Description The VNMR menu system allows up to eight buttons to be active at a time, enabling the user to perform most actions with the mouse rather than typing in commands. All menus are stored in the library `menulib` in the system directory or in the user's `menulib`. See menuvi to change these menus.

If the menu system becomes deactivated for some reason, select the Menu On button in the Permanent Menu to reactivate it. Entering `menu('main')` also works.

Arguments `menu_name` is the name of the file controlling the menu (e.g., `'main'`). Including this argument activates the menu system and displays the menu controlled by `menu_name`.

`'off'` is a keyword to turn off the menu system.

Examples
```
menu
menu('fitspec')
menu('off')
```

See also *User Programming*

Related	menuvi	Edit a menu with the `vi` text editor (M)

mlabel	Menu label (P)
newmenu	Select a menu without immediate activation (C)

menuvi Edit a menu with vi text editor (M)

Syntax `menuvi(menu)`

Description Edits a Classic VNMR menu file using the UNIX `vi` text editor. On the Sun workstation, a pop-up window contains the edit. On the GraphOn, the edit is done on the entire terminal.

Arguments `menu` is the name of file controlling a menu.

Examples `menuvi('display_1D')`

See also *User Programming*

Related

menu	Change status of menu system (C)
newmenu	Select a menu without immediate activation (C)
vi	Edit text file with `vi` text editor (C)

method Autoshim method (P)

Description Selects the method for automatic shimming. Refer to the manual *NMR Spectroscopy User Guide* for information on how to write or alter methods.

Values Name of file in the `/vnmr/shimmethods` library for one of the defined shim methods in the system. To display all available methods, enter `ls('/vnmr/shimmethods')`. Standard methods include `'z1z2'` (selects shimming of the Z1 and Z2 gradients) and `'allzs'` (selects shimming all spinning gradients, Z1 to Z4 or Z5, depending on the magnet type). Shim methods can also be stored in a user's `shimmethods` directory (e.g., `/home/vnmr1/vnmrsys/shimmethods`).

See also *NMR Spectroscopy User Guide*

Related

ls	List files in current directory (C)
newshm	Interactively create a shim method with options (M)
stdshm	Interactively create a shim method (M)

mf Move FIDs between experiments (C)

Syntax `mf(<from_exp,>to_exp)`

Description Moves the last acquired FID, as well as its associated parameters, from one experiment to another. The text, the processed acquisition parameters and the current display and processing parameters are also moved to the specified experiment.

Arguments `from_exp` specifies number of the experiment from which the FID is to be taken. The default is the FID is moved from the currently active experiment.

`to_exp` specifies to which experiment the FID is to be moved.

Examples `mf(4)`
`mf(3,2)`

See also *NMR Spectroscopy User Guide*

Related
md	Move display parameters between experiments (C)
mp	Move parameters between experiments (C)

mfblk Copy FID block (C)

Syntax `mfblk(<src_expno,>src_blk_no,dest_expno,dest_blk_no)`

Description Copies data from a source FID block specified by `src_blk_no` to a destination FID block specified by `dest_expno` and `dest_blk_no`, using memory-mapped input and output.

`mfblk` searches for the source and destination FID file in the directory `$vnmruser/expN/acqfil`, where `N` is the requested experiment number or the current experiment number. If the FID file is not open, `mfblk` opens the file, copies the data, and closes the file. If a number of blocks need to be copied, explicitly opening and closing the files with the commands `mfopen` and `mfclose` can significantly speed up the data reformatting process.

`mfblk` can also be used to append blocks of data to a FID file by specifying that the `dest_blk_no` is greater than the number of blocks in a file.

Be aware that `mfblk` can modify data returned to an experiment with the `rt` command. To avoid modification, enter the following sequence of VnmrJ commands before running `mfblk`:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments `src_expno` specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

`src_blk_no` specifies the source block of data to be copied. Block numbers start at 1 and run from 1 to the number of blocks in a file.

`dest_expno` specifies the experiment number of the destination FID file.

`dest_blk_no` specifies the destination block to send the copied data.

Examples `mfblk(1,2,1)` copies current experiment, block 1 to exp 2, block 1.
`mfblk(3,2,6,2)` copies exp 2, block 2 to exp 6, block 2.

See also *User Programming*

Related
mfclose	Memory map close FID file (C)
mfdata	Move FID data (C)

mfopen	Memory map open FID file (C)
mftrace	Move FID trace (C)

mfclose Close memory map FID (C)

Description Closes experiment source and destination FID files that have been explicitly opened with mfopen.

See also *User Programming*

Related

mfblk	Move FID block (C)
mfdata	Move FID data (C)
mfopen	Memory map open FID file (C)
mftrace	Move FID trace (C)
rfblk	Reverse FID block (C)
rfdata	Reverse FID data (C)
rftrace	Reverse FID trace (C)

mfdata Move FID data (C)

Syntax

```
mfdata(<src_expno,>src_blk_no,src_start_loc,  \
  dest_expno,dest_blk_no,dest_start_loc,num_points)
```

Description Copies data specified by src_start_loc from a FID block specified by src_blk_no to a destination location specified by dest_expno, dest_blk_no, and dest_start_lo, using memory-mapped input and output. The data point locations and the num_points to be copied are specified by data points corresponding to the np parameter, not bytes or complex points.

mfdata searches for the source and destination FID file in th directory $vnmruser/expN/acqfil, where N is the requested experiment number or the current experiment number. If the FID file is not open, mfdata opens the file, copies the data, and closes the file. If a number of blocks need to be copied, explicitly opening and closing the files with the commands mfopen and mfclose can significantly speed up the data reformatting process.

Be aware that mfdata can modify data returned to an experiment with the rt command. To avoid modification, enter the following sequence of VnmrJ commands before running mfdata:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments src_expno specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

src_blk_no specifies the source block of data to be copied. Block numbers start at 1 and run from 1 to the number of blocks in a file.

src_start_loc specifies the starting data location within the specified block to copy the data. Data locations start from 0 and are specified as data points corresponding to the np parameter.

dest_expno specifies the experiment number of the destination FID file.

dest_blk_no specifies the destination block to send the copied data.

dest_start_loc specifies the starting data destination location within the specified block to send the copied data.

Examples mfdata(1,0,2,1,(nv-1)*np,np) copies np points of data from the starting location 0 of block 1 of the current experiment to the data location (nv-1)*np of block 1 of experiment 2.

See also *User Programming*

Related

mfblk	Move FID block (C)
mfclose	Memory map close FID file (C)
mfdata	Move FID data (C)
mfopen	Memory map open FID file (C)
mftrace	Move FID trace (C)
rfblk	Reverse FID block (C)
rftrace	Reverse FID trace (C)

mfopen Memory map open FID file (C)

Syntax mfopen<(<src_expno,>dest_expno)>

Description Explicitly opens experiment source and destination FID files for using memory-mapped input and output. Opening a file explicitly can significantly speed up the data reformatting process.

mfopen searches for the FID file to be opened in the directory $vnmruser/expN/acqfil, where N is the requested experiment number or the current experiment number. Without arguments, mfopen assumes the source and destination files are the same and are in the current experiment.

After a file is open, the data reformatting commands mfblk, mfdata, mftrace, rfblk, rfdata, and rftrace can be used for moving around data. The mfclose must be used to close the file when data reformatting has been completed.

Arguments src_expno specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

dest_expno specifies the experiment number of the destination FID file.The default is the FID file of the current experiment.

If only one argument is provided, mfopen uses that as the experiment number of the destination FID file and assumes the source is the FID file of the current experiment.

Examples

```
mfopen
mfopen(3)
mfopen(1,2)
```

See also *User Programming*

Related		
	mfblk	Move FID block (C)
	mfclose	Memory map close FID file (C)
	mfdata	Move FID data (C)
	mftrace	Move FID trace (C)
	rfblk	Reverse FID block (C)
	rfdata	Reverse FID data (C)
	rftrace	Reverse FID trace (C)

mftrace Move FID trace (C)

Syntax
```
mftrace(<src_expno,>src_blk_no,src_trace_no,  \
  dest_expno,dest_blk_no,dest_trace_no)
```

Description Copies FID traces specified by `src_trace_no` from a FID block specified by `src_blk_no` to a destination location specified by `dest_expno`, `dest_blk_no`, and `dest_trace_no`, using memory-mapped input and output. If a number of blocks need to be copied, explicitly opening and closing the files with the commands `mfopen` and `mfclose` can significantly speed up the data reformatting process.

`mftrace` searches for the source and destination FID file in the directory `$vnmruser/expN/acqfil`, where `N` is the requested experiment number or the current experiment number. If the FID file is not open, `mftrace` opens the file, copies the data, and closes the file.

`mftrace` cannot be used to append data to a FID file. Its purpose is for moving around data.

Be aware that `mftrace` can modify data returned to an experiment with the `rt` command. To avoid modification, enter the following sequence of VnmrJ commands before running `mftrace`:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments `src_expno` specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

`src_blk_no` specifies the source block of data to be copied. Block numbers start at 1 and run to the number of blocks in a file.

`src_trace_no` specifies the source trace of data within the specified block to be copied. Trace numbers run from 1 to number of traces in a file.

`dest_expno` specifies the experiment number of the destination FID file.

`dest_blk_no` specifies the destination block to send the copied data.

src_trace_no specifies the destination trace of data within the specified block to be copied. Trace numbers run from 1 to the number of traces in a file.

Examples: mftrace(1,1,2,1,nv) copies trace 1 from block 1 of the current experiment to trace nv of block 1 of experiment 2.

See also: *User Programming*

Related:

mfblk	Move FID block (C)
mfclose	Memory map close FID file (C)
mfdata	Move FID data (C)
mfopen	Memory map open FID file (C)
rftrace	Reverse FID trace (C)
rfblk	Reverse FID block (C)
rfdata	Reverse FID data (C)

mht — Move Hadamard parameters from one workspace to another

Syntax:
```
mht( <from_exp,> to_exp )
from_exp is the workspace number to move parameters from.
If not specified, the current workspace is used.
to_exp is the workspace number to move Hadamard
parameters into.
```

Applicability: VnmrJ 3.1

Description: The mht macro moves Hadamard parameters from one workspace to another. It transfers the following parameters: htfrq1, htbw1, sw or sw1, tof or dof.

mht is used in the "Move HT pars to exp" entry box in the editht dialog. It may also be used from the command line.

Arguments: htfrq1 - Hadamard frequency list in indirect dimension, in Hz from center of spectrum, or ppm.

htbw1 - Hadamard band width in indirect dimension, in Hz. It may be a single value or a list of values for each element in the htfrq1 list.

tn - nucleus used for frequency list.

sw - spectral width in direct dimension

sw1 - spectral width in 1st indirect dimension

tof - frequency offset in direct dimension

dof - frequency offset in 1st indirect dimension

Examples:

See also: ht
editht
HsqcHT
tocsyHT

minsw Reduce spectral width to minimum required (M)

Description Searches the spectrum for peaks, sets new limits accordingly, and then calls movesw to calculate a new transmitter offset tof and spectral width sw.

See also *NMR Spectroscopy User Guide*

Related

movesw	Move spectral window according to cursors (M)
movetof	Move transmitter offset (M)
sw	Spectral width in directly detected dimension (P)
tof	Frequency offset for transmitter offset (P)

mkCPprotocol Make Protocol

Syntax `mkCPprotocol(<name,path,tabname,time,seqfil,type,setup_macro, required_experiments,menu1,menu2,dialog option)>`

Applicability VnmrJ 3.1

Description This utility is used to create a protocol.

Examples `mkCPprotocol('cobalt',userdir+'/templates/vnmrj/protocols','Std1d',23,'s2pul','LIB','cobalt','','exotic','nucleus','')`

Arguments

```
arg1 - protocol name if $# < 1 then arg1=pslabel
arg2 - directory where the protocol will be written
       if $# < 2 then
arg2=userdir+'/templates/vnmrj/protocols
arg3 - tabname - name of the ExperimentPanel tab
       if $# < 3 then arg2=apptype from parlib
arg4 - default time (real number)
       if $# < 4 then arg2=ACQtime from parlib
arg5 - seqfil, if $# < 5 then seqfil from parlib entry
or =arg1
arg6 - type, if $# < 6 then $arg6='LIB'
arg7 - macro, if $# < 7 then $arg7=arg1
arg8 - required exp, if $# < 8 then $arg8 is not used
arg9 - menu1, if $# < 9 then $arg9 is not used
arg10 - menu2, if $# < 10 then $arg10 is not used
arg11 - dodialog option
```

See also

mkdir Create new directory (C)

Syntax `mkdir(directory)`

Description Creates a new UNIX directory. The function of the VnmrJ `mkdir` command is similar to the UNIX `mkdir` command.

Arguments `directory` is the name of the new directory to be created.

Examples `mkdir('tests')`
`mkdir('/home/george')`

See also *NMR Spectroscopy User Guide*

Related `rmdir` Remove directory (C)

mlabel Menu label (P)

Description Stores the label for a menu button. Usually this parameter is arrayed, with one label for each button in the menu. This parameter is stored in a user's global file and is set whenever a menu is called.

See also *User Programming*

Related `menu` Change status of menu system (C)
`mstring` Menu string (P)

move Move to an absolute location to start a line (C)

Syntax `move(<'graphics'|'plotter'>,x,y)`

Description Moves the start of a line to an absolute location with the coordinates given as an argument. `move` is part of a line drawing capability that includes the `pen` and `draw` commands. `pen` selects the pen number of the plotter (`'pen1'`, `'pen2'`, etc.) or the color (`'red'`, `'green'`, `'blue'`, etc.). `move` sets the point from which to start drawing the line. `draw` draws a line from that point to the point given by the `draw` arguments. Refer to the description of the `draw` command for examples of using the line drawing capability.

Arguments `'graphics'` and `'plotter'` are keywords selecting output to the graphics window or a plotter device. The default is `'plotter'`. The output selected is passed to subsequent `pen`, `move`, or `draw` commands, remaining unchanged until different output is specified.

`x,y` are the absolute coordinates, in mm, of a point to move to. The range of `x` is 0 at the left edge of the chart and `wcmax` at the right edge of the chart. The range of `y` is `-20` at the bottom of the chart and `wc2max` at the top.

See also *NMR Spectroscopy User Guide*

Related `draw` Draw line from current location to another location (C)
`gin` Return current mouse position and button values (C)

pen	Select a pen or color for drawing (C)
wcmax	Maximum width of chart (P)
wc2max	Maximum width of chart in second direction (P)

movedssw Set downsampling parameters for selected spectral region (M)

Description Sets the parameters dslsfrq and downsamp to appropriate values for digital filtering and downsampling in a cursor-selected spectral region. To accomplish this, Fourier transform an oversampled data set, and then run the ds program. In the resulting spectral display, enclose the desired region with the cursors, and then run movedssw.

See also *NMR Spectroscopy User Guide*

Related

downsamp	Downsampling factor applied after digital filtering (P)
ds	Display a spectrum (C)
dslsfrq	Bandpass filter offset for downsampling (P)

moveossw Set oversampling parameters for selected spectral region (M)

Description Sets the parameters oslsfrq and sw to appropriate values for oversampling and digital filtering in a cursor-selected spectral region. To accomplish this, acquire a data set without digital filtering, and then run the ds program. In the resulting spectral display, enclose the desired region with the cursors, and then run moveossw. The value of oversamp is manually set.

See also *NMR Spectroscopy User Guide*

Related

ds	Display a spectrum (C)
oslsfrq	Bandpass filter offset for oversampling (P)
oversamp	Oversampling factor for acquisition (P)
sw	Spectral width in directly detected dimension (P)

movesw Move spectral window according to cursors (M)

Syntax movesw<(width)>

Description Uses the parameters cr and delta to calculate a new transmitter offset tof and a new spectral width sw. If referencing was used, it is also adjusted. The movesw macro also sets sp and wp to display the spectral window.

Arguments width specifies the spectral width sw. The default is to use a value calculated from the parameter delta.

Examples movesw
movesw(5000)

See also *NMR Spectroscopy User Guide*

Related		
	cr	Cursor position in directly detected dimension (P)
	delta	Cursor difference in directly detected dimension (P)
	minsw	Reduce spectral width to minimum required (M)
	movetof	Move transmitter offset (M)
	sp	Start of plot (P)
	sw	Spectral width in directly detected dimension (P)
	tof	Frequency offset for observe transmitter (P)
	wp	Width of plot (P)

movetof Move transmitter offset (M)

Syntax movetof<(frequency)>

Description Moves the transmitter offset parameter tof so that the current cursor position, defined by cr, becomes the center of the spectrum. If referencing was used, movetof maintains the referencing.

Arguments frequency specifies the transmitter frequency rather than using the cursor position to define the frequency. This provides a convenient method of moving the transmitter frequency outside the current spectral window.

See also *NMR Spectroscopy User Guide*

Related		
	cr	Cursor position in directly detected dimension (P)
	minsw	Reduce spectral width to minimum required (M)
	movesw	Move spectral window according to cursors (M)
	tof	Frequency offset for observe transmitter (P)

mp Move parameters between experiments (C)

Syntax mp(<from_exp,>to_exp)

Description Moves text and the current display, processing, and acquisition parameters from one experiment to another. No FID is transferred.

Arguments from_exp specifies the number of the experiment from which the parameters are to be taken; default is the parameters are moved from the currently active experiment.

to_exp specifies to which experiment the parameters are to be moved.

Examples mp(4)
mp(2,3)

See also *NMR Spectroscopy User Guide*

Related `md` Move display parameters between experiments (C)
`mf` Move FIDs between experiments (C)

mparval Moves a Paramter Value Between Experiments

Description Moves a parameter value between experiments.

Syntax `mparval(parametername,<origin>,target)`

Examples `mparval('sw',2,3)`

Arguments If only two arguments are supplied, the value of the first argument is moved to the workspace defined by the second argument.

Related `mf, mp, md`

mqcosy Set up parameters for MQCOSY pulse sequence (M)

Syntax `mqcosy<(level)>`

Description Sets up a multiple-quantum filtered COSY experiment.

Arguments `level` is the desired quantum level of filtration.

Examples `mqcosy`
`mqcosy(3)`

See also *NMR Spectroscopy User Guide*

mref Set referencing based on a existing spectrum of the sample (M)

Syntax `mref(<source_exp,>target_exp)<:$ret>`

`mref(source_fid)<:$ret>`

Description Use a primary referenced spectrum to reference a secondary spectrum acquired in another work space (or experiment) at the same temperature, using the same lock sample, and either a different or the same nucleus without adding a secondary reference sample. The primary spectrum must be properly referenced using the IUPAC recommended Ξ values. Ξ is the normalized frequency such that the ^{1}H signal from TMS is 100.00 MHz.

Begin with a `source_exp` spectrum (typically a ^{1}H spectrum) and reference it using an internal reference (such as TMS, see the IUPAC recommendations).

Join a different experiment and acquire a `target_exp` spectrum on a different (or same) nucleus. Enter `mref(<source_exp,>target_exp)`.

Referencing of 2D data sets using mref only applies to the directly detected dimension. The indirect dimensions is referenced using reff1 and reff2 (after using mref or after manual referencing of the observe dimension). The reference frequency for the secondary spectrum, reffrq_b, is calculated as follows:

reffrq_b = (reffrq_a / Ξ_a) * Ξ_b

mref also corrects for possible changes in the lock frequency:

```
reffrq_b = (reffrq_a / lockfreq_a) * lockfreq_b
```

mref works if the lock frequency changed between the two acquisitions, if the two spectra were acquired on different instruments, or at different field strengths.

mref calculates rfl and rfp after calculating reffrq:

```
rfp = 0
rfl = sw/2 - (sfrq - reffrq) * 1e6
```

The systemglobal parameters lockfreq and h1freq must saved in the local parameters using the saveglobal mechanism when the go command is executed. The mref macro only tracks lock frequency changes if these systemglobal parameters are saved in the local parameters.

The mref macro works with earlier data if both data sets were:

- acquired at the same lock frequency (on the same instrument).
- the lockfreq (on a data station) and (on older instruments) h1freq parameters are set to the values used to acquire the data.

Referencing action from mref are reported the on line 3. Suppress the report by suppling a return argument, e.g.:

```
$ret='' mref('myfid.fid'):$ret
```

The referencing message is captured in the return argument "$ret" and the contents of this string variable can be used to label plots with the referencing information.

Limitations: the macro works with data recalled from an archive or acquired on an other system provided the data was acquired using VNMR6.1C or newer.

Setting the global (or local) flag bioref='y' enables Bio-NMR referencing (based on nuctables/nuctabrefBio) and disables standard IUPAC / organic chemistry referencing (based on nuctables/nuctabref).

See /vnmr/nuctables/nuctabref.

Arguments source_exp — experiment containing the primary referenced spectrum or the full (or relative) path and fid file name containing the primary references spectrum.

target_exp — experiment contining spectra to be referenced based upon the primary experiment referencing.

$ret — return argument for output of mref.

Alternatively, the name of a FID file (with or without extension) can be given as a single argument; in this case, the data in the CURRENT experiment are referenced based on the referencing in the specified FID file.

Examples `mref(3)` — uses the current experiment as the source and applies the reference to the specified experiment as the target.

`mref(1,2)` — experiment 1 is the source and experiment 2 is the target.

`mref('myfid')`

`mref('/data/fids/myfid.fid')`

Related

setref	Set Frequency Referencing Based on Lock Signal Shift (M)
setref1	Set Frequency Referencing for f1 Evolution Dimension (M)
setref2	Set Frequency Referencing for f2 Evolution Dimension (M)
reff1	Reference f1 Indirect Dimension from Observe Dimension (M)
reff2	Reference f2 Indirect Dimension from Observe Dimension (M)
bioref	Flag for Bio-NMR Referencing (P)

mrev8 Set up parameters for MREV8 pulse sequence (M)

Applicability Systems with a solids module.

Description Converts FLIPFLOP, BR24, or S2PUL parameter set into the MREV8 multiple-pulse line narrowing sequence.

See also *User Guide: Solid-State NMR*

Related

br24	Set up parameters for BR24 pulse sequence (M)
cylmrev	Set up parameters for cycled MREV8 pulse sequence (M)
flipflop	Set up parameters for FLIPFLOP pulse sequence (M)
s2pul	Set up parameters for standard two-pulse sequence (M)

mrfb Set the filter bandwidths for multiple receivers (P)

Applicability Systems with multiple receivers

Description An array of `fb` settings to apply to individual receivers in a multiple receiver system. The first element applies to the first receiver, the second to the second receiver, and so on. If `mrfb` exists and is active, these settings override the setting specified by the `fb` parameter; otherwise, `fb` is used as the filter bandwidth setting for all receivers. If there are fewer elements in `mrfb` than there are receivers, the remaining receivers are set to the `fb` value.

Note that some older multiple receiver systems do not have the hardware to provide individual receiver control. In that case, the filter setting for receiver 1 is used on receivers 1 and 2 and the setting for receiver 3 is used on receivers 3 and 4.

Also note that mrfb is not automatically set when sw is changed. Normally, you can leave mrfb inactive and let fb be used for all receivers.

Examples — mrfb=fb/3,fb/2 sets the filter bandwidth of the first receiver to fb/3, the second to fb/2, and of the rest to fb.

Related — fb — Filter bandwidth (P)

mrgain — Set the gain for multiple receivers (P)

Applicability — Systems with multiple receivers

Description — An array of 'gain' settings to apply to individual receivers in a multiple receiver system. If it exists and is active, these settings override the setting specified by the 'gain' parameter; otherwise, 'gain' is used as the gain setting for all receivers.
Note that not all multiple receiver systems have the hardware set up to provide individual receiver control. In that case, the gain setting for receiver 1 is used on receivers 1 and 2 and the setting for receiver 3 is used on receivers 3 and 4.

Examples — mrgain=30,40,20 sets the gains of receiver 1 to 30, receiver 2 to 40 and receivers 3 and 4 to 20.

Related — gain — Receiver gain (P)

mstat — Display memory usage statistics (C)

Syntax — mstat<(program_id)>

Description — Displays statistics on memory usage by programs that use the procedures allocateWithId and release.

Arguments — program_id is the program ID, usually the same name as the program. The default is to display all program IDs and associated memory statistics.

Examples —
```
mstat
mstat('proc2d')
```

See also — *User Programming*

mstring — Menu string (P)

Description — Stores command strings to be executed when a VnmrJ menu button is clicked. Usually the mstring parameter is arrayed, with one string for each button in the menu. The string can be any string of commands that can otherwise appear in a macro or on the command line. This parameter is stored in a user's global file and is set whenever a menu is called.

See also *User Programming*

Related	menu	Change status of menu system (C)
	mlabel	Menu label (P)

mtune Tune probe using swept-tune graphical display (M)

Description `mtune` replaces `qtune` on the Varian NMR System and/or Linux. mtune runs in the spectra screen and uses VnmrJ panels. Enter `mtune` to retrieve parameters and panels.

- all parameters changeable on-the-fly (exception: tune channel for the Varian NMR System).
- one or two markers are selectable to tune at the same time.
- vertical autoscale button.
- number of acquired points changeable for better resolution at large spectral widths (more points will update less often).
- quit button returns user to current experiment and returns mtune to the original frequencies.

See also *NMR Spectroscopy User Guide*

Related	tchan	RF channel number used for tuning (P)
	tugain	Amount of receiver gain used by qtune (P)
	tune	Assign frequencies (C)

mv Move and/or rename a file (C)

Syntax `mv(from_file,to_file)`

Description Renames and/or moves a file or directory. `mv` functions the same as the command `rename`.

Arguments `from_file` is the name of the file to be moved and/or renamed.

`to_file` is the new name of the file and/or the new location. If the `from_file` argument has an extension such as `.fid` or `.par`, be sure the `to_file` argument has the same extension.

Examples

```
mv('/home/vnmr1/vnmrsys/seqlib/d2pul',
   '/vnmr/seqlib/d2pul')
```

See also *NMR Spectroscopy User Guide*

Related	copy	Copy a file (C)
	cp	Copy a file (C)
	delete	Delete a file, parameter directory, or FID directory (C)

rename	Move and/or rename a file (C)
rm	Delete a file (C)

mvsampglobal Moves sample global parameters

Description Loads sample global parameters into the current workspace from the designated workspace.

Syntax `mvsampglobal(origin)`

Examples `mvsampglobal(3)`

Related getsampglobal, resetsampglobal, savesampglobal, mvsampglobal, showsampglobal

mxconst Maximum scaling constant (P)

Description Before the start of data acquisition, noise is sampled to determine the number of bits of noise present. This number is used to set the maximum number of scaling operations on the data that can occur (essentially relevant only if dp='n'). mxconst is used to adjust this amount of scaling.

Increasing mxconst to 1, for example, permits additional scaling operations, allowing acquisition to proceed slightly longer in single-precision mode. Decreasing mxconst to -1 allows fewer scaling operations before reaching the message “maximum transients accumulated”.

One special case exists. If mxconst is set to less than -90 and single-precision acquisition is used (dp='n'), then scaling of the data is disabled. In this mode, reports of data overflowing the 16 bits is also disabled.

mxconst does not exist in standard parameter sets. If it does not exist, its value defaults to 0. To modify mxconst, first create it by entering create('mxconst','integer') and then enter the desired value.

CAUTION: **Do not change mxconst unless you are fully aware of the consequences.**

See also *NMR Spectroscopy User Guide*

Related

create	Create new parameter in a parameter tree (C)
dp	Double precision (P)

mz Move Integral Reset Points to specified experiment

Syntax `mz(<from,> to)`

Applicability VnmrJ 3.1

Description mz takes the same arguments as mf. It only moves the integral reset points (lifrq and liamp parameters) from one experiment to another.

N

n1,n2,n3	Name storage for macros (P)
ncomp	The number of components to be used in discrete DOSY fitting (P)
newexp	Create a new VNMR experiment
newmenu	Select a menu without immediate activation (C)
newshm	Interactively create a shim method with options (M)
nextexp	Value of Next Experiment
nextlocQ	Next Available Location
nextpl	Display the next 3D plane (M)
nfni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
ni3	Number of increments in 3rd indirectly detected dimension (P)
niter	Number of iterations (P)
nimax	Maximum limit of ni (P)
nl	Position cursor at the nearest line (C)
nli	Find integral values (C)
nlivast	Produces a text file of integral regions without a sum region (M)
nlivast2	Produces a text file with normalized integral regions (M)
nlivast3	Produces a text file with normalized integral regions (M)
nll	Find line frequencies and intensities (C)
nlni	Find normalized integral values
nm	Select normalized intensity mode (C)
nm1	Returns the current transmitter corresponding to the nucleus in argument 1
nm2d	Select Automatic 2D normalization (M)
Noesy	Convert the parameter to a NOESY experiment (M)
Noesy1d	Convert the parameter set to a Noesy1d experiment (M)

noise	Measure noise level of FID (C)
noisemult	Control noise multiplier for automatic 2D processing (M)
noislm	Limit noise in spectrum (M)
notebook	Notebook name (P)
np	Number of data points (P)
npoint	Number of points for fp peak search (P)
nrecords	Determine number of lines in a file (M)
nt	Number of transients (P)
ntrig	Number of trigger signals to wait before acquisition (P)
ntype3d	Specify whether f_1 or f_2 display expected to be N-type (P)
nuctable	Display VNMR style nucleus table for a given H1 frequency (M)
nugcal	A parameter array containing calibration information from calibration of non-uniform field gradients (P)
nugcal_	A parameter array containing calibration information from calibration of non-uniform field gradients (P)
nugcalib	The nugcalib macro calculates the probe/pulse sequence specific coefficients from an experiment designed to map the non-uniformity (NUG) of the pulsed field gradients (M)
nugflag	Tells the macro dosy to use processing with correction for non-uniform field gradients (P)
numrcvrs	Number of receivers in the system (P)
numreg	Return the number of regions in a spectrum (C)
numrfch	Number of rf channels (P)

n1,n2,n3 Name storage for macros (P)

Description Stores arbitrary character strings for macros. Each experiment has these three string parameters available.

See also *User Programming*

Related

dgs	Display group of special/automation parameters (M)
r1–r7	Real value storage for macros (P)

ncomp

The number of components to be used in discrete DOSY fitting

Syntax `ncomp`

Applicability VnmrJ 3.1

Description `ncomp` determines the number of components to be used in fitting the signal decay in DOSY when the parameter `dosyproc='discrete'`.

Arguments `ncomp` should be an integer >0

See also `dosy`

newexp

Create a new VNMR experiment (M)

Syntax `newexp<:$num>`

Applicability VnmrJ 3.1

Description `newexp` creates a new VNMR experiment which is used as a temporary work space and can hold a complete 1D, 2D, or 3D data set. The `newexp` macro will copy the "`current`" and "`processed`" parameter trees to the newly created experiment's curpar and procpar files.
If the global '`newexpdir`' parameter exists and is not the null string (''), and its value is the path name of an existing directory, the new experiment will be created in that directory. The `newexp` macro will return the number of the experiment it created.

Arguments There are no arguments for `newexp`.

Examples
```
newexp
newexp:$expnum
```

newmenu

Select a menu without immediate activation (C)

Syntax (1) `newmenu(menu_name)`
(2) `newmenu:$current_menu`

Description Selects a menu but does not activate it (syntax 1). This is most useful when picking which menu will be active when an interactive command exits. `newmenu` can also return the name of the currently active menu (syntax 2).

Arguments `menu_name` is the name of the file controlling the menu selected. For example, the command string `newmenu('manipulate_1D') ds` causes the menu controlled by `manipulate_1D` to be displayed when the Return button in the `ds` menu is selected.

`$current_menu` returns the file name of the currently active menu.

Examples
```
newmenu('display_1D')
newmenu:$name1
```

See also *User Programming*

Related
- menu — Change status of menu system (C)
- menuvi — Edit a menu with the *vi* text editor (M)

newshm — Interactively create a shim method with options (M)

Syntax `newshm`

Description Interactively creates a *method* string to be used in autoshimming of the magnetic field homogeneity. The string may consist of a series of shimming operations. The command dshim('method') describes method strings. Any text editor may be used to make and modify the strings.

newshm provides for either lock shimming or FID shimming, permitting the user to choose whichever is best. Lock shimming is much faster, but FID shimming is frequently much more effective in improving the field. With FID shimming, the FID evaluation range limits are requested. The full range is 0 to 100. Sensitivity to higher order gradients is greatly increased by setting the finish limit to about 5 or 10 with the start limit at 0.

newshm begins by asking for the name of the user's new shim method. If the non-spin (transverse) controls are chosen for adjustment, the spinner is turned off; otherwise, it is turned on. If uncertain about the shim criteria, the “medium to medium” choice is suitable in most circumstances. The new method is found in curexp+'/.../shimmethods.

To shim after running newshm, type method='methodname' and then enter shim or set the wshim parameter to shim before the start of acquisition. 'methodname' is the name supplied to newshm. For more information on shimming, see the manual *NMR Spectroscopy User Guide*.

Compared to stdshm, the newshm macro is more flexible and provides for a shimming time and FID evaluation limits supplied by the user. The primary difference between the macros is that stdshm provides for determining an estimated shimming time for the selected shim controls. When no time limit is supplied, autoshim continues until the exit criteria is met or the number of cycles reaches a limit.

See also *NMR Spectroscopy User Guide*

Related
- curexp — Current experiment directory (P)
- dshim — Display a shim method string (M)
- method — Autoshim method (P)
- shim — Submit an Autoshim experiment to acquisition (C)
- stdshm — Interactively create a shim method (M)
- wshim — Conditions when shimming is performed (P)
- vi — Edit text file with vi text editor (C)

nextexp — Value of Next Experiment

Description: This macro returns the value of the next highest workspace that does not exist.

Syntax: `nextexp`

Examples: `nextexp:$next_open_exp`

nextlocQ — Next Available Location

Description: This utiltiy returns the next open location in an automation tray.

Syntax: `nextlocQ`

Related: `showtray, hidetray`

nextpl — Display the next 3D plane (M)

Syntax: `nextpl`

Description: Displays the 2D color map of the next 3D plane in the set of planes defined by the parameters `plane` and `path3d`. If `nextpl` immediately follows the command `dproj`, `nextpl` results in the display of the first 3D plane within that specified set and is therefore equivalent to the command `dplane(1)`. For example, if `dplane(40)` has just been executed, `nextpl` results in the display of 3D plane 41 of that set. The `nextpl` macro is more efficient than `dplane` or `dproj` because the 3D parameter set (`procpar3d`) is not loaded into VnmrJ—it is assumed to have already been loaded by `dplane` or `dproj`, for example.

See also: *NMR Spectroscopy User Guide*

Related:

`dplane`	Display a 3D plane (M)
`dproj`	Display a 3D plane projection (M)
`dsplanes`	Display a series of 3D planes (M)
`getplane`	Extract planes from a 3D spectral data set (M)
`path3d`	Path to currently displayed 2D planes from a 3D data set (P)
`plane`	Currently displayed 3D plane type (P)
`plplanes`	Plot a series of 3D planes (M)
`prevpl`	Display the previous 3D plane (M)

nfni Number of increments in 1st indirectly detected dimension (P)

Description Number of increments of the evolution time d2, and thus the number of FIDs that will comprise the first indirectly detected dimension of a multidimensional data set. To create parameters ni, phase, and sw1 to acquire a 2D data set in the current experiment, enter addpar('2d').

Values 8 is minimum; typical values range from 32 to 512. In microimaging, ni greater than 0 is the imaging mode and ni equal to 0 is the projection mode.

See also *NMR Spectroscopy User Guide; VnmrJ Imaging NMR*

Related

addpar	Add selected parameters to the current experiment (M)
celem	Completed FID elements (P)
d2	Incremented delay in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)

ni2 Number of increments in 2nd indirectly detected dimension (P)

Description Number of increments of the evolution time d3, and thus the number of FIDs that will comprise the second indirectly detected dimension of a multidimensional data set. To create parameters d3, ni2, phase2, and sw2 to acquire a 3D data set in the current experiment, enter addpar('3d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d3	Incremented delay in 2nd indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
par3d	Create 3D acquisition, processing, and display parameters (M)
phase2	Phase selection for 3D acquisition (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

ni3 Number of increments in 3rd indirectly detected dimension (P)

Description Number of increments of the evolution time d4, and thus the number of FIDs that will comprise the third indirectly detected dimension of a multidimensional data set. To create parameters d4, ni3, phase3.

and sw3 to acquire a 4D data set in the current experiment, enter addpar('4d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d4	Incremented delay in 3rd indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
par4d	Create 4D acquisition parameters (M)
phase3	Phase selection for 4D acquisition (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

niter Number of iterations (P)

Description Sets the maximum number of iterations in an iterative simulation.

Values 1 to 9999. The value is initialized to 20 if the Set Params button is used in setting up spin simulation parameters.

See also *NMR Spectroscopy User Guide*

nimax Maximum limit of ni (P)

Description Maximum limit of ni. Used to prevent running an unrealistic number of Hadamard-encoded experiments.

Values Any positive real integer.

See also *NMR Spectroscopy User Guide*

Related

sethtfrq1	Set a Hadamard frequency list from a line list (M)
ni	Number of increments in 1st indirectly detected dimension (P)
htfrq1	Hadamard frequency in ni (P)

nl Position cursor at the nearest line (C)

Syntax `nl<:height<,frequency>>`

Description Moves the cursor to the nearest calculated line position.

Arguments `height` is a return value set to the height of the line.

`frequency` is a return value set to the frequency of the line.

Examples
```
nl
nl:r1,r2
```

See also *NMR Spectroscopy User Guide*

nli — Find integral values (C)

Description Equivalent to the dli command except that no screen display is produced. For a list of integrals, nli stores the reset points in the parameter lifrq and stores the amplitudes in the parameter liamp.

See also *NMR Spectroscopy User Guide*

Related

cz	Clear integral reset points (C)
dli	Display list of integrals (C)
dlni	Display list of normalized integrals (M)
liamp	Amplitudes of integral reset points (P)
lifrq	Frequencies of integral reset points (P)
z	Add integral reset point at cursor position (C)

nlivast — Produces a text file of integral regions without a sum region (M)

Applicability Systems with VAST accessory.

Syntax `nlivast(last)`

Description Using predefined integral regions from the spectra for each well, nlivast writes a text file, integ.out, containing the integrals of the regions. The file is written into the current experiment. Does not add an additional region that is the sum of all the defined regions for each well (see dlivast).

Arguments last is the number of the last well. The default is 96.

See also *NMR Spectroscopy User Guide*

nlivast2 — Produces a text file with normalized integral regions (M)

Applicability Systems with VAST accessory.

Syntax `nlivast(well)`

Description Using predefined integral regions from the spectra for each well, nlivast2 writes a text file, integ.out, containing the integrals of the regions. The file is written into the current experiment. Integrals are normalized to the integral specified by the argument well.The macro nlivast2 does not add an additional region that is the sum of all the defined regions for each well (see dlivast). All of the spectra are integrated.

Arguments well is the number of the reference sample well. The default reference is well 96.

See also *NMR Spectroscopy User Guide*

nlivast3 Produces a text file with normalized integral regions (M)

Applicability Systems with VAST accessory.

Syntax `nlivast(well)`

Description Using predefined integral regions from the spectra for each well, `nlivast3` writes a text file, integ.out, containing the integrals of the regions. The file is written into the current experiment. Integrals are referenced to the integral specified by the argument `well`. The integral of spectrum from the sample specified by `well` is set to 1000. The macro `nlivast3` does not add an additional region that is the sum of all the defined regions for each well (see dlivast). All of the spectra are integrated.

Arguments `well` is the number of the reference sample well. Reference integral set to 1000. The default reference is well 96.

See also *NMR Spectroscopy User Guide*

nll Find line frequencies and intensities (C)

Syntax `nll<('pos'<,noise_mult>)><:number_lines,scale>`

Description Equivalent to the command `dll` except that the line listing is not displayed or printed. The results of this calculation are stored in `llfrq` and `llamp`. The frequencies are stored as Hz and are not referenced to `rfl` and `rfp`. Amplitudes are stored as the actual data point value; they are not scaled by `vs`.

Arguments `'pos'` is a keyword that causes only positive lines to be listed.

`noise_mult` is a numerical value that determines the number of noise peaks listed for broad, noisy peak. The default is 3. A smaller value results in more peaks, a larger value results in fewer peaks, and a value of 0.0 results in a line listing containing all peaks above the threshold `th`. Negative values of `noise_mult` are changed to 3.

`number_lines` is a return argument with the number of lines in the line list.

`scale` is a return argument with a scaling factor for line amplitudes. This scaling factor accounts for `vs` and whether the lines are listed in absolute intensity mode or normalized mode.

Examples
```
nll:n1
nll('pos'):pn
nll(2.5),sc
```

See also *User Programming*

Related

`dll`	Display listed line frequencies and intensities (C)
`llamp`	List of line amplitudes (P)
`llfrq`	List of line frequencies (P)

nlni Find normalized integral values

Applicability VnmrJ 3.1

Description `nli` is the equivalent of `dli` except that no screen display is produced.

nm Select normalized intensity mode (C)

Description Selects the normalized intensity mode in which spectra are scaled so that the largest peak in the spectrum is `vs` mm high. The alternative is the absolute intensity mode (selected by the `ai` command) in which the scale is kept constant from spectrum to spectrum to allow comparison of peak heights from one spectrum to another. The modes are mutually exclusive (i.e., the system is always in either `nm` or `ai` mode). Enter `aig`? to show which mode is currently active.

See also *NMR Spectroscopy User Guide*

Related

`ai`	Select absolute intensity mode (C)
`aig`	Absolute intensity group (P)
`vs`	Vertical scale (P)

nm1 Returns the current transmitter corresponding to the nucleus in argument 1.

Syntax

Applicability VnmrJ 3.1

Description Returns the transmitter corresponding to the nucleus in `argument 1.` `nm1` is used to set the channel number for `obs, dec, dec2` `and dec3` on the Channels screen of the Acquisition Page. If `probeConnect` and `preAmpConfig` are present they are used. Otherwise if `rfchannel` is present, it is used or if `rfchannel` is not present the defaults are set.

nm2d Select Automatic 2D normalization (M)

Syntax `nm2d<(noisemult)>`

Description Sets up parameters `th` and `vs2d` automatically for a 2D contour plot and color map display. `nm2d` measures the highest signal in the spectrum and sets `vs2d` so that the highest signal is in the range of the highest color level. It then calculates the noise threshold so that the number of points above the noise threshold is between 10% and 30% of all the points. At the same time, the difference between the mean value of all the points above the threshold (peak points) and the mean value of all the points under the threshold (noise points) is maximized. This noise threshold is then multiplied by the noise multiplier.

nm2d works both with absolute-value and phase-sensitive spectra. trace can be set to 'f1' or 'f2'.

Arguments noisemult specifies the noise multiplier number that multiplies the noise threshold:

- For ^{1}H, ^{19}F and ^{31}P (high dynamic range nuclei), and homonuclear spectra in general, the default value is 4.
- For HMQC/HSQC type spectra, the default value is also 4 but noise multipliers of 3 to 5 are often more adequate.
- For HETCOR and 2D-INADEQUATE spectra, the default value is 2.
- For "quick & dirty" COSY spectra with lots of t1 noise and other artifacts, a value of 8 and higher may be adequate for suppressing the artifacts.
- For 2D-INADEQUATE spectra, a value below 3 is appropriate to catch signals right above the noise level.
- If the multiplied noise threshold is below th=1, vs2d is scaled up; otherwise, th is increased to the desired level.
- Minimum value is 1.5 (if a lower value is entered, the value is set to 1.5).

Examples

```
nm2d
nm2d(3)
```

See also *NMR Spectroscopy User Guide*

Related

dconi	Interactive 2D contour display (C)
noisemult	Control noise multiplier for automatic 2D processing (M)
proc2d	Process 2D spectra (M)
th	Threshold (P)
trace	Mode for *n*-dimensional data display (P)
vs2d	Vertical scale for 2D displays (P)

Noesy Convert the parameter to a NOESY experiment (M)

Description Convert the parameter to a NOESY experiment.

See also *NMR Spectroscopy User Guide*

Related foldt Fold COSY-like spectrum along diagonal axis (C)

Noesy1d Convert the parameter set to a Noesy1d experiment (M)

Description Convert the parameter set to a NOESY 1D experiment.

See also *NMR Spectroscopy User Guide*

Related		
	Proton	Set up parameters for ^{1}H experiment (M).
	sel1d	Selective 1D protocols to set up (M).

noise Measure noise level of FID (C)

Syntax `noise<(excess_noise<,last_noise<,block_number>>)>`
`:r1,r2,r3,r4,r5,r6`

Description Measures the noise level of a FID. By using pw=0 so that no real signal is accumulated, one or more transients can be acquired. The value of np must be greater than 4096. noise then performs a statistical analysis of the noise, providing noise level, dc level, etc., for each channel. The noise level measurement can be repeated at various settings of gain and various settings of fb, etc., for a full system diagnosis.

Arguments excess_noise is excess noise and is used to calculate the noise figure.

last_noise is the last measured mean square noise and is used to calculate the noise figure.

block_number is the block number. The default is 1.

r1 returns the real dc offset.

r2 returns the imaginary dc offset.

r3 returns the real rms noise.

r4 returns the imaginary rms noise.

r5 returns the average rms noise.

r6 returns the percentage channel imbalance.

r7 returns the noise figure.

See also *NMR Spectroscopy User Guide*

Related		
	ddf	Display data file in current experiment (C)
	ddff	Display FID file in current experiment (C)
	ddfp	Display phase file in current experiment (C)
	fb	Filter bandwidth (P)
	gain	Receiver gain (P)
	np	Number of data points (P)
	pw	Pulse width (P)

noisemult Control noise multiplier for automatic 2D processing (M)

Syntax `noisemult<(noise_multiplier)>`

Description Predetermines the noise multiplier used by the nm2d macro when starting automatic 2D experiments. This multiplier determines the threshold level in 2D spectra.

Arguments noise_multiplier is a noise multiplier, the same as used in the nm2d macro. The default is 8 for homonuclear 2D spectra or 4 for other spectra.

Examples noisemult
noisemult(10)

See also *NMR Spectroscopy User Guide*

Related	nm2d	Automatic 2D normalization (M)
	proc2d	Process 2D spectra (M)

noislm Limit noise in spectrum (M)

Syntax noislm<(max_noise)>

Description Limits the noise present in a spectrum by reducing the vertical scale vs. If the noise is smaller than the noise limit, vs is left untouched. The noise limit is in single root-mean-square noise size; the peak-to-peak noise (width of the noise band) is about twice that value. The noise is determined by taking the smallest value from four 5% regions at the left end of the spectrum. Any filter cutoff at the end will decrease the apparent noise in the spectrum, and therefore increase the noise limit in the central part of the spectrum. Because of the particular algorithm used in this macro, signals at the left end of the spectrum should not affect the result of noislm.

Arguments max_noise is the maximum root-mean-square size, in mm, of the noise. The default is 2.

Examples noislm
noislm(5)

See also *NMR Spectroscopy User Guide*

Related	vs	Vertical scale (P)
	vsadj	Automatic vertical scale adjustment (M)
	vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
	vsadjh	Automatic vertical scale adjustment for ^{1}H spectra (M)

notebook Notebook name (P)

Description Specifies the notebook name of a sample, which is saved with a study.

Related	cqsavestudy	Macro to save study queue parameters (M)
	page	Name of page (P)
	samplename	Sample name (P)
	studypar	Study parameters (P)

np Number of data points (P)

Description Sets number of data points to be acquired. Generally, np is a *dependent* parameter and is calculated automatically when sw or at is changed. If a particular number of data points is desired, np can be entered, in which case at becomes the dependent parameter and is calculated based on sw and np.

Values np is constrained to be a multiple of 2 (Acquisition Controller or Pulse Sequence Controller board) or a multiple of 64 (Output board). (See the acquire statement in the manual *User Programming* for a description of these boards.)

See also *NMR Spectroscopy User Guide*

Related		
	at	Acquisition time (P)
	dp	Double precision (P)
	setlimit	Set limits of a parameter in a tree (C)
	sw	Spectral width in directly detected dimension (P)

npoint Number of points for fp peak search (P)

Description If npoint is defined in the current parameter set and has a value, it determines the range of data points over which the fp command searches for a maximum for each peak. To create npoint and give it a value other than the default, enter create('npoint','integer') npoint=x, where x is the new value.

Values 1 to fn/4. The default is 2.

See also *NMR Spectroscopy User Guide*

Related		
	create	Create new parameter in a parameter tree (C)
	fn	Fourier number in directly detected dimension (P)
	fp	Find peak heights (C)

nrecords Determine number of lines in a file (M)

Syntax nrecords(file):$number_lines

Description Returns the number of lines (or records) in a file.

Arguments file is the name of the file.

$number_lines returns the number of lines in the named file.

Examples nrecords(userdir+'/mark1d.out'):$num

See also *User Programming*

nt — Number of transients (P)

Description: Sets the number of transients to be acquired (i.e., the number of repetitions or scans performed to make up the experiment or FID).

Values: 1 to 1e9. For an indefinite acquisition, set `nt` to a very large number such as 1e9.

See also: *NMR Spectroscopy User Guide; VnmrJ Imaging NMR*

ntrig — Number of trigger signals to wait before acquisition (P)

Applicability: Systems with LC-NMR accessory.

Description: Sets the number of trigger signals from the LC to wait for on the external gate line before beginning acquisition. If `ntrig` is 0 or the parameter does not exist, the external gate signal is ignored. If `ntrig` noes not exist, the `parlc` macro can create it. `ntrig` is not normally entered by the user.

See also: *NMR Spectroscopy User Guide*

Related:

`parlc`	Create LC-NMR parameters (M)

ntype3d — Specify whether f_1 or f_2 display expected to be N-type (P)

Description: Indicates whether the f_1 or f_2 display is expected to be N-type, that is, opposite to the sense of precession defined by f_3, under normal 3D processing conditions.

Values: `'yn'` specifies that f_1 is expected to have an N-type display under normal 3D processing conditions.

`'ny'` specifies that f_2 is expected to have an N-type display under normal 3D processing conditions.

`'yy'` specifies that both f_1 and f_2 are expected to have N-type displays under normal 3D processing conditions. Setting `ntype3d ='yy'` changes the sense of precession in f_1 and f_2 by negating the imaginary portion of the t_1 and t_2 interferograms prior to Fourier transformation.

See also: *NMR Spectroscopy User Guide*

Related:

`fiddc3d`	3D time-domain dc correction (P)
`ft3d`	Perform a 3D Fourier transform on a 3D FID data set (M,U)
`ptspec3d`	Region-selective 3D processing (P)
`specdc3d`	3D spectral drift correction (P)
`ssfilter`	Full bandwidth of digital filter to yield a filtered FID (P)
`ssorder`	Order of polynomial to fit digitally filtered FID (P)
`rftype`	Type of rf generation

nuctable Display VNMR style nucleus table for a given H1 frequency (M)

Syntax `nuctable<(h1_freq)>`

Description The VnmrJ nucleus table is a single nucleus table, `/vnmr/nuctables/nuctable`, which is calculated based on a proton frequency of 1000.000 MHz. `nuctable` can be used to reconstruct a traditional nucleus table, e.g., based on a proton frequency of 200.057 MHz, or to calculate a nucleus table for any given proton frequency.

Arguments `h1_freq` (optional): proton frequency on which the calculated / displayed nucleus table will be based. Without argument, `nuctable` prints a nucleus table based on the proton frequency for which the current VnmrJ / VNMR installation is configured.

Examples `nuctable(200.057)`
`nuctable`:

Related `restorenuctable` Calculate and (Re-)store accurate nuctable (M)

nugcal A parameter array containing calibration information from calibration of non-uniform field gradients

Syntax `nugcal`

Applicability VnmrJ 3.1

Description nugcal is a parameter array summarising the results of a calibration of non-uniform field gradients. The first value is the gradient calibration value `gcal` used; c1-c4 are the coefficients of a fourth order power series in the exponent of the Stejskal-Tanner equation. nugcal is a global parameter specific for a a given probe and pulse sequence. The parameter `nugcal_` is a local copy that is set when a `dosy` experiment is run, to ensure that the correct parameters are available for subsequent processing if `nugflag`='y'.

See also `dosy`

`nugcalib`

`nugflag`

nugcal_ A parameter array containing calibration information from calibration of non-uniform field gradients

Syntax `nugcal_`

Applicability VnmrJ 3.1

Description `nugcal_` is a local copy of the parameter `nugcal`, set when a `dosy` experiment is run to ensure that the correct parameters are available for subsequent processing if nugflag='y'. nugcal is a parameter array summarising the results of a calibration of non-uniform field gradients. The first value is the gradientcalibration value gcal used; c1-c4 are the coefficients of a fourth order power series in the exponent of the Stejskal-Tanner equation.

See also
dosy
nugcal
nugcalib
nugflag

nugcalib

The nugcalib macro calculates the probe/pulse sequence specific coefficients from an experiment designed to map the non-uniformity (NUG) of the pulsed field gradients.

Syntax `nugcalib`

nugcalib(calibrant,(T|D),saveglobal,saveprobe)

Applicability VnmrJ 3.1

Description `nugcalib` calculates a set of four coefficients that relate the nominal gradient strength per DAC point, gcal, to the calculated diffusional signal attenuation as a function of gradient for a given probe and pulse sequence. As input, `nugcalib` requires:

- the calibrant used ('w' for pure water, 'd' for dilute HDO, 'o' for other;
- the temperature (T) in Celsius if 'w' or 'd', or the diffusion coefficient (D) in units of 10**-10 m2/s if 'o';
- decisions on whether or not to save the results in the global parameter file and/or in the current probe file.

This information is supplied either as four arguments (see below) or by dialogue. The macro:

- takes a set of signal profiles measured under a read gradient, performs monoexponential DOSY fitting on each point across the profile, and uses the resultant data and the known diffusion coefficient for the calibrant to obtain a map of relative gradient strength as a function of position;
- fits this map with gradfit (C) to obtain a set of coefficients;
- uses these coefficients to extrapolate into regions of small signal;
- normalises the signal profile with profile_int (C);
- takes the gradient coefficients and signal profile and uses deacy_gen to calculate the diffusional attenuation as a function of nominal gradient strength;
- and uses powerfit (C) to fit this decay to the exponential of a power series in the Stejskal-Tanner exponent, storing the results in the array nugcal_ (and optionally in the global parameter nugcal and/or the current probe file).

Arguments `nugcalib('w',temperature,('n'|'y'),('n'|'y'))`

```
nugcalib('d',temperature,('n'|'y'),('n'|'y'))
nugcalib('o',diffusion coefficient,('n'|'y'),('n'|'y'))
```

See also
decay_gen
dosy
gcal
gcal_
gradfit
nugcal
nugcal_
nugflag
powerfit
profile_int

nugflag — Tells the macro dosy to use processing with correction for non-uniform field gradients

Syntax `nugflag='y'`
nugflag='n'

Applicability VnmrJ 3.1

Description When `nugflag`='n', DOSY processing invoked by the dosy macro uses simple mono- or multi-exponential fitting to estimate diffusion coefficients by fitting to the Stejskal-Tanner equation. When nugflag='y', a modified Stejskal-Tanner equation is used in which the exponent is replace by a power series, the coefficients for which are stored in the array nugcal. Correction for non-uniform gradients is available in both 2D and 3D DOSY, but only for discrete fitting (dosyproc='discrete') and not for CONTIN.

See also
nugcal
nugcalib
dosy
dosyproc

numrcvrs — Number of receivers in the system (P)

Applicability Systems with multiple receivers.

Description An integer giving the number of receivers installed in the system. `numrcvrs` is set from the config panel by the vnmr1 user.

numreg — Return the number of regions in a spectrum (C)

Syntax `numreg:number_regions`

Description Returns the number of regions in a spectrum previously divided by the `region` command, by manual means using the `z` command, or by the Resets button in `ds`. A *region* is the area between two reset points in integral mode, with every other reset point designating the start of a *baseline* region and not included in the count of regions.

Arguments `number_regions` returns the number of peak regions in the spectrum.

Examples `numreg:$num`

See also *User Programming*

Related

`ds`	Display a spectrum (C)
`getreg`	Get frequency limits of a specified region (C)
`region`	Divide spectrum into regions (C)
`z`	Add integral reset point at cursor position (C)

numrfch — Number of rf channels (P)

Description Holds the number of rf channels available. The value is set with the Number of RF Channels label in the Spectrometer Configuration window. `numrfch` represents the hardware in the system. For example, if the last experiment used the second decoupler, `numrfch` is set to 2. The software then leaves the second decoupler on if it was on and leaves it off if it was off.

CAUTION Do not reset numrfch to eliminate the use of a channel. See the description of `dn2` and `dn3` for the method to disable channels.

Values The fifth channel can only be used with the deuterium decoupler channel.

See also *VnmrJ Installation and Administration*

Related

`config`	Display current configuration and possibly change it (M)
`dn2`	Nucleus for the second decoupler (P)
`dn3`	Nucleus for the third decoupler (P)
`dn4`	Nucleus for the fourth decoupler (P)

O

off — Make a parameter inactive (C)

Syntax `off(parameter<,tree>)`

Description Turns off an active parameter in any tree.

Arguments `parameter` is the name of the parameter.

`tree` is type of parameter tree: `'current'`, `'global'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the `create` command for more information on the types of trees.

Examples

```
off('gf')
off('n','global')
```

See also *User Programming*

Related

create	Create new parameter in a parameter tree (C)
on	Make a parameter active or test its state (C)
typeof	Return identifier for argument type (O)

on — Make a parameter active or test its state (C)

Syntax `on(parameter<,tree>)<:$active>`

Description Turns on an inactive parameter in any tree or tests if a parameter is active. Real variables (not strings) can be turned on and off. This can be done in any tree with the commands `on` and `off`, and by entering `name='y'` or `name='n'` to change the active flag for variables in the current tree only. The variable trees are `'current'`, `'global'`, `'processed'` and `'systemglobal'`. The default tree is `'current'`.

To test the active flag of a variable, use `on(...):$x`. This does not change the active flag of the variable, but sets `$x` to `1`, if the variable is active, or to `0`, if it is not active. If the variable does not exist, a value of `-1` is returned. Care should be taken if using the return value as a test for a conditional statement. For example, in the following fragment,

```
on('var1'):$e
if $e then
   write('line3','if statement is true with value of
%d',$e)
endif
```

the `write` command will be executed if `'var1'` is active, writing the message *if statement is true with value of 1* It will also be executed if `'var1'` does not exist, writing the message *if statement is true with value of -1*.

To only execute the `write` command if the variable is active, use something like the following:

```
on('var1'):$e
if ($e > 0.5) then
   write('line3','var1 is active')
endif
```

Arguments `parameter` is the name of the parameter to make active or to test.

`tree` is type of parameter tree: `'current'`, `'global'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the `create` command for more information on the types of trees.

`$active` is 1 if the parameter is active, or is 0 if it is not active. Adding a return argument makes `on` conduct only a test of whether the specified parameter is active and does *not* turn on the parameter if it is inactive.

Examples `on('lb'):$ison`
`on('gain','global')`

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`off`	Make a parameter inactive (C)

operator Operator name (P)

Applicability *VnmrJ Walkup*

Description Specifies the operator name. It is set when an operator logs into the Walkup interface. Multiple operators may be defined for a single user using the VnmrJ Administrator interface.

Related
acct	Writes records for operator login and logoff (M)
operatorlogin	Sets workspace and parameters for the operator (M)
vnmr_accounting	Open Accounting window (U)

operatorlogin Sets workspace and parameters for the operator (M)

Syntax `operatorlogin operator email panellevel`

Description Sets the panel display level and other parameters for an operator when the operator logs in. It also clears the new sample area in the study queue, and disables the command line if the operator has insufficient privileges. An operator may be logged in from the Switch operator dialog in the Utilities menu.

Related
acct	Writes records for operator login and logoff (M)
email	Email address (P)
operator	Operator name (P)
panellevel	Display level for VnmrJ interface pages (P)
vnmr_accounting	Open Accounting window (U)

opx Open shape definition file for Pbox (M)

Syntax `opx<(name<.ext>)>`

Description Opens the pulse shape/pattern definition input file `shapelib/Pbox.inp` for the Pbox software and writes the file header.

Arguments `name` is the name of the output shape file.
`ext` is a file name extension that specifies the file type.

Examples
```
opx
opx('newfile.DEC')
```

Related
Pbox	Pulse shaping software (U)

oscoef Digital filter coefficients for over sampling (P)

Description Specifies number of coefficients used in the digital filter. Enter addpar('oversamp') to add oscoef to the current experiment if oscoef does not exist. addpar('oversamp') creates digital filtering and oversampling parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp.

Values The default is 7.5*oversamp for inline DSP (dsp='i'). A larger number of coefficients gives a filter with sharper cutoffs; a smaller number gives a filter with more gradual cutoffs. The value of oscoef does not need to be changed when oversamp is changed because oscoef is automatically adjusted by VnmrJ to give filter cutoffs that are the same regardless of the value of oversamp.

The number of coefficients for real-time DSP (dsp='r') is determined by the hardware and is not adjustable.

Related

addpar	Add selected parameters to current experiment (M)
dsp	Type of DSP for data acquisition (P)
filtfile	File of FIR digital filter coefficients (P)
osfb	Digital filter bandwidth for oversampling (P)
oslsfrq	Bandpass filter offset for oversampling (P)
oversamp	Oversampling factor for acquisition (P)
paros	Create additional parameters used by oversampling (M)

osfb Digital filter bandwidth for oversampling (P)

Description Specifies bandwidth of the digital filter used for oversampling. If osfb does not exist in the current experiment, enter addpar('oversamp') to add it. addpar('oversamp') creates digital filtering and oversampling parameters def_osfilt, filtfile, oscoef, osfilt, oslsfrq, and oversamp.

Values Number, in Hz. A value less than sw/2 rejects frequencies at the edges of the spectrum; a value greater than sw/2 aliases noise and signals at frequencies outside of ±sw/2.

'n' sets the bandwidth to sw/2.

Related

addpar	Add selected parameters to current experiment (M)
def_osfilt	Default value of osfilt (P)
filtfile	File of FIR digital filter coefficients (P)
oscoef	Digital filter coefficients for oversampling (P)
osfilt	Oversampling filter for real-time DSP (P)
oslsfrq	Bandpass filter offset for oversampling (P)
oversamp	Oversampling factor for acquisition (P)
paros	Create additional parameters used by oversampling (M)
sw	Spectral width in directly detected dimension (P)

osfilt Oversampling filter for real-time DSP (P)

Applicability	Systems with real-time DSP.
Description	Sets the type of real-time digital filter to be used on systems equipped with the real-time DSP hardware option. `osfilt` is normally set automatically by the software based on the user's global parameter `def_osfilt`, so that `osfilt` only needs to be changed if a particular experiment is to be run with a different digital filter than the default.
Values	`'a'` or `'A'` for the Analog*Plus*™ digital filter. `'b'` or `'B'` for the brickwall digital filter. `''` (null string) causes `osfilt` to be set to the value contained in the `def_osfilt` when an acquisition is initiated (with `go`, for example).

Related
`def_osfilt`	Default value of `osfilt` (P)
`dsp`	Type of DSP for data acquisition (P)

oslsfrq Bandpass filter offset for oversampling (P)

Description	Selects a bandpass filter that is not centered about the transmitter frequency. In this way `oslsfrq` works much like lsfrq. If `oslsfrq` does not exist in the current experiment, add it with `addpar('oversamp')`, which creates digital filtering and oversampling parameters, the same as the `paros` macro.
Values	Number, in Hz. A positive value selects a region upfield from the transmitter frequency. A negative value selects a downfield region.

Related
`addpar`	Add selected parameters to current experiment (M)
`def_osfilt`	Default value of `osfilt` (P)
`filtfile`	File of FIR digital filter coefficients (P)
`fsq`	Frequency-shifted quadrature detection (P)
`lsfrq`	Frequency shift of the `fn` spectrum in Hz (P)
`oscoef`	Digital filter coefficients for oversampling (P)
`osfb`	Digital filter bandwidth for oversampling (P)
`osfilt`	Oversampling filter for real-time DSP (P)
`oversamp`	Oversampling factor for acquisition (P)
`paros`	Create additional parameters used for oversampling (M)

overrange Frequency synthesizer overrange (P)

Applicability	Systems with optional version X46 of the PTS frequency synthesizer.
Description	Configures whether an rf channel has version X46 of the PTS frequency synthesizer. The value for each channel is set using the label Frequency Overrange in the Spectrometer Configuration window.

Values `Not Present`, `10000 Hz`, or `100000 Hz`

`Not Present` indicates that this rf channel does not have the frequency overrange option.

`10000` or `100000` indicate that this rf channel has the frequency overrange option. The **10000 Hz** or **100000 Hz** choices are determined by the letters *H*, *J*, or *K* found in the PTS Synthesizers model number. The normal value for `overrange` is 10000 Hz. If **Frequency Overrange** is set to 10000 Hz or 100000 Hz, the **Latching** value for that RF channel must also be set to **Present**. When set to either 10000 Hz or 100000 Hz, `overrange` guarantees a range of phase-continuous frequency jumps of at least 10 kHz or 100 kHz in each jump direction.

See also *VnmrJ Installation and Administration*

Related
`config` Display current configuration and possibly change it (M)
`latch` Frequency synthesizer latching (P)

`oversamp` Oversampling factor for acquisition (P)

Description Specifies the oversampling factor for the acquisition. With inline digital filtering (`dsp='i'`), `np*oversamp` data points are acquired at a rate of `sw*oversamp`. The data is then transferred to the host computer, digitally filtered, and downsampled to give `np` points and a spectral width of `sw`.

With real-time digital filtering (`dsp='r'`), the oversampling, digital filtering, and down sampling all occur as each data point is collected, so that only `np` data points are ever stored in the acquisition computer memory and subsequently transferred to the host computer.

If `oversamp` does not exist in the current experiment, enter the command `addpar('oversamp')` to add it. `addpar('oversamp')` creates digital filtering and oversampling parameters `def_osfilt`, `filtfile`, `oscoef`, `osfb`, `osfilt`, `oslsfrq`, and `oversamp`.

If `oversamp` is set to a number, then that number represents the amount of oversampling to apply when collecting the data. The `oversamp` value is automatically calculated whenever `sw` is changed, provided `oversamp` is not set to `'n'`. That is the distinction between `oversamp='n'` and `oversamp=1`. In both cases, no oversampling will be used. This occurs, for example, if the `sw` parameter is greater than half the maximum spectral width. However, if `sw` is reduced so that oversampling is possible, then if `oversamp` is set to `'n'`, `oversamp` will remain set to `'n'` and oversampling will not occur. On the other hand, if `oversamp` is set to 1, then `oversamp` is recalculated and oversampling will occur. Therefore, the `oversamp` parameter accurately represents whether oversampling is performed for a data set. When `oversamp` is automatically determined based on a change to `sw`, it is set to the maximum possible oversampling factor. The value of `oversamp` can be manually reset.

Note that setting oversamp greater than 1 means oversampling is selected for the experiment. However, if the oversampling facility is not present in the system (i.e., dsp='n'), then the oversamp parameter is automatically reset to 1, indicating that no oversampling will be performed.

Two other experiment local parameters reflect whether DSP is used during the acquisition of a data set:

- fb is set to Not Active if DSP is used.
- oscoef reflects whether real-time (dsp='r') or inline (dsp='i') DSP was used. If real-time, oscoef is set to Not Active. If inline, oscoef is set to the value used by the inline algorithm.

Values Number less than or equal to 68. For inline DSP, sw*oversamp and np*oversamp are limited by the values in the following table:

Maximum sw*oversamp	*Maximum* np*oversamp
500 kHz	2M
100 kHz	128K

The maximum np*oversamp is given for double precision data (dp='y'). For dp='n', multiply this value by 2.

'n' causes normal acquisition to be done without digital filtering.

Related

addpar	Add selected parameters to current experiment (M)
def_osfilt	Default value of osfilt parameter (P)
dp	Double precision (P)
dsp	Type of DSP for data acquisition (P)
fb	Filter bandwidth (P)
filtfile	File of FIR digital filter coefficients (P)
fsq	Frequency-shifted quadrature detection (P)
np	Number of data points (P)
oscoef	Digital filter coefficients for oversampling (P)
osfb	Digital filter bandwidth for oversampling (P
osfilt	Oversampling filter for real-time DSP (P)
oslsfrq	Bandpass filter offset for oversampling (P)
paros	Create additional parameters used by oversampling (M)
sw	Spectral width in directly detected dimension (P)

owner — Operating system account owner (P)

Description Set to the Unix or Linux account owner. It is set when VnmrJ is started.

P

p1	Enter pulse width for p1 in degrees (C)
p1	First pulse width (P)
p2pul	Set up sequence for PFG testing (M)
p31	Automated phosphorus acquisition (M)
p31p	Process 1D phosphorus spectra (M)
pa	Set phase angle mode in directly detected dimension (C)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
pacosy	Plot automatic COSY analysis (C)
pad	Preacquisition delay (P)
padept	Perform adept analysis and plot resulting spectra (C)
page	Submit plot and change plotter page (C)
page	Name of page (P)
panellevel	Display level for VnmrJ interface pages (P)
pap	Plot out "all" parameters (C)
par2d	Create 2D acquisition, processing, and display parameters (M)
par3d	Create 3D acquisition, processing, and display parameters (M)
par3rf	Get display templates for 3rd rf channel parameters (M)
par4d	Create 4D acquisition parameters (M)
paramedit	Edit a parameter and its attributes with user-selected editor (C)
paramvi	Edit a parameter and its attributes with vi editor (M)
pardiff	Report differences between parameter sets (M)
pards	Create additional parameters used by down sampling (M)
parfidss	Create parameters for time-domain solvent subtraction (M)
parfix	Update parameter sets (M)
parlc	Create parameters for LC-NMR experiments (M)

parlist	List complete parameters in simple format (M)
parll2d	Create parameters for 2D peak picking (M)
parlp	Create parameters for linear prediction (M)
parmax	Parameter maximum values (P)
parmin	Parameter minimum values (P)
paros	Create additional parameters used by over sampling (M)
parside	Sets Up Parameters for Plotting Reference on Side
parstep	Parameter step size values (P)
partop	Sets Up Parameters for Plotting Reference on Top
parversion	Version of parameter set (P)
path3d	Path to currently displayed 2D planes from a 3D data set (P)
paxis	Plot horizontal LC axis (M)
Pbox	Pulse shaping software (U)
pbox_shapeinfo	Returns Pbox Shape Information
pbox_bw	Define excitation band (M)
pbox_bws	Define excitation band for solvent suppression (notch) pulses (M)
pbox_dmf	Extract dmf value from pbox.cal or Pbox shape file (M)
pbox_dres	Extract dres value from pbox.cal or Pbox shape file (M)
pbox_name	Extract name of last shape generated by Pbox from pbox.cal (M)
pbox_pw	Extract pulse length from pbox.cal or Pbox shape file (M)
pbox_pwr	Extract power level from Pbox.cal or Pbox shape file (M)
pbox_pwrf	Extract fine power level from pbox.cal or Pbox shape file (M)
pbox_rst	Reset temporary Pbox/Vnmr variables (M)
pboxget	Extract Pbox calibration data (M)
pboxpar	Add parameter definition to the Pbox.inp file (M)
pboxrst	Reset temporary Pbox variables (M)
pboxunits	Converts to Pbox default units (M)
pcmapapply	Apply Phase Correction Map to Data (C)
pcmapgen	Generate Phase Correction Map (C)
pcmapopen	Phase Correction Map Open (C)
pcon	Plot contours on a plotter (C)
pcss	Calculate and show proton chemical shifts spectrum (M)

peak	Find tallest peak in specified region (C)
peak2d	Return information about maximum in 2D data (C)
peakmin	Find the minimum point
pen	Select a pen or color for drawing (C)
pexpl	Plot exponential or polynomial curves (C)
pexpladd	Add another diffusion analysis to current plot (M)
pfgon	Pulsed field gradient amplifiers on/off control (P)
pfww	Plot FIDs in whitewash mode (C)
pge	Convert parameter set to PGE pulse sequence (M)
pge_calib	Calibrate gradient strengths for PGE pulse sequence (M)
pge_data	Extract data from single element of PGE pulse sequence (M)
pge_output	Output results from PGE pulse sequence (M)
pge_process	Automated processing of data from PGE pulse sequence (M)
pge_results	Calculate diffusion constant for integral region (M)
pge_setup	Set up gradient control parameters for PGE pulse sequence (M)
ph	Set phased mode in directly detected dimension (C)
ph1	Set phased mode in 1st indirectly detected dimension (C)
ph2	Set phased mode in 2nd indirectly detected dimension (C)
phase	Change frequency-independent phase rp (M)
phase	Phase selection (P)
phase1	Phase of first pulse (P)
phase2	Phase selection for 3D acquisition (P)
phase3	Phase selection for 4D acquisition (P)
phasing	Control update region during interactive phasing (P)
phfid	Zero-order phasing constant for the np FID (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
Phosphorus	Set up parameters for ^{31}P experiment (M)
pi3ssbsq	Set up pi/3 shifted sinebell-squared window function (M)
pi4ssbsq	Set up pi/4 shifted sinebell-squared window function (M)
pin	Pneumatics Router Interlock ((P)
pintvast	Plot VAST Intergral Data in a stacked 1D-NMR matrix format

pir	Plot integral amplitudes below spectrum (C)
pirn	Plot normalized integral amplitudes below spectrum (M)
piv	Plot integral amplitudes below spectrum (M)
pivn	Plot normalized integral amplitudes below spectrum (M)
pl	Plot spectra (C)
pl2d	Plot 2D spectra in whitewash mode (C)
plt2Darg	Plot 2D arguments (P)
plane	Currently displayed 3D plane type (P)
plapt	Plot APT-type spectra automatically (M)
plarray	Plotting macro for arrayed 1D spectra (M)
plate_glue	Define a glue order for plotting and display (U)
plc	Plot a carbon spectrum (M)
pLCNMR	Plot all forms of LC-NMR data
plcosy	Plot COSY- and NOESY-type spectra automatically (M)
pldept	Plot DEPT data, edited or unedited (M)
plexpinfo	Plots Experiment Information
plfid	Plot FIDs (C)
plfit	Plot deconvolution analysis (M)
plgrid	Plot a grid on a 2D plot (M)
plh	Plot proton spectrum (M)
plhet2dj	Plot heteronuclear J-resolved 2D spectra automatically (M)
plhom2dj	Plot homonuclear J-resolved 2D spectra automatically (M)
plhxcor	Plot X,H-correlation 2D spectrum (M)
pll	Plot a line list (M)
pllogo	Plots Logo
pll2d	Plot results of 2D peak picking (C)
Plock	Sets Protection Bit for a Parameter
plockport	Port number to use to lock out multiple ProTune processes (P)
plot	Automatically plot spectra (M)
plot1d	Plotting macro for simple (non-arrayed) 1D spectra (M)
plot2D	Plot 2D spectra (M)
plotfile	Plot to a file (M)

plothiresprep	High resolution plot output preparation (M)
plotlcnmr	An LC-NMR plotting macro (M)
plotmanual	Plot manually (M)
plotlogo	Plots a logo (M)
plotside	Plot spectrum on side (M)
plotter	Plotter device (P)
plottop	Plot spectrum on top (M)
plottopside	Plot spectrum on top and side (M)
plp	Plot phosphorus spectrum (M)
plplanes	Plot a series of 3D planes (M)
plt2Darg	Plot 2D arguments (P)
pltext	Plot text file (M)
pltmod	Plotter display mode (P)
plvast	Plot VAST Data in a stacked 1D-NMR matrix format
plvastget	Plot VAST spectral data in a vertical stacked plot mode
plvast_replot	Replot VAST spectral data one spectrum per page of paper
plvast2d	Plot VAST data in a stacked pseudo-2D format (M)
plww	Plot spectra in whitewash mode (C)
pmode	Processing mode for 2D data (P)
poly0	Display mean of the data in regression.inp file (M)
powerfit	Fits the diffusional attenuation calcuated by decay_gen to the exponential of a power series in the calibration of the non-uniformity of pulsed field gradients (C)
pp	Decoupler pulse length (P)
ppa	Plot a parameter list in plain English (M)
ppcal	Proton decoupler pulse calibration (M)
ppf	Plot peak frequencies over spectrum (C)
pph	Print pulse header (M)
ppmm	Resolution on printers and plotters (P)
pprofile	Plot pulse excitation profile (M)
pps	Plot pulse sequence (C)
prealfa	Specify a delay for longer ring down (P)
preAmpConfig	Set the band of the preamp, high or low, connected to each transmitter channel (P)

prep	Run prepare acquisition macro (M)
Presat	Set up parameters for presat ^{1}H experiment (M)
prescan	Study queue prescan (P)
presig	Preamp Signal Level Selection Parameter (parameter)
prevpl	Display the previous 3D plane (M)
prescan_CoilTable	Read or update the CoilTable File (M)
prescan_tn	Return tn string for a given atomic number (M)
printer	Printer device (P)
printfile	Path to the print-to-file image (P)
printformat	Format of saved-to-file image (P)
printlayout	Layout of printed image (P)
printoff	Stop sending text to printer and start print operation (C)
printon	Direct text output to printer (C)
printregion	Screen region to be printed (P)
printsize	Size of printed image (P)
printsend	Defines where image will print (P)
probe	Probe type (P)
probeConnect	Specify which nucleus can be acquired on each RF channel (P)
Probe_edit	Edit probe for specific nucleus (U)
probe_edit	Edit probe for specific nucleus (M)
probe_protection	Probe protection control (P)
proc	Type of processing on np FID (P)
proc1	Type of processing on ni interferogram (P)
proc1d	Processing macro for simple (non-arrayed) 1D spectra (M)
proc2	Type of processing on ni2 interferogram (P)
proc2d	Process 2D spectra (M)
procarray	Process arrayed 1D spectra (M)
process	Generic automatic processing (M)
procplot	Automatically process FIDs (M)
profile	Set up pulse sequence for gradient calibration (M)
profile_int	Normalise the experimental signal profile during calibration of non-uniform pulsed gradients (C)
proj	Project 2D data (C)

proshimhelp	Proshim help (C)
Proton	Set up parameters for ^{1}H experiment (M)
protune	Macro to start ProTune (M)
protune	Shell script to start ProTune operation (U)
protunegui	Macro to start ProTune in graphical user interface (M)
prune	Prune extra parameters from current tree (C)
pscale	Plot scale below spectrum or FID (C)
pseudo	Set default parameters for pseudo-echo weighting (M)
psg	Display pulse sequence generation errors (M)
psggen	Compile a user PSG object library (M,U)
psgset	Set up parameters for various pulse sequences (M)
psgupdateon	Enable update of acquisition parameters (C)
psgupdateoff	Prevent update of acquisition parameters (C)
pshape	Plot pulse shape or modulation pattern (M)
pshapef	Plot the last created pulse shape (M)
pshr	PostScript High Resolution plotting control (P)
pslabel	Pulse sequence label (P)
psMain	Prescan controlling macro
pslw	PostScript Line Width control (P)
pssl	Plot Arrayed Numbers (C)
ptext	Print out a text file (M)
ptspec3d	Region-selective 3D processing (P)
ptsval	PTS frequency synthesizer value (P)
pulseinfo	Shaped pulse information for calibration (M)
pulsetool	RF pulse shape analysis (U)
purge	Remove macro from memory (C)
puttxt	Put text file into a data file (C)
putwave	Write a wave into Pbox.inp file (M)
pw	Enter pulse width pw in degrees (C)
pw	Pulse width (P)
pw90	90° pulse width (P)
pwd	Display current working directory (C)

pwr	Set power mode in directly detected dimension (C)
pwr1	Set power mode in 1st indirectly detected dimension (C)
pwr2	Set power mode in 2nd indirectly detected dimension (C)
pwsadj	Adjust pulse interval time (M)
pwxcal	Decoupler pulse calibration (M)
pxbss	Bloch-Siegert shift correction during Pbox pulse generation (P)
pxrep	Flag to set the level of Pbox reports (P)
pxset	Assign Pbox calibration data to experimental parameters (M)
pxshape	Generates a single-band shape file (M)
Pxsim	Simulate Bloch profile for a shaped pulse (U)
Pxspy	Create shape definition using Fourier coefficients (U)
<pslabel>_plot	Experiment-Specific Plot Macro
<pslabel>_process	Experiment-Specific Processing Macro
<pslabel>_setup	Experiment-Specific Setup Macro

p1 Enter pulse width for p1 in degrees (C)

Syntax `p1(flip_angle<,90_pulse_width>)`

Description Calculates the flip time, in µs, given a desired flip angle and the 90° pulse. The value is entered into the pulse width parameter p1.

Arguments `flip_angle` is the desired flip angle, in degrees.

`90_pulse_width` is the 90° pulse, in µs. The default is the value of parameter pw90 if it exists.

Examples
```
p1(30)
p1(90,12.8)
```

See also *NMR Spectroscopy User Guide*

Related
ernst	Calculate the Ernst angle pulse (C)
p1	First pulse width (P)
pw90	90° pulse width (P)

p1 First pulse width (P)

Description Length of first pulse in the standard two-pulse sequence.

Values 0, 0.2 µs to 150,000 µs, in 0.1 µs steps
0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related `p1` Enter pulse width *p1* in degrees (C)

p1pat Shape of excitation pulse (P)

Applicability Systems with imaging capabilities.

Description Specifies the shape of pulse `p1` when used in imaging experiments.

Values `'hard'`, `'sinc'`, `'gauss'`, `'sech'`, `'sine'`, or any shape resident in the system pulse shape library or libraries.

See also *VnmrJ Imaging NMR*

Related `p1` First pulse width (P)
`pwpat` Shape of refocusing pulse (P)

p2pul Set up sequence for PFG testing (M)

Applicability Systems with the pulsed field gradient (PFG) module. *This sequence is not for NMR applications.*

Description Sets up the PFG two-pulse sequence, a system checkout sequence for PFG installation. Several modes are controlled by the `cmd` parameter.

- `cmd='twinkle'` sequentially addresses DACs 0 through 4. On the gradient channel interface, lights become a slow binary counter.
- `cmd='pulse'` makes a pulse of value `gzlvl1` for a time `gt1`.
- `cmd='bipulse'` makes a pulse of value `gzlvl1` for a time `gt1` followed by a pulse of value `-gzlvl1` for a time `gzlvl1`.

For other modes, see the PFG installation manual.

See also *Pulsed Field Gradient Modules Installation*

p31 Automated phosphorus acquisition (M)

Syntax `p31<(solvent)>`

Description Prepares parameters for automatically acquiring a standard ^{31}P spectrum. The parameter `wexp` is set to `'procplot'` for standard processing. If `p31` is used as the command for automation via the `enter` command, then the macro `au` is supplied automatically and should not be entered on the MACRO line of the `enter` program. However, it is possible to customize the standard `p31` macro on the MACRO line by following it with additional commands and parameters. For example, `p31 nt=1` will use the standard `p31` setup but with only one transient.

Arguments `solvent` is the name of the solvent. The default is `CDCl3`. In automation mode, the solvent is supplied by the `enter` program.

Examples `p31`
`p31('DMSO')`

See also *NMR Spectroscopy User Guide*

Related

`au`	Submit experiment to acquisition and process data (M)
`enter`	Enter sample information for automation run (C)
`p31p`	Process 1D phosphorus spectra (M)
`proc1d`	Processing macro for simple, non-arrayed 1D spectra (M)
`procplot`	Automatically process FIDs (M)
`wexp`	When experiment completes (P)

p31p Process 1D phosphorus spectra (M)

Syntax `p31p`

Description Processes non-arrayed 1D ^{31}P spectra using a set of standard macros. `p31p` is called by the `proc1d` macro but can also be used directly. Fully automatic processing (up to a point where a spectrum could be plotted) is provided: Fourier transformation (using preset weighting functions), automatic phasing (`aphx` macro), automatic integration (`integrate` macro, if required only), vertical scale adjustment (`vsadjc` macro), avoiding excessive noise (`noislm` macro), threshold adjustment (`thadj` macro), and referencing to the TMS signal, if present (`tmsref` macro).

See also *NMR Spectroscopy User Guide*

Related

`aphx`	Perform and check automatic phasing (M)
`integrate`	Automatically integrate 1D spectrum (M)
`noislm`	Avoids excessive noise (M)
`p31`	Automated phosphorus acquisition (M)
`proc1d`	Automatically process non-arrayed 1D fids (M)
`thadj`	Adjust threshold (M)
`tmsref`	Reference spectrum to TMS line (M)
`vsadjc`	Adjust vertical scale for carbon spectra (M)

pa Set phase angle mode in directly detected dimension (C)

Description Selects the phase angle mode by setting the parameter `dmg='pa'`. In the *phase angle display mode,* each real point in the displayed spectrum is calculated from the phase angle of the real and imaginary points comprising each respective complex data point. The phase angle also takes into account the phase parameters `rp` and `lp`.

For 2D data, if `pmode='partial'` or `pmode=''` (two single quotes with no space in between), `pa` has an effect on the data prior to the second Fourier transform. If pmode='full', pa acts in concert with the commands pa1, av1, pwr1, or ph1 to yield the resultant contour display for the 2D data.

See also *NMR Spectroscopy User Guide*

Related

av	Set abs. value mode in directly detected dimension (C)
dmg	Data display mode in directly detected dimension (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
lp	First-order phase in directly detected dimension (P)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
ph	Set phased mode in directly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr	Set power mode in directly detected dimension (C)
pwr1	Set power mode in 1st indirectly detected dimension (C)
rp	Zero-order phase in directly detected dimension (P)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 of 2D data (M)
wft2d	Weight and Fourier transform 2D data (M)

pa1 — Set phase angle mode in 1st indirectly detected dimension (C)

Description Selects the phase angle spectra display mode along the first indirectly detected dimension by setting the parameter dmg1 to the string value 'pa1'. If the parameter dmg1 does not exist, pa1 will create it and set it to 'pa1'.

In the phase angle mode, each real point in the displayed trace is calculated from the phase angle of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the phase angle uses the real-real and imaginary-real points from each respective hypercomplex data point. The phase angle also takes into account the phase parameters rp1 and lp1.

The pa1 command is only needed if mixed-mode display is desired. If the parameter dmg1 does not exist or is set to the null string, the display mode along the first indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter dmg). For the contour display of multidimensional data, the result of pa1 is the same as for traces provided that pmode='partial' or pmode=''.

See also *NMR Spectroscopy User Guides*

Related

av1	Set abs. value mode in 1st indirectly detected dimension (C)
dmg1	Data display mode in 1st indirectly detected dimension (P)
lp1	First-order phase in 1st indirectly detected dimension (P)
pa	Set phase angle mode in directly detected dimension (C)

ph1	Set phased mode in 1st indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr1	Set power mode in 1st indirectly detected dimension (C)
rp1	Zero-order phase in 1st indirectly detected dimension (P)

pacosy — Plot automatic COSY analysis (C)

Description: Automatically analyzes and plots a COSY data set with fn=fn1 and sw=sw1. Symmetrization of the data with the command foldt is recommended, but not required. First, select a proper threshold and perform a 2D line listing with the command ll2d. Next, plot the 2D data with the contour plot command pcon; leaving enough room at the left side of the plot for the connectivity table. Then, pacosy will analyze the data and plot the connectivities on the plotter. pacosy gets its input from the file ll2d.out in the current experiment directory. The command acosy performs the same analysis and displays the connectivities on the screen.

See also: *NMR Spectroscopy User Guide*

Related:

acosy	Automatic analysis of COSY data (C)
fn	Fourier number in directly detected dimension (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
foldt	Fold COSY-like spectrum along diagonal axis (C)
hcosy	Automated proton and COSY acquisition (M)
ll2d	Automatic and interactive 2D peak picking (C)
pcon	Plot contours on plotter (C)
relayh	Set up parameters for COSY pulse sequence (M)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 1st indirectly detected dimension (P)

pad — Preacquisition delay (P)

Description: Each NMR experiment starts with a single delay time equal to pad over and above the delay d1 that occurs before each transient. Normally, pad is set to a small, nominal time (0.5 seconds) to allow any hardware changes that may be required at the start of the acquisition to “settle in.” During experiments in which the temperature is changed, the acquisition starts pad seconds after the temperature regulation system comes to regulation. Since the sample temperature does not actually come to equilibrium for some time after that, it is generally desirable to increase pad to perhaps 300 seconds. This is especially true when running experiments involving arrays of temperatures. The pad parameter is most useful for running kinetics experiments. For example, pad=0,3600,3600,3600,3600 will run an experiment immediately when go is typed (pad=0), then wait an hour (3600 seconds), run the second experiment, etc.

Values	0,0.1 µs to 8190 sec in 12.5 ns steps 0,0.2 µs to 150,000 sec in 0.1 µs steps.	
See also	*NMR Spectroscopy User Guide; VnmrJ Walkup*	
Related	`d1`	First delay (P)
	`go`	Submit experiment to acquisition (C)

padept — Perform adept analysis and plot resulting spectra (C)

Syntax `padept<(<'noll'><,'coef'><,'theory'>)>`

Description Performs the `adept` analysis and plots the resulting spectra with a scale and the assigned line listing. Leave enough space at the left end of the display for the line list.

Arguments The following arguments can be supplied in any order:

`'noll'` is a keyword that specifies no line listing.

`'coef'` is a keyword that causes the combination coefficients to be printed.

`'theory'` is a keyword that causes the theoretical coefficients rather than optimized coefficients to be used.

Examples `padept('noll','coef')`

See also *NMR Spectroscopy User Guide*

Related

`adept`	Automatic DEPT analysis and spectrum editing (C)
`autodept`	Automated complete analysis of DEPT data (M)
`cdept`	Automated carbon and DEPT acquisition (C)
`Dept`	Set up parameters for DEPT experiment
`deptproc`	Process DEPT data (M)
`hcdept`	Automated proton, carbon, and DEPT acquisition (C)
`pldept`	Plot DEPT data, edited or unedited (M)

page — Submit plot and change plotter page (C)

Syntax `page<(number_pages<,'clear'|file>)>`

Description Submits the current plotter file, which has been created by all previous plotter commands, and changes the paper after the plot has been completed. Actual plotting is controlled by the `vnmrplot` script in the `bin` subdirectory of the system directory. The `page` command can also clear the current plotter file or save the data to a specified file name.

Arguments `number_pages` is the number of pages to move the plotter forward. The default is 1. If `number_pages` is 0, `page` submits the plot but does not change the paper.

`'clear'` is a keyword to clear the plot made thus far; that is, clear the data in the current plotter file.

`file` is the name of a file to save the plot for import into a document. If the file already exists, it is overwritten.

Examples
```
page
page(0)
page('clear')
page('myplotfile')
```

See also *NMR Spectroscopy User Guide*

Related `vnmrplot` Plot files (U)

page Name of page (P)

Description Specifies the page of a sample. It is saved with a study.

Related

`cqsavestudy`	Macro to save study queue parameters (M)
`notebook`	Notebook name (P)
`samplename`	Sample name (P)
`studypar`	Study parameters (P)

panellevel Display level for VnmrJ interface pages (P)

Description Determines which VnmrJ interface pages are available under the tabs in the parameter page area. The higher the number, the more pages are available. The only time panellevel is changed is during the login process of an operator in the Walkup interface. For the Walkup interface, the value is set by the VnmrJ Administrator (default is 10).

Values **0-9** — shows the minimum number of pages.

No shim, lock, or processing, and minimal parameter control is available. This may be used for routine automation users.

10-29 — typical for a basic Walkup user.

Shim and lock are available only if there is a sample changer. Basic processing is available. Pages are not fully populated, allowing control of a few basic parameters.

30-100 — typical for the system owner.

All pages are available and fully populated.

See also VnmrJ Installation and Administration

Related

`operator`	Operator name (P)
`operatorlogin`	Sets workspace and parameters for the operator (M)

pap Plot out “all” parameters (C)

Syntax `pap<(<template><,x><,y><,character_size>)>`

Description Plots a parameter list containing “all” parameter names and values.

Arguments `template` is the name of a template that controls the display. The default is the string parameter `ap`, which can be modified using `paramvi('ap')`. See the manual *User Programming* for rules on building a template.

`x` is the starting position in the x direction of the plot on the paper, in mm. The default is a preset value.

`y` is the starting position in the y direction of the plot on the paper, in mm. If `y` is specified, the `x` position must be also. The default is a preset value.

`character_size` is the character size of the list and is specified as a multiplier. The default is 0.70 (not available on all plotters or printers acting as plotters).

Examples
```
pap
pap(wcmax-40)
pap(10,wc2max*.9)
pap('newpap',wcmax–50,100,1.4)
```

See also *NMR Spectroscopy User Guide, User Programming*

Related

`ap`	Print out “all” parameters (C)
`ap`	“All” parameters display control (P)
`hpa`	Plot parameters on special preprinted chart paper (C)
`paramvi`	Edit a variable and its attributes using `vi` text editor (M)
`ppa`	Plot a parameter list in “English” (M)

par2d Create 2D acquisition, processing, and display parameters (M)

Description Creates the acquisition parameters `ni`, `sw1`, and `phase`, which can be used to acquire a 2D data set. `par2d` also creates any missing processing and display parameters for the `ni` (or second) dimension, including `f1coef`, `reffrq1`, `refpos1`, and `refsource1`. The `par2d` macro is functionally the same as `addpar('2d')`.

See also *NMR Spectroscopy User Guide*

Related

`addpar`	Add selected parameters to the current experiment (M)
`f1coef`	Coefficient to construct F1 interferogram (P)
`ni`	Number of increments in 1st indirectly detected dimension (P)
`phase`	Phase selection (P)
`reffrq1`	Reference frequency of reference line in 1st indirect dimension (P)
`refpos1`	Position of reference line in 1st indirect dimension (P)

refsource1	Center frequency in 1st indirect dimension (P)
set2d	General setup for 2D experiments (M)
sw1	Spectral width in 1st indirectly detected dimension (P)

par3d — Create 3D acquisition, processing, and display parameters (M)

Description: Creates the acquisition parameters ni2, sw2, d3, and phase2 that can be used to acquire a 3D data set. par3d also creates any missing processing or display parameters for the ni2 (or third) dimension, including f2coef, fiddc3d, specdc3d, and ptspec3d. The par3d macro is functionally the same as addpar('3d').

See also: *NMR Spectroscopy User Guide*

Related:

addpar	Add selected parameters to the current experiment (M)
d3	Incremented delay in 2nd indirectly detected dimension (P)
f2coef	Coefficient to construct F2 interferogram (P)
fiddc3d	3D time-domain dc correction (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
phase2	Phase selection for 3D acquisition (P)
ptspec3d	Region-selective 3D processing (P)
specdc3d	3D spectral drift correction (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

par3rf — Get display templates for 3rd rf channel parameters (M)

Applicability: Systems with a second decoupler.

Description: Retrieves the dg2 and modified ap display templates from the parameter set s2pul3rf in the system parlib directory. These two templates support the display of second decoupler acquisition parameters and 3D acquisition and processing parameters.

See also: *User Programming*

Related:

ap	"All" parameters display control (P)
dg2	Control dg2 parameter group display (P)

par4d — Create 4D acquisition parameters (M)

Applicability: Systems with a third decoupler.

Description Creates the acquisition parameters ni3, sw3, d4, and phase3 that can be used to acquire a 4D data set. The par4d macro is functionally the same as addpar('4d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d4	Incremented delay for 3rd indirectly detected dimension (P)
ni3	Number of increments in 3rd indirectly detected dimension (P)
phase3	Phase selection for 4D acquisition (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

paramedit Edit a parameter and its attributes with user-selected editor (C)

Syntax `paramedit(parameter<,tree>)`

Description Opens a parameter file for editing with a user-selected text editor. The default editor is vi. If vi is used as the editor, paramedit is functionally the same as the paramvi command. To select another editor, set the UNIX environmental variable vnmreditor to the editor name (change .login line setenv vnmreditor old_editor to become setenv vnmreditor new_editor *(e.g.,* setenv vnmreditor emacs*)* and make sure a script with the prefix vnmr_ followed by the name of the editor is placed in the bin subdirectory of the system directory (e.g., vnmr_emacs). The script file makes adjustments for the type of graphic interface in use.

Scripts in the software release include vnmr_vi and vnmr_textedit. To create other scripts, refer to the vnmr_vi script for non-window editor interfaces and to vnmr_textedit for window-based editor interfaces. The vnmreditor variable must be set before starting VnmrJ.

Arguments parameter is the name of the parameter file to be edited.

tree is a keyword for one of the parameter trees 'current', 'global', or 'processed'. The default is 'current'.

Examples
```
paramedit('ap')
paramedit('b','global')
```

See also *NMR Spectroscopy User Guide; User Programming*

Related

paramvi	Edit a parameter and its attributes with vi editor (M)
vi	Edit text file with the vi text editor (C)

paramvi Edit a parameter and its attributes with vi editor (M)

Syntax `paramvi(parameter<,tree>)`

Description Opens a parameter file for editing using the UNIX `vi` text editor. The parameter file contains various attributes of the parameter in a format documented in the manual *User Programming*. Be sure you understand the format before modifying the parameter because if an error in the format is made, the parameter will not load. When the editor is exited, the modified parameter is reloaded into the system.

Arguments `parameter` is the name of the parameter file to be edited.

`tree` is a keyword for one of the parameter trees `'current'`, `'global'`, or `'processed'`. The default is `'current'`.

Examples `paramvi('ap')`
`paramvi('b','global')`

See also *NMR Spectroscopy User Guide, User Programming*

Related

create	Create new parameter in a parameter tree (C)
destroy	Destroy a parameter (C)
destroygroup	Destroy parameters of a group in a tree (C)
display	Display parameters and their attributes (C)
fread	Read parameters from file and load them into a tree (C)
fsave	Save parameters from a tree to a file (C)
groupcopy	Copy parameters of group from one tree to another (C)
paramedit	Edit a parameter and its attributes with user-selected editor (C)
prune	Prune extra parameters from current tree (C)
setgroup	Set group of a parameter in a tree (C)
setlimit	Set limits of a parameter in a tree (C)
setprotect	Set protection mode of a parameter (C)
vi	Edit text file with the `vi` text editor (C)

pardiff Report differences between parameter sets (M)

Syntax `pardiff(set1<,set2<,parameter_group>>)`

Applicability VnmrJ 3.1

Description Reports differences between VNMR parameter sets, based on the output of the `listparam` command. Calls the UNIX diffparam shell script.

Arguments set1 and set2 are VNMR directories or parameter sets, like experiments, parameter (*.par) or FID (*.fid) files, or actuall parameter text files, like curexp+'/procpar', or userdir+'/exp4/curpar'. Experiments can also be specified by giving just their number. Unless '`procpar`' is specified, for experiments the subfile '`curpar`' will be taken, for FID or parameter file the subfile '`procpar`' is selected for the comparision. If only one file is specified, this is compared with the current experiment. The '.fid' or '.par' extension can be omitted if an FID or parameter file (directory) is specified.

parametergroup is an optional argument that permits specifying the parameter type. By default, only acquisition parameters are compared. The following options exist (only the first two characters are relevant):

- acquisition - compare acquisition parameters (default)
- processing - compare processing parameters only
- display - compare display parameters only
- spsim - compare spin simulation parameters only
- sample - compare sample parameters only
- all - compare ALL parameters (output indicates group for for each parameter)
- JCAMP - compare acquisition and processing parameters in JCAMP-DX compatible format. Inactive parameters are suppressed.

Examples pardiff(3)

pardiff(1,3,'processing')

pardiff('abc.fid')

pardiff(2,'abc.fid')

pardiff('abc.fid',3)

pardiff('xyz.par','abc.fid','all')

Related
`listparam` list parameters in simple format (UNIX)
`diffparam` report differences between parameter sets (UNIX)

pards — Create additional parameters used by downsampling (M)

Description Creates the parameters `downsamp`, `dscoef`, `dsfb`, `dslsfrq`, and `filtfile` necessary for digital filtering and downsampling. The `pards` macro is functionally the same as `addpar('downsamp')`.

See also *NMR Spectroscopy User Guide*

Related
`addpar` Add selected parameters to current experiment (M)
`downsamp` Downsampling factor applied after digital filtering (P)
`dscoef` Digital filter coefficients for downsampling (P)
`dsfb` Digital filter bandwidth for downsampling (P)
`dslsfrq` Bandpass filter offset for downsampling (P)
`filtfile` File of FIR digital filter coefficients (P)
`movedssw` Set downsampling parameters for selected spectral region (M)

parfidss — Create parameters for time-domain solvent subtraction (M)

Description Creates solvent subtraction parameters ssfilter, sslsfrq, ssntaps, and ssorder. Entering addpar('ss') is functionally equivalent to parfidss.

In a 1D transform, subtraction of the zero-frequency component from the time-domain data, usually in the context of solvent subtraction, is selected by setting ssorder and ssfilter to desired values and entering wft:

- The zfs (zero-frequency suppression) option is selected if both ssfilter and ssorder are set to a value other than “Not Used.”
- The lfs (low-frequency suppression) option is selected if ssfilter is set to a value other than “Not Used” and ssorder is set to “Not Used.”
- The zfs and lfs options are both turned off if ssfilter is set to “Not Used.”

The zfs option leads to the following series of processing events: (1) the raw FID is frequency-shifted by sslsfrq Hz, (2) the raw FID is subjected to a low- pass digital filter, (3) the filtered FID is fit to a polynomial of order ssorder, (4) the polynomial function is subtracted from the raw FID, and (5) the resulting FID is frequency-shifted by –sslsfrq Hz.

The lfs option does not include a polynomial fit (step 3 of the zfs option), which leads to the following series of processing events: (1) the raw FID is frequency-shifted by sslsfrq Hz, (2) the raw FID is subjected to a low-pass digital filter, (3) the filtered FID is directly subtracted from the raw FID, (4) the resulting FID is frequency-shifted by –sslsfrq Hz.

The quality of filtering with zfs diminishes rapidly as the solvent peak moves off the exact center of the digital filter. It may be necessary to adjust lsfrq or sslsfrq to move the solvent peak to within ± 0.2 Hz of the center of the filter to obtain optimal solvent suppression. The lfs option is less sensitive to small offsets, but typically removes or distorts peaks near to the solvent peak.

In a 2D transform, solvent correction to the t_2 FIDs is invoked in the same manner with the ft1d, ft2d, wft1d, and wft2d commands and with the ft2da, ft1da, wft2da, and wft1da macros.

In a 3D transform, solvent suppression works on t_3 FIDs of 3D spectra just like in the 1D and 2D cases.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M,U)

lsfrq	Frequency shift of the fn spectrum in Hz (P)
ntype3d	N-type peak selection in f_1 or f_2 (P)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
sslsfrq	Center of solvent-suppressed region of spectrum (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
ssntaps	Number of coefficients to be used in the digital filter (P)
wft	Weight and Fourier transform 1D data (C)

parfix — Update parameter sets (M)

Description Corrects upper limits, lower limits, and step sizes of a number of parameters in the current experiment. In addition, the template parameter dgs is updated. This is automatically done via the macro fixpar if the parameter parversion is less than 4.3. parfix is used by the macro updatepars to correct saved data. This macro has been applied to all parameters as of VNMR version 4.3 and should be run on older parameter sets (e.g., rtp('pars') svp('pars') update a parameter set named pars).

See also *NMR Spectroscopy User Guide*

Related

ap	"All" parameters display control (P)
dgs	Control dgs parameter group display (P)
fixpar	Correct parameter characteristics in experiment (M)
parversion	Version of parameter set (P)
updatepars	Update all parameter sets saved in a directory (M)

parlc — Create parameters for LC-NMR experiments (M)

Applicability Systems with LC-NMR accessory.

Description Creates the following parameters used for a variety of LC-NMR experiments: curscan, dtrig, inject, ntrig, and savefile. The parlc macro also creates ni and sw1 (if they don't exist) for use in isocratic runs. Finally, it creates a display parameter dglc, so that the dg('dglc') command (or the equivalent macro dglc) can be used to display all the LC-related parameters.

Note that parlc can be used without worrying about losing existing values or attributes; if the parameters already exist, they are left untouched.

See also *NMR Spectroscopy User Guide*

Related

curscan	Scan currently in progress (P)
dglc	Control LC-NMR parameter display (P)
dtrig	Delay to wait for another trigger or acquire a spectrum (P)
inject	Trigger the injection of a sample (P)

ntrig	Number of trigger signals to wait before acquisition (P)
savefile	Base file name for saving FIDs or data sets (P)

parlist List complete parameters in simple format (M)

Syntax parlist<(parameter_group)>

Applicability VnmrJ 3.1

Description Reports differences between VNMR parameter sets, based on the output of the listparam command. Calls the UNIX diffparam shell script

Arguments parametergroup is an optional argument that permits specifying the parameter type. By default, only acquisition parameters are listed. The following options exist (only the first two characters are relevant):

- acquisition - list acquisition parameters (default)
- processing - list processing parameters only
- display - list display parameters only
- spsim - list spin simulation parameters only
- sample - list sample parameters only
- all - list ALL parameters (output indicates group for for each parameter)
- JCAMP - list acquisition and processing parameters in JCAMP-DX compatible format. Inactive parameters are suppressed.

Examples parlist

parlist('processing')

parlist('JCAMP')

Related

listparam	list parameters in simple format (UNIX)
pardiff	report differences between parameter sets (M)
diffparam	report differences between parameter sets (UNIX)

parll2d Create parameters for 2D peak picking (M)

Description Creates additional parameters th2d and xdiag for use with ll2d 2D peak picking program. parll2d is functionally the same as addpar('ll2d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
ll2d	Automatic and interactive 2D peak picking (C)
th2d	Threshold for integrating peaks in 2D spectra (P)
xdiag	Threshold for excluding diagonal peaks when peak picking (P)

parlp Create parameters for linear prediction (M)

Syntax `parlp<(dimension)>`

Description Creates parametrized options for linear prediction (LP) in the current experiment. The display template for the `dglp` macro is also created if necessary. `parlp` is functionally the same as `addpar('lp')`.

Arguments `dimension` is the dimension of a multidimensional data set. The default is to create the LP parameters `lpalg`, `lpopt`, `lpfilt`, `lpnupts`, `strtlp`, `lpext`, `strtext`, `lptrace`, and `lpprint`.

`parlp(1)` creates LP parameters `lpalg1`, `lpopt1`, `lpfilt1`, `lpnupts1`, `strtlp1`, `lpext1`, `strtext1`, `lptrace1`, and `lpprint1`. `addpar('lp',1)` is functionally equivalent to `parlp(1)`.

`parlp(2)` creates LP parameters `lpalg2`, `lpopt2`, `lpfilt2`, `lpnupts2`, `strtlp2`, `lpext2`, `strtext2`, `lptrace2`, and `lpprint2`. `addpar('lp',2)` is functionally equivalent to `parlp(2)`.

Examples
```
parlp
parlp(1)
```

See also *NMR Spectroscopy User Guide*

Related

`lpalg`	LP algorithm for `np` dimension (P)
`lpext`	LP data extension for `np` dimension (P)
`lpfilt`	LP coefficients to calculate for `np` dimension (P)
`lpnupts`	LP number of data points for `np` dimension (P)
`lpopt`	LP algorithm data extension for `np` dimension (P)
`lpprint`	LP print output for `np` dimension (P)
`lptrace`	LP output spectrum for `np` dimension (P)
`proc`	Type of processing on `np` FID (P)
`proc1`	Type of processing on `ni` interferogram (P)
`proc2`	Type of processing on `ni2` interferogram (P)
`strtext`	Starting point for LP data extension for `np` dimension (P)
`strtlp`	Starting point for LP calculation for `np` dimension (P)

parmax Parameter maximum values (P)

Description An array that holds the maximum values of other parameters. The maximum value of a parameter is an index into the array, and more than one parameter can have the same index into `parmax`. Several global parameters set in the Spectrometer Configuration window are part of `parmax`. To display all `parmax` values, enter `display('parmax','systemglobal')`.

See also *User Programming*

Related

`config`	Display current configuration and possibly change it (M
`display`	Display parameters and their attributes (C)
`paramedit`	Edit a parameter and its attributes with user-selected editor (C)

paramvi	Edit a parameter and its attributes using vi text editor (M)
parmin	Parameter minimum values (P)
parstep	Parameter step size values (P)

parmin Parameter minimum values (P)

Description An array that holds the minimum values for other parameters. The minimum value of a parameter is the index into the parmin array. More than one parameter may have the same index into the array. To display all the values in parmin, enter display('parmin','systemglobal').

See also *User Programming*

Related

paramvi	Edit a parameter and its attributes using vi text editor (M)
display	Display parameters and their attributes (C)
paramedit	Edit a parameter and its attributes with user-selected editor (C)
parmax	Parameter maximum values (P)
parstep	Parameter step size values (P)

paros Create additional parameters used by oversampling (M)

Description Creates the parameters def_osfilt, filtfile, oscoef, osfb, osfilt, oslsfrq, and oversamp for oversampling and digital filtering. paros is functionally the same as addpar('oversamp').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to current experiment (M
def_osfilt	Default value of osfilt parameter (P)
filtfile	File of FIR digital filter coefficients (P)
oscoef	Digital filter coefficients for oversampling (P)
osfb	Digital filter bandwidth for oversampling (P)
osfilt	Oversampling filter for real-time DSP (P)
oslsfrq	Bandpass filter offset for oversampling (P)
oversamp	Oversampling factor for acquisition (P)

parside Sets Up Parameters for Plotting Reference on Side

Description Sets up plotting parameters for plotting a reference spectrum on top of a 2D data set using pl('side').

Syntax parside

Related `partop`

`parstep` Parameter step size values (P)

Description An array that holds the step size values for other parameters. The step size value of a parameter is the index into the array. More than one parameter can have the same index into `parstep`. Several configuration parameters set in the Spectrometer Configuration window are part of `parstep`. To display all `parstep` values, enter `display('parstep','systemglobal')`.

See also *User Programming*

Related

`config`	Display current configuration and possibly change it (M)
`display`	Display parameters and their attributes (C)
`paramedit`	Edit a parameter and its attributes with user-selected editor (C)
`paramvi`	Edit a parameter and its attributes using `vi` text editor (M)
`parmax`	Parameter maximum values (P)
`parmin`	Parameter minimum values (P)

`partop` Sets Up Parameters for Plotting Reference on Top

Description Sets up plotting parameters for plotting a reference spectrum on top of a 2D data set using `pl`('top').

Syntax partop

Related `parside`

`parversion` Version of parameter set (P)

Description Stores the version of a parameter set. When a parameter set is updated with `updatepars` or `parfix`, `parversion` is set to 4.3 to indicate that fact. When a parameter set is retrieved into an experiment, `fixpar` checks `parversion` to determine if other parameters need to be updated using `parfix`.

See also *NMR Spectroscopy User Guide*

Related

`fixpar`	Correct parameter characteristics in experiment (M)
`parfix`	Update parameter sets (M)
`updatepars`	Update all parameter sets saved in a directory (M)

path3d Path to currently displayed 2D planes from a 3D data set (P)

Description Stores the absolute path to the current 3D data directory tree. If path3d does not exist, it is created by the macro par3d. The command select, as well as the many macros that make use of select, require path3d in order to know where the 2D planes extracted from a 3D data set can be found.

path3d is set automatically by the macros ft3d and getplane:

- ft3d sets path3d to curexp/datadir3d if ft3d is not supplied with a directory path for the transformed 3D data. If ft3d is supplied with such a directory path (e.g., /home/data/test3D), path3d is set equal to that directory path. In this case, the 3D spectral data would reside in the directory /home/data/test3D/data.
- getplane sets path3d to curexp/datadir3d if getplane is not supplied with a directory path to the transformed 3D data. If getplane is supplied with such a directory path (e.g., /home/data/test3D), path3d is set equal to that directory path. In this case, the extracted 3D planes would reside in the directory /home/data/test3D/extr.

See also *NMR Spectroscopy User Guide*

Related

dplane	Display a 3D plane (M)
dproj	Display a 3D plane projection (M)
dsplanes	Display a series of 3D planes (M)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M)
getplane	Extract planes from a 3D spectral set (M)
nextpl	Display the next 3D plane (M)
par3d	Create 3D acquisition, processing, display parameters (C)
plane	Currently displayed 3D plane type (P)
plplanes	Plot a series of 3D planes (M)
prevpl	Display the previous 3D plane (M)
select	Select a spectrum or 2D plane without displaying it (C)

paxis Plot horizontal LC axis (M)

Applicability Systems with the LC-NMR accessory.

Syntax `paxis(time,major_tic,mino_tic)`

Description Plots a horizontal LC axis. Horizontal axes are assumed to be used with “LC plots” of an entire LC run are labeled accordingly. It is assumed that relevant parameters (e.g., sc, wc, vo, vp) have not been changed after plotting the data.

Arguments `time` is the time scale, in minutes (decimal values are fine), of the axis.

`major_tic` is spacing, in minutes (decimal values are fine), of major tics.

`minor_tic` is spacing, in minutes (decimal values are fine), of minor tics.

See also *NMR Spectroscopy User Guide*

Pbox Pulse shaping software (U)

Syntax `Pbox file options`

Description Main Pbox (Pandora's Box) program for the generation of shape files for RF and gradients. (See *NMR Spectroscopy User Guide* manual for description of interactive Pbox usage).

Arguments `file` is the name of a shape file.

`options` is any of the Pbox parameters initialized by the `'-'` sign and followed by the parameter value. The following options can be in any order and combinations:

`-b time`	Activates Bloch simulator, sets `simtime`, in sec.
`-c`	Calibrate only, do not create a shape file.
`-f file`	Set name of the output file.
`-h wave`	Print wave file header.
`-i wave`	Print wave file parameters.
`-l ref_pw90`	Length, in µs, of reference pw90 pulse.
`-o`	List options.
`-p ref_pwr`	Reference power level, in dB.
`-r file`	Reshape Pbox pulse.
`-s stepsize`	Define length, in µs, of a single step in waveform.
`-t wave`	Print wave title.
`-w wavestr`	Set wave data string.
`-v`	Run in verbose mode. Also print Pbox version.
`-value`	Sets reps to value.

Examples
```
Pbox -i eburp2
Pbox newshape -wc 'eburp1 450 -1280.0' -1
Pbox sel.RF -w 'eburp1 420 -800' 'eburp1 420 1200'
Pbox -w 'eburp1 200 -1200' -attn e -p1 45 54.2 -b
Pbox tst -w 'esnob 20p 170p' -sfrq 150.02 -refofs 55p
-ref_pwr 45 -ref_pw90 54.2
```

See also *NMR Spectroscopy User Guide*

Related

`cpx`	Create Pbox shape file (M)
`dprofile`	Display pulse excitation profile from Pbox software (M)
`dshape`	Display pulse shape (M)

dshapef	Display last generated pulse shape (M)
dshapei	Display pulse shape interactively (M)
opx	Open shape definition file for Pbox (M)
pbox_bw	Define excitation band (M)
pbox_bws	Define excitation band for solvent suppression (notch) pulses (M)
pbox_dmf	Extract dmf value from Pbox shape file (M)
pbox_dres	Extract dres value from Pbox shape file (M)
pbox_name	Extract name of last shape file generated by Pbox (M)
pbox_pw	Extract pulse length from Pbox shape file (M)
pbox_pwr	Extract pulse power from Pbox shape file (M)
pbox_pwrf	Extract pulse fine power from Pbox (M)
pboxget	Extract all calibration data from a Pbox shape file (M)
pboxpar	Add parameter definition to the pbox.inp file (M)
pboxrst	Reset temporary Pbox/VnmrJ variables (M)
pboxunits	Converts to Pbox default units (M)
pph	Print pulse header (M)
pprofile	Plot pulse excitation profile from Pbox software (M)
pshape	Plot pulse shape (M)
pshapef	Display pulse shape or modulation pattern interactively (M)
putwave	Write a wave into Pbox.inp file (M)
pxset	Assign Pbox calibration data to experimental parameters (M)
pxshape	Generates a single-band shape file (M)
Pxsim	Simulate Bloch profile for a shaped pulse (M)
Pxspy	Create shape definition using Fourier coefficients (U)
selex	Defines excitation band (M)
setwave	Sets a single excitation band in Pbox.inp file (M)
shdec	Shaped observe excitation sequence (M)

pbox_bw — Define excitation band (M)

Syntax `pbox_bw<(shapename)>`

Description Defines the excitation band from the position of cursors in the graphics window and reports them to user. It also sets `r1` to excitation bandwidth and `r2` to offset. This macro is used mainly in Pbox menus and macros.

Arguments `shapename` is the name of a shape as in `wavelib`; mainly for use with menus.

See also *NMR Spectroscopy User Guide*

Related Pbox — Pulse shaping software (U)

pbox_bws — Define excitation band for solvent suppression (notch) pulses (M)

Syntax `pbox_bws<(shapename)>`

Description Defines the excitation band from the position of cursors in the graphics window and reports them to user. It also sets r1 to excitation bandwidth and r2 to offset. Note, the left cursor should be placed on the left side of the excitation band and the right cursor on resonance of the solvent signal. This macro is mainly used in Pbox menus and macros.

Arguments shapename is the name of a shape file as in wavelib, mainly for use with menus.

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pbox_dmf — Extract dmf value from pbox.cal or Pbox shape file (M)

Syntax `pbox_dmf<(shapefile.DEC)>:exp_param`

Description Extracts the dmf value from the file shapefile.DEC created by Pbox or, if file name is not provided, from the pbox.cal file containing parameters of the last created Pbox shape file.

Arguments shapefile.DEC is the name of a shape file.

exp_param is a dmf type experiment parameter.

Examples

```
pbox_dmf('myfile.DEC'):mydmf
pbox_dmf:dmf2
```

See also *NMR Spectroscopy User Guide*

Related

dmf	Decoupler modulation frequency for first decoupler (P)
Pbox	Pulse shaping software (U)

pbox_dres — Extract dres value from pbox.cal or Pbox shape file (M)

Syntax `pbox_dres<(shapefile.DEC)>:exp_param`

Description Extracts the dres value from the file shapefile.DEC created by Pbox or, if file name is not provided, from the Pbox.cal file containing parameters of the last created Pbox shape file.

Arguments shapefile.DEC is the name of a shape file.

exp_param is a dres type experiment parameter.

Examples

```
pbox_dres('myfile.DEC'):mydres
pbox_dres:dres2
```

See also *NMR Spectroscopy User Guide*

Related		
	dres	Tip-angle resolution for first decoupler (P)
	Pbox	Pulse shaping software (U)

pbox_name Extract name of last shape generated by Pbox from pbox.cal (M)

Syntax `pbox_name:exp_name`

Description Extracts name of the last shape file generated by Pbox and stored in the Pbox.cal file. Note, that the file name extension is not stored explicitly and is not provided by this macro.

Arguments `exp_name` returns the name of last shape file.

Examples
```
pbox_pw:shname
pbox_pw:pwpat
```

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pbox_pw Extract pulse length from pbox.cal or Pbox shape file (M)

Syntax `pbox_pw<(shapefile.RF)>:exp_param`

Description Extracts pulse length from the file `shapefile.RF` generated by Pbox or, if file name is not provided, from `pbox.cal` file containing parameters of the last created Pbox shape file. Returns the pulse length, in µs.

Arguments `shapefile.RF` is the shape file name, including the extension.

`exp_param` is a `pw` type experiment parameter.

Examples
```
pbox_pw('myfile.RF'):softpw
pbox_pw:selpw
```

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pbox_pwr Extract power level from Pbox.cal or Pbox shape file (M)

Syntax `pbox_pwr<(shapefile.ext)>:exp_param`

Description Extracts the power lever from the file `shapefile.ext` generated by Pbox or, if file name is not provided, from the `pbox.cal` file containing parameters of the last created Pbox shape file. Returns the power level, in dB. The `exp_param` parameter will not be changed by this macro if the parameter is previously set to `'n'` (not used).

Arguments `shapefile.ext` is the name of the shape file.

`exp_param` is a power type experiment parameter.

Examples `pbox_pwr('myfile.DEC'):mypwr`
`pbox_pwr:dpwr2`

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pbox_pwrf Extract fine power level from pbox.cal or Pbox shape file (M)

Syntax `pbox_pwrf<(shapefile.ext)>:exp_param`

Description Extracts the fine power lever from the file `shapefile.ext` generated by Pbox or, if file name is not provided, from the `pbox.cal` file containing parameters of the last created Pbox shape file. Returns the value of fine power, in dB. Note that the parameter will not be changed by this macro if it was previously set to `'n'` (not used).

Arguments `shapefile.ext` is the name of the shape file.

`exp_param` is a fine power type experiment parameter.

Examples `pbox_pwrf('myfile.DEC'):mypwrf`
`pbox_pwrf:dpwrf`

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pbox_rst Reset temporary Pbox/Vnmr variables (M)

Syntax `pbox_rst`

Applicability VnmrJ 3.1

Description `pbox_rst` resets variables r1-r4 = 0, n2='n' and n3=''. The macro adds also some standrd comment lines to Pbox.inp file. Used in menues and other `Pbox` macros.

Examples opx selex('isnob3') pbox_rst pboxpar('name','selinv.RF') cpx

Related opx
selex
cpx
setwave

pbox_shapeinfo Returns Pbox Shape Information

Description Returns values of shape, bandwidth, offset, and pulsewidth for a given Pbox shapefile.

Syntax `pbox_shapeinfo(shapefile)`

Examples pbox_shapeinfo('WURST40.DEC'):$shape,$bandwidth,$offset,$pulsewidth Pbox

pboxget Extract Pbox calibration data (M)

Syntax `pboxget<(shfile.ext)>:$name,$pw,$pwr,$pwrf,$dres,$dmf`

Description Extracts calibration data from the file `shfile.ext` generated by `Pbox` or, if a file name is not provided, from the `pbox.cal` file containing parameters of the last created Pbox shape file. Returns shape name and the values of total pulse length (in µs), power (dB), fine power, `dres`, and `dmf`. The parameter will not be changed by this macro if the parameter was previously set to `'n'` (not used).

Arguments `shfile.ext` is the name of the shape file, including the extension.

`name` is the experiment parameter receiving the shape name (without the extension).

`pw` is the experiment parameter receiving the total pulse length, in µs.

`pwr` is the experiment parameter receiving the power level, in dB.

`pwrf` is the experiment parameter receiving the fine power level.

`dres` is the experiment parameter receiving the decoupler resolution.

`dmf` is the experiment parameter receiving the decoupler modulation frequency.

Examples
```
pboxget('myfile.DEC'):dseq,r1,dpwr,dpwrf,dres,dmf
pboxget('selshape.RF'):pwpat,selpw,selpwr
pboxget:dseq2,r1,dpwr2,dpwrf2,dres2,dmf2
```

See also *NMR Spectroscopy User Guide*

Related `Pbox` Pulse shaping software (U)

pboxget Extract Pbox calibration data from pbox.cal or Pbox shapefile (M)

Syntax `pboxget<(shapefile.EXT)>:$name,$par1,$par2,$par3,$par4,$par5`

Applicability VnmrJ 3.1

Description `pboxget` extracts calibration data from shapefile.ext generated by Pbox or, if filename is not provided, from pbox.cal file containing parameters of the last generated Pbox shapefile. Order of the returned parameters is as follows : name, pw, `pwr`, pwrf, `dres`, `dmf`.

Warning : parameter is not changed by this macro if it was set to 'n' (not used)!

Arguments shapefile.EXT - shapefile name including extension.

name - name without extension

pw - length of the waveform (us)

pwr - power level (dB)

pwrf - fine power level

dres - decoupler resolution

dmf - decoupler modulation frequency

Examples pboxget('myfile.DEC'):dseq,dres,dpwr,dpwrf,dres,dmf

pboxget('selshape.RF'):pwpat,selpw,selpwr

pboxget:dseq2,dres2,dpwr2,dmf2,dres2,dmf2

Related

pbox_dmf	extract dmf value from Pbox shapefile
pbox_dres	extract dres value from Pbox shapefile
pbox_name	extract name of last shapefile generated by Pbox
pbox_pw	extract pulse length from Pbox shapefile
pbox_pwr	extract pulse power from Pbox shapefile
pbox_pwrf	extract pulse fine power from Pbox shapefile
Pbox	Pandora's box pulse/pattern generator (UNIX)

pboxpar Add parameter definition to the Pbox.inp file (M)

Syntax `pboxpar('name'<,value>)`

Applicability VnmrJ 3.1

Description `pboxpar` adds a parameter definition to Pbox.inp file.

Arguments name - parameter name

value - value of the parameter

Examples pboxpar('name','myfile.DEC')

pboxpar('bsim','y')

pboxpar('T1', 0.24)

Related opx
selex
cpx
setwave

pboxrst Reset temporary Pbox variables (M)

Description Resets `r1=0`, `r2=0`, `r3=0`, `r4=0`, `n2='n'`, `n3=''`, and adds some standard comment lines to the `Pbox.inp` file. This macro is used in menus and other Pbox macros.

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pboxunits Converts to Pbox default units (M)

Syntax pboxunits

Description Used by Pbox menus to scale parameters related to time or frequency down to Pbox default units (Hz or seconds) before the parameter is stored in the Pbox.inp file.

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

pcmapapply Apply Phase Correction Map to Data (C)

Syntax pcmapapply([<filename>,]<index>)

Applicability VnmrJ 3.1

Description "pcmapapply" applies a pixel by pixel phase shift to the current datafile using the complex phase correction values from the phase correction map $vnmruser/expN/datdir/<filename>.

It assumes the phase correction map file to be opened resides in the user's $vnmruser/expN/datdir directory where N is the current experiment number. If the "filename" argument is not given, the command defaults to the filename "pcmap". The "index" argument must always be supplied, ranges from 1 to n, and specifies the desired correction map block within the file.

The phase correction values are generated by "pcmapgen". One or more phase correction maps may be generated. In the case of a multislice EPIexperiment there may be one phase correction map for each slice. As mentioned before, the command uses data from the current datafile; which means that a fourier transform must have been performed on the data. For images, a "ft1d" should be done on the data before using this command.

"pcmapapply" will open and close the phase map file unless it has been explicitly opened with "pcmapopen". Explicitly opening a phase correction map file may be desired if there are a large number of images or data sets to be processed.

The "pcmap" commands have been developed for processing Echo Planar images, but can also be used for other applications if applicable.

Arguments 'filename' optional argument specifying the phase correction map file name residing in the user's $vnmruser/expN/datdir directory. The default file is $vnmruser/expN/datdir/pcmap.

'index' argument specifying which phase correction map to use in the file. This value will usually be 1.

Examples `ft1d`('nf',2)

`pcmapapply`(1)

`ft2d`('nf',2)

Related `pcmapopen` Phase Correction Map Open
`pcmapgen` Generate Phase Correction Map

pcmapgen Generate Phase Correction Map (C)

Syntax `pcmapgen([<filename>,]<index>)`

Applicability VnmrJ 3.1

Description "`pcmapgen`" generates pixel by pixel complex phase correction values from the current datafile and stores them into the <index> block in the phase correction map file `$vnmruser/expN/datdir/<filename>`.

It assumes the phase correction map file to store the values resides in the user's $vnmruser/expN/datdir directory where N is the current experiment number. If the "filename" argument is not given, the command defaults to the filename "`pcmap`". The "index" argument must always be supplied, ranges from 1 to n, and specifies the desired correction map block within the file.

One or more phase correction maps may be generated, although they can only be generated one at a time. As mentioned before, the command uses data from the current datafile; which means that a fourier transform must have been performed on the data. For images, a "`ft1d`" should be done on the data before using this command.

"`pcmapgen`" will create, open, and close the phase map file unless it has been explicitly opened with "`pcmapopen`". Explicitly opening a phase correction map file may be desired if there are a large number of images or data sets to be processed.

The "`pcmap`" commands have been developed for processing Echo Planar images, but can also be used for other applications if applicable.

Arguments 'filename' optional argument specifying the phase correction map file name residing in the user's $vnmruser/expN/datdir directory. The default file is $vnmruser/expN/datdir/pcmap.

'index' argument specifying which phase correction map to use in the file. This value will usually be 1.

Examples `ft1d`('nf',1)

`pcmapgen`(1)

Related `pcmapopen` Phase Correction Map Open
`pcmapapply` Apply Phase Correction Map to Data

pcmapclose Phase Correction Map Close (C)

Syntax `pcmapopen([<filename>,]<max_index>)`
pcmapclose

Applicability VnmrJ 3.1

Description "pcmapopen" explicitly opens a phase correction map file using memory mapped I/O. It assumes the phase correction map file to be opened resides in the user's $vnmruser/expN/datdir directory where N is the current experiment number. If the "filename" argument is not given, the command defaults to the filename "pcmap". The "max_index" argument must always be supplied and be greater than or equal to the maximum number of phase maps stored in the file. Once the phase correction map is opened the phase correction commands "pcmapgen" and "pcmapapply" can be used to generate maps and correct data.

Explicitly opening a phase correction map file can significantly speed up the data processing. The "pcmap" commands have been developed for processing Echo Planar images, but can also be used for other applications if applicable.

Once the file has been opened a "pcmapclose" command must be used to close the file when finished. "pcmapclose" closes phase correction map file that has been explicitly opened with a "pcmapopen" command.

Arguments 'filename' optional argument specifying the phase correction map file name residing in the user's $vnmruser/expN/datdir directory.

'max_index' argument specifying the maximum number of phase correction maps in the file. This is to ensure the memory mapping extends to or past the end of the file.

Examples pcmapopen('pcmap',2)
pcmapclose

Related

pcmapapply	Apply Phase Correction Map to Data
pcmapgen	Generate pcmap

pcon Plot contours on a plotter (C)

Syntax `pcon<(<'pos'|'neg'><,'noaxis'><,levels><,spacing>)>`

Description Plots positive and negative peaks of a contour plot display using different colors. Specifically, if maxpen is set for *n* pens, positive peaks are plotted using colors 1 through (n+1)/2, and negative peaks are plotted using colors ((n+1)/2)+1 through n (i.e., half the colors for each, plus one extra for positive if an odd number of pens is specified). Pen 1 is always used for the axes, and the lowest contour of the positive peaks is also plotted with pen1. In all cases, the pen colors are cycled if more contours are to be plotted than there are pens available.

To plot both negative and positive contours of a phase-sensitive spectrum on a monochrome device such as a LaserJet or a plotter with a single pen, different numbers of contours may be plotted for the different sign. For example, pcon('pos',10,1.4) pcon('neg',1*)* will plot ten closely spaced positive contours and one negative contour.

Arguments 'pos' is a keyword specifying that phase-sensitive spectra plot positive peaks only. The default is to plot both positive and negative peaks.

'neg' is a keyword specifying that phase-sensitive spectra plot negative peaks only. The default is to plot both positive and negative peaks.

'noaxis' is a keyword to omit outlining the plot and omit plotting the horizontal and vertical axes.

levels is maximum number of contour levels to plot. The default is 4.

spacing is relative intensity of successive contour levels. The default is 2.

Examples

```
pcon
pcon(4,1.4)
pcon('pos','noaxis')
pcon('neg',3)
```

See also *NMR Spectroscopy User Guide*

Related

dpcon	Display plotted contours (C)
maxpen	Maximum number of pens to use (P)

pcss Calculate and show proton chemical shifts spectrum (M)

Syntax pcss<(<threshold><,max_cc><,max_width>)>

Description Calculates and shows the proton chemical shifts spectrum. The dsp command is used to display the results. The list of chemical shifts is saved in the file pcss.outpar. The original spectrum can be calculated by the wft command.

Arguments threshold sets the level whether a point belongs to a peak or is noise. The default is that pcss automatically calculates the threshold.

max_cc is the maximum allowable coupling constant in the spectrum. The default is 20 Hz.

max_width is the maximum width of a spin multiplet in the spectrum. The default is 60 Hz.

Examples

```
pcss
pcss(10)
pcss(9,20,80)
```

See also *NMR Spectroscopy User Guide*

Related

do_pcss	Calculate proton chemical shifts spectrum (C)
dsp	Display pulse sequence (C)
wft	Weight and Fourier transform 1D data (C)

peak — Find tallest peak in specified region (C)

Syntax `peak<(min_freq,max_freq)><:height,freq>`

Description Returns the height and frequency of the tallest peak in the selected region, including any referencing (i.e., the same frequency that you would measure by placing a cursor on the peak). A spectrum need not actually be displayed for `peak` to work.

Arguments With no return arguments, `peak` displays on the screen information about peak height and frequency. If two cursors are displayed, `peak` without arguments finds the tallest peak between the cursors.

`min_freq` is minimum frequency limit of the region to be searched. The default value is `sp`.

`max_freq` is maximum frequency limit, in Hz, of the region to be searched. The default value is `sp` + `wp`.

`height` returns the height, in mm, of the tallest peak in the selected region.

`freq` returns the frequency, in Hz, of the tallest peak in the selected region.

Examples
```
peak:$ht,$freq
peak(0,2000):r3
peak:$ht,cr
```

See also *User Programming*

Related

`sp`	Start of plot (P)
`wp`	Width of plot (P)

peak2d — Return information about maximum in 2D data (C)

Syntax `peak2d:$maximum_intensity<,$trace,$point>`

Description Searches the area defined by `sp`, `wp`, `sp1`, and `wp1` in a 2D data set for a maximum intensity.

Arguments `$maximum_intensity` returns the maximum intensity value found.

`$trace` returns the trace number of the maximum. The parameter `trace` defines whether f_1 or f_2 traces are counted.

`$point` returns the data point number of the maximum on that trace.

See also *NMR Spectroscopy User Guide*

Related

`sp`	Start of plot (P)
`sp1`	Start of plot in 1st indirectly detected dimension (P)
`trace`	Mode for *n*-dimensional data display (P)
`wp`	Width of plot (P)
`wp1`	Width of plot in 1st indirectly detected dimension (P)

peakmin Find the minimum point

Syntax `peakmin<(highfield,lowfield)>:ht,frq,amp`

Applicability VnmrJ 3.1

Description `peak` finds the height and frequency of the maximum point in the specified region. `peakmin` finds the height and frequency of the minimum point in the specified region.

For both `peak` and `peakmin`, height is measured in mm, and frequency is measured in Hz, including any referencing (i.e. the same frequency that you would measure by placing a cursor on that point). Default parameters for highfield and lowfield are "`sp`" and "sp+`wp`", respectively. The value of the height and frequency of the point can be returned to the caller if the command is suffixed with a colon and parameter names. An unscaled amplitude may be returned as the third value. This unscaled amplitude is independent of the current value of vs and whether the spectrum is in absolute intensity or normalized mode (ai or nm).

pen Select a pen or color for drawing (C)

Syntax `pen(<'graphics'|'plotter',><'xor'|'normal',>`
`pen|color)`

Description Selects the pen number for a plotter or the color for the graphics screen. This command is part of a line drawing capability that includes the `move` and `draw` commands. `move` sets the coordinates from which the line starts. `draw` draws a line from that point to the new coordinates specified by `draw`. Refer to the description of `draw` for examples of using the line drawing capability.

Arguments `'graphics'` and `'plotter'` are keywords selecting the output device. The default is `'plotter'`. The output selected is passed to subsequent `pen`, `move`, or `draw` commands and remains active until a different output is specified.

`'xor'` and `'normal'` are keywords selecting the drawing mode for the `'graphics'` output device. In the `'xor'` mode, if a line is drawn such that one or more points of the line are in common with a previously drawn line, the common points are erased. In the `'normal'` mode, the common points remain. The mode selected is passed to subsequent `pen`, `draw`, *or* `move` commands and remains active until a different mode is specified. The default mode is `'normal'`.

`pen` is the plotter pen number: `'pen1'`, `'pen2'`, `'pen3'`, etc. `'pen1','pen2','pen3', ...`

`color` is the active color for the graphics screen: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, `'white'`, `'cursor'`, `'integral'`, `'threshold'`, `'scale'`, `'fid'`, `'spectrum'`, `'imaginary'`, `'parameter'`

This list includes eight symbolic color names (cursor, integral, etc). The actual colors associated with the symbolic names may be set with the "Display options..." tool in the Edit menu. The advantage of using the symbolic names is that they are probably adjusted to look good with the chosen background color. For example, using the color white for drawing on the graphics screen may look fine with a dark background, but will be invisible if the background is white. Using the color 'spectrum' will probably look good for both light and dark backgrounds.

Examples
`pen('pen2')`
`pen('graphics','red')`

See also *NMR Spectroscopy User Guide*

Related

draw	Draw line from current location to another location (C)
move	Move to an absolute location (C)

pexpl — Plot exponential or polynomial curves (C)

Syntax `pexpl<(<options,><line1,line2, ...)>`

Description Plots exponential curves resulting from T_1, T_2, or kinetics analysis. Also plots polynomial curves from diffusion or other types of analysis. The `analyze.out` file is the data input file used to make the plot. Refer to the expl entry for the format of this file. The parameters sc, wc, sc2, and wc2 control the size of the plot.

Arguments `options` are any of the following keywords:

- `'linear'`, `'square'`, and `'log'` provide for plotting of the data points against the square or log of the data. `'linear'` controls x-axis scale, `'square'` controls the y-axis. The default is `'linear'`.
- `'link'` causes the data points to be connected rather than a plot of the theoretical curve.
- `'nocurve'` produces a plot of data points only.
- `'oldbox'` plots an additional curve on an existing plot. Only the first data set in `analyze.out` is plotted. It causes the program to get box and scale description from `expfit.out` in the current experiment.
- `'file'` followed by a file name replaces `analyze.out` as the input.

`line1, line2,...` specify curves to be plotted. The default is to plot the first six curves (if that many exist) along with the data points.

Examples
`pexpl`
`pexpl(1,3,6)`

See also *NMR Spectroscopy User Guide, User Programming*

Related

expl	Display exponential or polynomial curves (C)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

pexpladd Add another diffusion analysis to current plot (M)

Applicability Systems with the diffusion option.

Syntax `pexpladd(integral_region)`

Description Adds results of another diffusion analysis to the currently plotted results.

Arguments `integral_region` specifies the number of the region whose results are to be added to the existing plot.

Examples `pexpladd(1)`

See also *NMR Spectroscopy User Guide*

Related

expl	Display exponential or polynomial curves (C)
pexpl	Plot exponential or polynomial curves (C)
expladd	Add another diffusion analysis to current display (M)

pfgon Pulsed field gradient amplifiers on/off control (P)

Applicability Systems with pulsed field gradient (PFG) modules.

Description A global string parameter controlling the X, Y, and Z gradients for the PFG current amplifiers. Entering `su` or `go` sets the amplifiers at the current value of `pfgon`. For `pfgon` to take effect, `gradtype` must equal `p`, `q`, `l`, `t`, or `u` for the corresponding X, Y, or Z gradient, and a `su` or a `go` must be issued.

Values A three-character string, with the first character controlling the X gradient, the second the Y gradient, and the third the Z gradient. For each gradient, setting the value to `y` turns on an amplifier and setting the value to `n` turns it off. For example, `pfgon='nny'` turns on only the PFG amplifier on the Z channel, and `pfgon='nnn'` turns off the PFG amplifiers on all channels.

See also *NMR Spectroscopy User Guide*

Related

go	Submit experiment to acquisition (M)
gradtype	Gradients for X, Y, and Z axes (P)
setup	Set up parameters for basic experiments (M)
su	Submit a setup experiment to acquisition (M)

pfww Plot FIDs in whitewash mode (C)

Syntax `pfww<(<start><,finish><,step><,'all'|'imag'>)>`

Description Plots FIDs in whitewash mode (after the first FID, each FID is blanked out in regions in which it is behind an earlier FID). The position of the first FID is governed by parameters `wc`, `sc`, and `vpf`.

Arguments `start` is the index of a particular FID for arrayed 1D or 2D data sets. For multiple FIDs, `start` is the index of the first FID.

`finish` is the index of the last FID for multiple FIDs.

`step` specifies the increment for the FID index. The default is 1.

`'all'` is a keyword to plot all of the FIDs. This is the default.

`'imag'` is a keyword to plot only the imaginary FID channel. The default is `'all'`.

Examples
```
pfww
pfww(4,10,2,'imag')
```

See also *NMR Spectroscopy User Guide*

Related

`dfs`	Display stacked FIDs (C)
`dfww`	Display FIDs in whitewash mode (C)
`plfid`	Plot FIDs (C)
`sc`	Start of chart (P)
`vpf`	Current vertical position of FID (P)
`wc`	Width of chart (P)

pge Convert parameter set to PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Description Adds all necessary parameters to perform the PGE (Pulse Gradient Experiment) pulse sequence, taking those parameters from the file `/vnmr/parlib/pge`.

See also *NMR Spectroscopy User Guide*

Related

`pge_calib`	Calibrate gradient strengths for PGE pulse sequence (M)
`pge_data`	Extract data from single element of PGE pulse sequence (M)
`pge_output`	Output results from PGE pulse sequence (M)
`pge_process`	Automated processing of data from PGE pulse sequence (M)
`pge_results`	Calculate diffusion constant for integral region (M)
`pge_setup`	Set up gradient control parameters for PGE pulse sequence (M)

pge_calib Calibrate gradient strengths for PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Description Calibrates the parameters `grad_cw_coef` and `grad_p_coef`, which relate the DAC values (in DAC units) to the gradient strengths (in gauss/cm). Given a diffusion constant measurement (made with `pge_results`) for a known diffusion constant, `pge_calib` then adjusts the calibration parameters to produce the correct diffusion constant.

See also *NMR Spectroscopy User Guide*

Related

`pge`	Calibrate gradient strengths for PGE pulse sequence (M)
`pge_results`	Calculate diffusion constant for integral region (M)

`pge_data` Extract data from single element of PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Syntax `pge_data(array_index)`

Description Extracts integral information from a currently displayed element of a PGE (Pulse Gradient Experiment) and writes the results in the current experiment directory as the file `info_#`, where `#` is the value of the `array_index` argument (e.g., if `array_index` is 5, the file is `info_5`)

Arguments `array_index` is the number of the array element from which the data is extracted.

Examples `pge_data(5)`

See also *NMR Spectroscopy User Guide*

Related `pge` Calibrate gradient strengths for PGE pulse sequence (M)

`pge_output` Output results from PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Description Prints the calculated results from the PGE (Pulse Gradient Experiment) pulse sequence on a printer and plots the graphs of calculated decay curves.

See also *NMR Spectroscopy User Guide*

Related `pge` Calibrate gradient strengths for PGE pulse sequence (M)

`pge_process` Automated processing of data from PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Syntax `pge_process`

Description Performs full automated processing of data from a PGE (Pulse Gradient Experiment) pulse sequence.

See also *NMR Spectroscopy User Guide*

Related `pge` Calibrate gradient strengths for PGE pulse sequence (M)

`pge_results` Calculate diffusion constant for integral region (M)

Applicability Systems with the diffusion option.

Syntax `pge_results(integral_region<,reference_region>)`

Description Calculates a diffusion coefficient based on a single integral region in the spectrum (if one input argument) or calculates diffusion coefficient of an integral region consisting of two components (if two input arguments).

Arguments `integral_region` is the number of the integral region on which to perform the analysis

`reference_region` is the number of the integral region used to get the value of the diffusion coefficient.

Examples `pge_results(2)`
`pge_results(1,3)`

See also *NMR Spectroscopy User Guide*

Related `pge` Calibrate gradient strengths for PGE pulse sequence (M)

`pge_setup` Set up gradient control parameters for PGE pulse sequence (M)

Applicability Systems with the diffusion option.

Syntax `pge_setup<('no')>`

Description Prompts the user for the values of the `g_max`, `g_min`, `g_steps`, `g_array`, `nt_first`, `nt_aray`, and other parameters for the PGE (Pulse Gradient Experiment) pulse sequence. These parameters are then used to calculate the `grad_p1` and `nt` arrays.

Arguments `'no'` is a keyword to turn off prompting the user and instead use the current values of the parameters to calculate the `grad_p1` and `nt` arrays.

Examples `pge_setup`
`pge_setup('no')`

See also *NMR Spectroscopy User Guide*

Related `pge` Calibrate gradient strengths for PGE pulse sequence (M)

ph — Set phased mode in directly detected dimension (C)

Description: Selects the phased mode by setting the parameter dmg='ph'. In the *phased spectra display mode,* each real point in the displayed spectrum is calculated from a linear combination of the real and imaginary points comprising each respective complex data point. The coefficients for this linear combination are derived from the phase parameters rp and lp.

For 2D data, if pmode='partial' or pmode='' (two single quotes with no space in between), ph has an effect on the data prior to the second Fourier transform. If pmode='full', ph acts in concert with the commands ph1, av1, or pwr1 to yield the resultant contour display for the 2D data.

See also: *NMR Spectroscopy User Guide*

Related:

av	Set abs. value mode in directly detected dimension (C)
av1	Set abs. value mode in 1st indirectly detected dimension (C)
dmg	Data display mode in directly detected dimension (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
lp	First-order phase in directly detected dimension (P)
pa	Set phase angle mode in directly detected dimension (C)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
ph1	Set phased mode in 1st indirectly detected dimension (C)
ph2	Set phased mode in 2nd indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr	Set power mode in directly detected dimension (C)
pwr1	Set power mode in 1st indirectly detected dimension (C)
rp	Zero-order phase in directly detected dimension (P)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 of 2D data (M)
wft2d	Weight and Fourier transform 2D data (M)

ph1 — Set phased mode in 1st indirectly detected dimension (C)

Description: Selects the phased spectra display mode along the first indirectly detected dimension by setting the parameter dmg1 to the string value 'ph1'. If the parameter dmg1 does not exist, ph1 will create it and set it to 'ph1'.

In the phased mode, each real point in the displayed trace is calculated from a linear combination of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the linear combination uses the real-real and imaginary-real points from each respective hypercomplex data point. The coefficients for this linear combination are derived from the phase parameters rp1 and lp1.

The `ph1` command is only needed if mixed-mode display is desired. If the parameter `dmg1` does not exist or is set to the null string, the display mode along the first indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter `dmg`). For the contour display of multidimensional data, the result of `ph1` is the same as for traces provided that `pmode='partial'` or `pmode=''`.

See also *NMR Spectroscopy User Guide*

Related

`av1`	Set abs. value mode in 1st indirectly detected dimension (C)
`dmg1`	Data display mode in 1st indirectly detected dimension (P)
`lp1`	First-order phase in 1st indirectly detected dimension (P)
`pa`	Set phase angle mode in directly detected dimension (C)
`pa1`	Set phase angle mode in 1st indirectly detected dimension (C)
`ph`	Set phased mode in directly detected dimension (C)
`pmode`	Processing mode for 2D data (P)
`pwr1`	Set power mode in 1st indirectly detected dimension (C)
`rp1`	Zero-order phase in 1st indirectly detected dimension (P)

ph2 Set phased mode in 2nd indirectly detected dimension (C)

Description Selects phased spectrum display mode processing along the second indirectly detected dimension by setting the parameter `dmg2='ph2'`. If `dmg2` does not exist or is set to the null string, `ph2` creates `dmg2` and sets it to `'ph2'`.

In the phased mode, each real point in the displayed trace is calculated from a linear combination of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the linear combination uses the real-real and imaginary-real points from each respective hypercomplex data point. The coefficients for this linear combination are derived from the phase parameters `rp2` and `lp2`.

The `ph2` command is only needed if mixed-mode display is desired. If the parameter `dmg2` does not exist or is set to the null string, the display mode along the second indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter `dmg`). For the contour display of multidimensional data, the result of `ph2` is the same as for traces provided that `pmode='partial'` or `pmode=''`.

See also *NMR Spectroscopy User Guide*

Related

`av2`	Set abs. value mode in 2nd indirectly detected dimension (C)
`dmg2`	Data display mode in 2nd indirectly detected dimension (P)
`ft1d`	Fourier transform along f_2 dimension (C)
`ft2d`	Fourier transform 2D data (C)
`lp2`	First-order phase in 2nd indirectly detected dimension (P)
`ph`	Set phased mode in directly detected dimension (C)

pmode	Processing mode for 2D data (P)
pwr2	Set power mode in 2nd indirectly detected dimension (C)
rp2	Zero-order phase in 2nd indirectly detected dimension (P)

phase — Change frequency-independent phase rp (M)

Syntax `phase(phase_change)`

Description Changes the phase of all peaks in the spectrum by adding a value to the current `rp` value. Any excess over 360° is removed.

Arguments `phase_change` is the value to be added to the current `rp` value (i.e., *new* `rp` = *old* `rp` + `phase_change`).

Examples phase(45)

See also *NMR Spectroscopy User Guide*

Related

rp	Zero-order phase in directly detected dimension (P)

phase — Phase selection (P)

Description Selects the phase cycling that determines the experiment type. To create the parameters `phase`, `ni`, and `sw1` for acquisition of a 2D data set in the current experiment, enter `addpar('2d')`.

Values The following values are generally used in experiments with phase cycling. For more details, see the specific pulse sequence.

`phase=0` selects an absolute-value 2D experiment.

`phase=1,2` selects the required two components of a hypercomplex (States-Haberkorn) experiment.

`phase=3` selects TPPI (Time Proportional Phase Incrementation).

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
cosyps	Set up parameters for phase-sensitive COSY (M)
Dqcosy	Set up parameters for double quantum filtered COSY (M)
Hmqc	Set up parameters for HMQC pulse sequence (M)
hmqcr	Set up parameters for HMQCR pulse sequence (M)
inadqt	Set up parameters for INADEQUATE pulse sequence (M)
mqcosy	Set up parameters for MQCOSY pulse sequence (M)
Noesy	Set up parameters for NOESY pulse sequence (M)
Roesy	Set up parameters for ROESY pulse sequence (M)
Tocsy	Set up parameters for TOCSY pulse sequence (M)

phase1 Phase of first pulse (P)

Applicability	Systems with a solids NMR module.
Description	Controls the first pulse phase in the cycle, in multipulse experiments.
See also	*NMR Spectroscopy User Guide*

Related

br24	Set up BR24 multiple pulse experiment (M)
flipflop	Set up sequences for multipulse (M)

phase2 Phase selection for 3D acquisition (P)

Description	Selects phase cycling type for 3D data acquisitions. Also selects the phase of the second pulse in the sequence set up by `flipflop`. To create the parameters `phase2`, `d3`, `ni2`, and `sw2` for acquisition of a 3D data set in the current experiment, enter `addpar('3d')`.
See also	*NMR Spectroscopy User Guide; User Guide: Solid-State NMR*

Related

addpar	Add selected parameters to the current experiment (M)
d3	Incremented delay for 2nd indirectly detected dimension (P)
flipflop	Set up sequences for multipulse (M)
ni2	Number of increments in 2nd indirectly detected dimension (P)
par3d	Create 3D acquisition, processing, display parameters (C)
sw2	Spectral width in 2nd indirectly detected dimension (P)

phase3 Phase selection for 4D acquisition (P)

Description	Selects phase cycling type for 4D data acquisitions. To create the parameters `phase3`, `d4`, `ni3`, and `sw3` for acquisition of a 4D data set in the current experiment, enter `addpar('4d')`.
See also	*NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d4	Incremented delay for 3rd indirectly detected dimension (P)
ni3	Number of increments in 3rd indirectly detected dimension (P)
par4d	Create 4D acquisition parameters (C)
sw3	Spectral width in 3rd indirectly detected dimension (P)

phasing Control update region during interactive phasing (P)

Description Controls the percentage of the spectrum updated during interactive phasing using the ds command.

Values 10 to 100, in percent, where 100 causes the entire spectrum to be updated, and 20 causes the area between the two vertical cursors to be updated.

See also *NMR Spectroscopy User Guide*

Related

ds	Display a spectrum (C)

phfid Zero-order phasing constant for the np FID (P)

Description Specifies the angle of zero-order rotation. This zero-order rotation is executed as a part of retrieving the time-domain data into the active region of the memory and can be used instead of the parameter rp applied to the frequency-domain data. phfid is used only in a complex phase rotation.

phfid (and related parameters lsfid and lsfrq) operate on complex np FID data, referred to as the t_2 dimension in a 2D experiment or as the t_3 dimension in a 3D experiment. phfid is in the processing group and is properly handled through the wti display.

Values –360.0 to +360.0, in degrees; 'n'

See also *NMR Spectroscopy User Guide*

Related

dfid	Display a single FID (C)
ds	Display a spectrum FID (C)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
lsfid	Number of complex points to left-shift the np FID (P)
lsfrq	Frequency shift of the fn spectrum in Hz (P)
np	Number of data points (P)
phfid1	Zero-order phasing constant for ni interferogram (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
rp	Zero-order phase in directly detected dimension (P)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 of 2D data (M)
wft2d	Weight and Fourier transform 2D data (M)
wti	Interactive weighting (C)

phfid1 Zero-order phasing constant for ni interferogram (P)

Description Specifies the angle of zero-order rotation. This zero-order rotation is executed as a part of retrieving the time-domain data into the active region of the memory and can be used instead of the parameter rp1 applied to the frequency-domain data. phfid1 is used in a complex phase rotation for complex t_1/t_2 interferograms and in a hypercomplex phase rotation for hypercomplex t_1/t_2 interferograms.

phfid1 (and related parameters lsfid1 and lsfrq1) operate on ni interferogram data, both hypercomplex and complex. ni interferogram data are referred to as the t_1 dimension in both a 2D and a 3D experiment. phfid1 is in the processing group and is properly handled through the wti display; that is, a wti operation on an ni interferogram applies the parameters phfid1, lsfid1, and lsfrq1, if selected, to the time-domain data prior to the Fourier transformation.

Values –360.0 to +360.0, in degrees; 'n'.

See also *NMR Spectroscopy User Guide*

Related

lsfid1	Number of complex points to left-shift the ni interferogram (P)
lsfrq1	Frequency shift of the fn1 spectrum in Hz (P)
ni	Number of increments in 1st indirectly detected dimension (P)
phfid	Zero-order phasing constant for np FID (P)
phfid2	Zero-order phasing constant for ni2 interferogram (P)
rp1	Zero-order phase in 1st indirectly detected dimension (P)
wti	Interactive weighting (C)

phfid2 Zero-order phasing constant for ni2 interferogram (P)

Description Specifies the angle of zero-order rotation. This zero-order rotation is executed as a part of retrieving the time-domain data into the active region of the memory and can be used instead of the parameter rp2 applied to the frequency-domain data. phfid2 is used in a complex phase rotation for complex t_1/t_2 interferograms and in a hypercomplex phase rotation for hypercomplex t_1/t_2 interferograms.

phfid2 (and related parameters lsfid2 and lsfrq2) operate on ni2 interferogram data, both hypercomplex and complex. ni2 interferogram data are referred to as the t_2 dimension in a 3D experiment. phfid2 is in the processing group and is properly handled through the wti display.

Values –360.0 to +360.0, in degrees; 'n'.

See also *NMR Spectroscopy User Guide*

Related
- `lsfid2` Number of complex points to left-shift `ni2` interferogram (P)
- `lsfrq2` Frequency shift of the `fn2` spectrum in Hz (P)
- `ni2` Number of increments in 2nd indirectly detected dimension (P)
- `phfid` Zero-order phasing constant for `np` FID (P)
- `phfid1` Zero-order phasing constant for `ni` interferogram (P)
- `rp2` Zero-order phase in 2nd indirectly detected dimension (P)
- `wti` Interactive weighting (C)

Phosphorus — Set up parameters for ^{31}P experiment (M)

Description Set up parameters for ^{31}P experiment.

pi3ssbsq — Set up pi/3 shifted sinebell-squared window function (M)

Syntax `pi3ssbsq<(<t1_inc><,t2_inc>)>`

Description Sets up a pi/3 unshifted sinebell-squared window function in 1, 2, or 3 dimensions. The macro checks whether the data is 1D, 2D, and 3D.

Arguments `t1_inc` is the number of t1 increments. The default is `ni`.

`t2_inc` is the number of t2 increments. The default is `ni2`.

See also *NMR Spectroscopy User Guide*

Related
- `gaussian` Set up unshifted Gaussian window function (M)
- `ni` Number of increments in 1st indirectly detected dimension (P)
- `ni2` Number of increments in 2nd indirectly detected dimension (P)
- `pi4ssbsq` Set up pi/4 shifted sinebell-squared window function (M)
- `sqcosine` Set up unshifted cosine-squared window function (M)
- `sqsinebell` Set up unshifted sinebell-squared window function (M)

pi4ssbsq — Set up pi/4 shifted sinebell-squared window function (M)

Syntax `pi4ssbsq<(<t1_inc><,t2_inc>)>`

Description Sets up a pi/4 unshifted sinebell-squared window function in 1, 2, or 3 dimensions. The macro checks whether the data is 1D, 2D, and 3D.

Arguments `t1_inc` is the number of t1 increments. The default is `ni`.

`t2_inc` is the number of t2 increments. The default is `ni2`.

See also *NMR Spectroscopy User Guide*

Related	gaussian	Set up unshifted Gaussian window function (M)
	ni	Number of increments in 1st indirectly detected dimension (P)
	ni2	Number of increments in 2nd indirectly detected dimension (P)
	pi3ssbsq	Set up pi/3 shifted sinebell-squared window function (M)
	sqcosine	Set up unshifted cosine-squared window function (M)
	sqsinebell	Set up unshifted sinebell-squared window function (M)

pin Pneumatics Router Interlock ((P)

Description This parameter controls the effect of a Pneumatics Router Fault. The Pneumatic Router can fault in four ways:

- Intake pressure < 20 psi
- Solids narrow bore stack temperature fault
- VT air flow exceeded.
- Power supply fault

When either of these fault occur, and interrupt alerts the console of the problem and this parameter determines how the fault is handled. Once a fault is registered, all subsequent acquisitions will see the error according to 'pin'. The error must be cleared and re-armed with sethw('pneufault','clear')

Values `'n'` -- the fault is ignored
`'w'` -- a warning msg is printed, acquisition continues
`'y'` -- an error msg is printed, acquisition is aborted

Related	tin	Temperature interlock (P)
	vtairflow	VT air flow (P)
	vtairlimits	VT air flow limits (P)

pintvast Plot VAST Intergral Data in a stacked 1D-NMR matrix format

Applicability VnmrJ 3.1

Description If an array of 1D spectra have been acquired (in particular if a block of 96 spectra has been acquired using VAST automation, especially in a microtiter-plate format), and if these spectra have been glued into a reconstructed 2D dataset (see vastglue), this macro will arrange and plot the integrals (on the plotter) in a convenient 8 x 12 sample format (as a matrix of 1D spectral intergrals).

Arguments The default is to plot all the intergrals (from 1 through arraydim). An optional argument (plvast(##)) allows one to specify that only intergrals from 1 through ## should be plotted.

See also `dsvast`
`dsvast2d`
`plvast`
`plvast2d`
`pintvast`

`pir` Plot integral amplitudes below spectrum (C)

Description Plots integral amplitudes below the appropriate spectral regions.

See also *NMR Spectroscopy User Guide*

Related

`dpf`	Display peak frequencies over spectrum (C)
`dpir`	Display integral amplitudes below spectrum (C)
`dpirn`	Display normalized integral amplitudes below spectrum (M)
`pirn`	Plot normalized integral amplitudes below spectrum (M)
`ppf`	Plot peak frequencies over spectrum (M)

`pirn` Plot normalized integral amplitudes below spectrum (M)

Description Equivalent to the command `pir` except that the sum of the integrals is normalized to the value of the parameter `ins`.

See also *NMR Spectroscopy User Guide*

Related

`dpirn`	Display normalized integral amplitudes below spectrum (M)
`ins`	Integral normalization scale (P)
`pir`	Plot integral amplitudes below spectrum (C)

`piv` Plot integral values below spectrum (M)

Syntax `piv<(vertical_position)>`

Description Labels integrals with a bracket below the spectrum and a vertical number indicating the integral value. See `dpiv` for description and use.

Related

`dpir`	Display integral amplitudes below spectrum (C)
`dpiv`	Display integral amplitudes below spectrum (M)
`dpirn`	Display normalized integral amplitudes below spectrum (C)
`dpivn`	Display normalized integral amplitudes below spectrum (M)
`pirn`	Plot normalized integral amplitudes below spectrum (C)

pir	Plot integral amplitudes below spectrum (C)
pivn	Plot normalized integral amplitudes below spectrum (M)

pivn Plot normalized integral values below spectrum (M)

Syntax `pivn<(vertical_position)>`

Description Labels integrals with a bracket below the spectrum and a vertical number indicating the integral value. See dpiv for description and use.

Related

dpir	Display integral amplitudes below spectrum (C)
dpiv	Display integral amplitudes below spectrum (M)
dpirn	Display normalized integral amplitudes below spectrum (C)
dpivn	Display normalized integral amplitudes below spectrum (M)
pirn	Plot normalized integral amplitudes below spectrum (C)
pir	Plot integral amplitudes below spectrum (C)
piv	Plot integral amplitudes below spectrum (M)

pl Plot spectra (C)

Syntax `pl<(<start,finish<,step>><,'int'><,'all'><,options>)>`

Description Plots one or more spectra. When a single spectrum is plotted, integral plotting is controlled by the parameter intmod as follows: intmod='off' turns off the integral plot, intmod='full' plots the entire integral, and intmod='partial' plots every other integral region.

For arrayed 1D spectra or for 2D spectra, a particular trace can be plotted by supplying the index number as an argument. For 2D data sets, spectra can be plotted from either the f_1 or f_2 domain by setting the parameter trace to 'f1' or 'f2', respectively. After the command ft1d, interferogram can be plotted by setting trace='f1' and then typing pl. Multiple spectra can be plotted by supplying the indexes of the first and last spectra.

The position of the first spectrum is governed by the parameters wc, sc, and vp. For 1D data, subsequent spectra are positioned relative to the preceding spectrum by the vertical and horizontal offset parameters vo and ho. For 2D data, ho defines the total horizontal offset between the first and last spectrum. Also for 2D data, vo is inactive while the parameter wc2 defines the total vertical offset between the first and last spectrum.

The parameter cutoff, if it exists and is active, defines the distance above and below the current vertical position vp at which peaks are truncated. By arraying cutoff to have two different values, truncation limits above and below the current vertical position can be controlled.

For example, cutoff=50 truncates peaks at vp+50 mm and vp–50 mm. cutoff=50,10 truncates peaks at vp+50 mm and vp–10 mm.

Arguments `start` is the index of a particular trace for arrayed 1D or 2D spectra. For multiple spectra, `start` is the index of the first spectrum.

`finish` is the index of the last spectrum for multiple spectra.

`step` specifies the increment for the spectral index. The default is 1.

`'int'` is a keyword that specifies displaying only the integral, independently of the value of intmod.

`'all'` is a keyword to plot all of the spectra. This value is the default.

`options` can be any of the following keywords:

- `'top'` or `'side'` cause the spectrum to be plotted either above or at the left edge of a contour plot. This assumes that the parameters sc, wc, sc2, and wc2 are those used to position the contour plot.
- `'dodc'` causes all spectra to be drift corrected independently.
- `'pen1'`, `'pen2'`, `'pen3'`, etc. specify a pen number on a plotter.

Examples
```
pl
pl(1,6,2)
```

See also *NMR Spectroscopy User Guide*

Related

cutoff	Data truncation limit (P)
dssa	Display stacked spectra automatically (C)
dsww	Display spectra in whitewash mode (C)
ft1d	Fourier transform along f_2 dimension (C)
ho	Horizontal offset (P)
intmod	Integral display mode (P)
plww	Plot spectra in whitewash mode (C)
pshr	PostScript High Resolution plotting control (P)
pslw	PostScript Line Width control (P)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
shownumx	x position counting from bottom left of every spectrum (P)
shownumy	y position counting from bottom left of every spectrum (P)
trace	Mode for 2D data display (P)
vo	Vertical offset (P)
vp	Vertical position of spectrum (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)

pl2d Plot 2D spectra in whitewash mode (C)

Syntax `pl2d<('nobase'|'fill'|'fillnb')>`

Description Plots a stacked plot of 2D spectra in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind an earlier spectra). Color does not represent intensity (unlike dcon), since intensity can be seen visually, but instead successive traces are displayed in different colors so that color represents frequency. The horizontal offset parameter ho is not active for this command.

Arguments 'nobase' is a keyword to activate th to suppress intensity below th.

'fill' is a keyword to fill in the peaks. Note that if 'fill' (or 'fillnb') is used, th operates linearly and not logarithmically (with factors of 2) as it does in contour or color intensity displays.

'fillnb' is a keyword to combine base suppression and peak filling.

Examples pl2d
pl2d('nobase')

See also *NMR Spectroscopy User Guide*

Related

dcon	Display noninteractive color intensity map (C)
ds2d	Display 2D spectra in whitewash mode (C)
dsww	Display spectra in whitewash mode (C)
ho	Horizontal offset (P)
plww	Plot spectra in whitewash mode (C)
th	Threshold (P)

plane — Currently displayed 3D plane type (P)

Description Stores the type of 3D plane currently displayed within VnmrJ. If plane does not exist, it is created by the macro par3d. The command select, as well as the many macros that make use of select, requires the parameter plane to exist for 3D data sets and to contain an appropriate value.

plane is set automatically by the macro getplane; it can also be set by the macro ft3d if automatic plane extraction is requested at the end of the 3D FT. The order of priority for the plane types is 'f1f3', 'f2f3', and then 'f1f2'. In other words, if getplane is requested to extract the f_1f_3 and the f_2f_3 planes, plane will be set to 'f1f3'. plane can also be set manually.

Values 'f1f3', 'f3f1', 'f2f3', 'f3f2', 'f1f2', or 'f2f1'

See also *NMR Spectroscopy User Guide*

Related

dplane	Display a 3D plane (M)
dproj	Display a 3D plane projection (M)
dsplanes	Display a series of 3D planes (M)
ft3d	Perform a 3D Fourier transform on a 3D FID data set (M,U)
getplane	Extract planes from a 3D spectral set (M)
nextpl	Display the next 3D plane (M)
par3d	Create 3D acquisition, processing, display parameters (C)
path3d	Number of complex points to left-shift np FID (P)
plplanes	Plot a series of 3D planes (M)

prevpl	Display the previous 3D plane (M)
select	Select a spectrum or 2D plane without displaying it (C)

plapt — Plot APT-type spectra automatically (M)

Syntax `plapt<(13Cexp_number)>`

Description Automatically plots APT spectra. The APT spectrum is plotted on top of a standard carbon spectrum if either an experiment with such data is specified or if a file `C13` is found in `curexp+'/subexp'`. If neither such a subfile is found nor an experiment with standard carbon data is specified, the APT spectrum is plotted alone.

Arguments `13Cexp_number` specifies the number, from 1 to 9, of an experiment with a standard ^{13}C spectrum.

Examples
```
plapt
plapt(2)
```

See also *NMR Spectroscopy User Guide*

Related

curexp	Current experiment directory (P)

plarray — Plotting macro for arrayed 1D spectra (M)

Description A generic macro for plotting arrayed 1D spectra. `plarray` is called by the `plot` macro, but can also be used directly. For the plot layout, `procarray` distinguishes between arrays with few elements (6 or less), which will be stacked vertically (no horizontal offset), and spectra with many (greater than 6) elements. Those are stacked horizontally by default, unless there are too many lines, in which case a diagonally stacked display is chosen. Horizontal stacking is mostly adequate for pulse and power calibrations, where there are usually few lines only; diagonally stacked displays/plots are frequently chosen for T_1 and T_2 experiments on entire spectra, often with many lines.

The automatic stacking mode can be overridden by creating and setting a string parameter `stackmode` in the startup macro or before calling `procplot` or `procarray`. Possible values for `stackmode` are `'horizontal'`, `'vertical'`, or `'diagonal'`. DEPT-type spectra can, in principle, also be processed with `procarray`, but no DEPT editing occurs, of course.

See also *NMR Spectroscopy User Guide*

Related

aexppl	Automatic expansion plot (M)
plc	Plot carbon spectrum (M)
plh	Plot proton spectrum (M)
plot	Automatically plot spectra (M)
procarray	Process arrayed 1D spectra (M)
stackmode	Stack control for processing arrayed 1D spectra (P)

plate_glue Define a glue order for plotting and display (U)

Applicability Systems with VAST accessory

Description In a Unix terminal or shell window type plate_glue. The glue order is determined by clicking on the wells to be displayed. Save the glue order file in the user's vnmrsys/templates/glue directory.

See also *NMR Spectroscopy User Guide*

Related
dsvast2d	Display VAST data in a pseudo-2D format (M)
plvast	Plot VAST data in a stacked 1D-NMR matrix (M)
plvast2d	Plot VAST data in a pseudo-2D format (M)

plc Plot a carbon spectrum (M)

Syntax `plc<(pltmod)>`

Description Plots a carbon spectrum based on the parameters pltmod (the options 'off', 'full', and 'fixed' are implemented) and intmod ('off', 'full', and 'partial' are implemented). Peak frequency labels, in ppm, are usually plotted.

Arguments `pltmod` is an alternate value of pltmod for this macro only. The value of the pltmod parameter is not changed.

Examples
```
plc
plc('full')
```

See also *NMR Spectroscopy User Guide*

Related
intmod	Integral display mode (P)
pltmod	Plotter display mode (P)

pLCNMR Plot all forms of LC-NMR data (M)

Applicability VnmrJ 3.1

Description This macro is executed with a button on the LC-NMR display pane (labeled spare). Plots on-flow and stopped-flow 1D LC-NMR data. With on-flow data, the NMR data is plotted with the time-aligned LC detector trace(s) along the left side. In the stopped-flow mode, pLCNMR plots the 1D NMR data for each stop code at a position that it is time-aligned with the relevant LC peak.

Examples pLCNMR(<number of contours>,<contour spacing>)

See also
dLC
pLC
dLCNMR
pLCNMR

plcosy Plot COSY- and NOESY-type spectra automatically (M)

Syntax `plcosy(<'pos'|'neg'><,><levels<,spacing<,exp1D>>>)`

Description Automatically plots 2D COSY- and NOESY-type spectra (homonuclear correlated spectra). Features include the following:

- Keeps the orientation (f_1, f_2) of the spectrum on the screen.
- Plot area is optimized.
- Number of contour levels and their spacing can be selected.
- Negative or positive contours can be suppressed.
- 1D traces can be plotted along both axes; such 1D traces are taken from a full (or reduced) 1D spectrum in an other experiment, or from a subfile from within the current experiment.
- Works correctly for expansions.
- 1D traces can be suppressed, allowing a larger area for the 2D spectrum.
- 1D spectrum can be in any experiment.
- With phase-sensitive spectra using a plotter with one pen or a printer such as a LaserJet, if `'pos'` or `'neg'` are not selected, seven positive levels (or the specified number of positive contours) and one negative level are plotted, to distinguish positive and negative signals.

In multiexperiment mode, for the first plot, the experiment with the 1D spectrum should be specified (at least if it is not in exp1). From then on, the 1D spectrum will be stored *within* the experiment with the 2D spectrum, which allows much faster switching between spectra and also frees the other (1D) experiment for other tasks. Because of this internal storage, the `exp1D` argument is not required for subsequent plots.

Arguments `'pos'` is a keyword to plot only positive contours.

`'neg'` is a keyword to plot only negative contours.

`levels` is the number of contour levels. The default is 7.

`spacing` is the spacing between the contours. The default is 2.

`exp1D` is the experiment in which the proton 1D spectrum resides. This can be a full 1D spectrum, but the referencing must be the same as for the 2D. A negative number suppresses the proton trace. The default is from a subfile.

Examples

```
plcosy
plcosy(12,1.5)
plcosy('pos',7,2,3)
plcosy(7,2,-1)
plcosy('neg')
```

See also *NMR Spectroscopy User Guide*

pldept — Plot DEPT data, edited or unedited (M)

Description: Plots out DEPT data, either edited or not edited.

See also: *NMR Spectroscopy User Guide*

Related:

adept	Automatic DEPT analysis and spectrum editing (C)
autodept	Automated complete analysis of DEPT data (M)
deptproc	Process DEPT data (M)
padept	Perform adept analysis and plot resulting spectra (C)

plexpinfo — Plots Experiment Information

Description: Plots experiment information at a specified position on the page.

Syntax: `plexpinfo(x,y)`

Examples: `plexpinfo(32,210)`

Related: pllogo, pltext, pltime, pap, ppa, pll, plexpinfo

plfid — Plot FIDs (C)

Syntax: `plfid<(<start><,finish><,step><,'all'|'imag'> <,pen>)>`

Description: Plots one or more FIDs. The position of the first FID is governed by the parameters wc, sc, and vpf. A subsequent FID is positioned relative to the preceding FID by the vertical and horizontal offset parameters vo and ho.

Arguments: `start` is the index of a particular FID for arrayed 1D or 2D data sets. For multiple FIDs, `start` is the index of the first FID.

`finish` is the index of the last FID for multiple FIDs. To include all FIDs, set `start` to 1 and `finish` to the parameter `arraydim` (see example).

`step` specifies the increment for the FID index. The default is 1.

`'all'` is a keyword to plot all of the FIDs. This is the default.

`'imag'` is a keyword to plot the imaginary FID channel only. The default is `'all'`.

`pen` is a keyword with the plotter pen number: `'pen1'`, `'pen2'`, `'pen3'`, etc. The default is `'pen1'`.

Examples: `plfid(1,arraydim,3)`

See also: *NMR Spectroscopy User Guide*

Related:

arraydim	Dimension of experiment (P)
dfs	Display stacked FIDs (C)
dfww	Display FIDs in whitewash mode (C)

ho	Horizontal offset (P)
sc	Start of chart (P)
vo	Vertical offset (P)
vpf	Current vertical position of FID (P)
wc	Width of chart (P)

plfit Plot deconvolution analysis (M)

Description Produces a complete output plot of a deconvolution analysis, plotting the observed spectrum, the full calculated spectrum, each individual component, as well as the numerical results of the analysis.

See also *NMR Spectroscopy User Guide*

Related

fitspec	Perform spectrum deconvolution (C)
showfit	Display numerical results of deconvolution (M)
usemark	Use “mark” output as deconvolution starting point (M)

plgrid Plot a grid on a 2D plot (M)

Syntax (1) `plgrid<(<spacing><,><pen>)>`
(2) `plgrid<(start_f2,incr_f2,start_f1,incr_f1<,pen>)>`

Description Plots grid lines over a 2D plot.

Arguments `spacing` specifies the approximate spacing of the grid lines, in cm. The default is intervals of approximately 1 cm, rounded so that the intervals fall at a multiple of 1, 2, or 5 (in Hz) or 1p, 2p, or 5p (in ppm).

`pen` is a keyword with the plotter pen number: `'pen1'`, `'pen2'`, `'pen3'`, etc. The default is `'pen1'`.

`start_f2, incr_f2, start_f1, incr_f1` define the starting and increment frequencies in both f_2 and f_1 for a grid. Add the `p` suffix to a value to enter it in ppm (see last example below).

Examples

```
plgrid
plgrid(2)
plgrid('pen5')
plgrid(1.5,'pen2')
plgrid(1p,0.5p,3p,0.5p)
```

See also *NMR Spectroscopy User Guide*

Related

grid	Draw a grid on a 2D display (C)

plh — Plot proton spectrum (M)

Syntax `plh<(pltmod)>`

Description Plots a proton spectrum based on the parameters pltmod (the options 'off', 'fixed', 'full', and 'variable' are implemented) and intmod ('off', 'full', and 'partial' are implemented).

Arguments `pltmod` is an alternate value of the parameter pltmod for this macro only. The value of the pltmod parameter is not changed.

Examples `plh`
`plh('full')`

See also *NMR Spectroscopy User Guide*

Related		
	intmod	Integral display mode (P)
	pltmod	Plotter display mode (P)
	sp	Start of plot (P)
	wp	Width of plot (P)

plhet2dj — Plot heteronuclear J-resolved 2D spectra automatically (M)

Syntax `plhet2dj<('pos'|'neg'<,levels<,spacing<,exp1D>>>)>`

Description Automatically plots 2D spectra of type HET2DJ (heteronuclear J-resolved 2D spectra) with the following features:

- Displayed portion of the spectrum is plotted in f2-mode
- Plot area is optimized
- Number of contour levels and their spacing can be selected
- Negative or positive contours can be suppressed
- A 1D trace can be plotted along the f_2 axis; such a 1D trace is taken from a full (or reduced) 1D spectrum in an other experiment, or from a file from within the current experiment.
- Expansions are handled correctly
- The 1D trace can be suppressed, which allows using a larger area for the 2D spectrum
- The 1D spectrum can be in any experiment
- With phase-sensitive spectra, if 'pos' or 'neg' are not selected and the plotter has only one pen (also for printers like the LaserJet), the specified number of positive contours are plotted (default is 7), but only one negative level, to distinguish positive and negative signals.

In multiexperiment mode, for the first plot the experiment with the 1D spectrum should be specified (at least if it is not in exp1). From then on, the 1D spectrum is stored *within* the experiment with the 2D spectrum, which allows much faster switching between the spectra and also frees the other 1D experiment for other tasks. Because of this internal storage, the `exp1D` argument is not required for subsequent plots.

Arguments `'pos'` is a keyword to only plot positive contours

`'neg'` is a keyword to only plot negative contours

`levels` is the number of contour levels. The default is 7.

`spacing` is the spacing between the contours. The default is 2.

`exp1D` is the number from 1 to 9 of the experiment in which the 1D spectrum resides. This can be a full 1D spectrum, but the referencing must be the same as for the 2D. A negative number will suppress the 1D trace. The default is 1 (for `exp1`).

Examples

```
plhet2dj
plhet2dj(12,1.5)
plhet2dj('pos',7,2,3)
plhet2dj(7,2,-1)
```

See also *NMR Spectroscopy User Guide*

`plhom2dj` Plot homonuclear J-resolved 2D spectra automatically (M)

Syntax (1) `plhom2dj<(levels<,spacing<,exp1D>>)>`
(2) `plhom2dj('pos'|'neg'<,levels<,spacing<,exp1D>>>)`

Description Automatically plots 2D spectra of type HOM2DJ (homonuclear J-resolved 2D spectra). Features include the following:

- The displayed portion of the spectrum is plotted in f2-mode
- The plot area is optimized
- Number of contour levels and their spacing can be selected
- Negative or positive contours can be suppressed
- A 1D trace can be plotted along the f_2 axis; such a 1D trace is taken from a full (or reduced) 1D spectrum in an other experiment, or from a file from within the current experiment.
- It also works correctly for expansions
- The 1D trace can be suppressed, which allows using a larger area for the 2D spectrum
- The 1D spectrum can be in any experiment
- With phase-sensitive spectra, if `'pos'` or `'neg'` are not selected and the plotter has only 1 pen (also for printers like the LaserJet) 7 or the specified number of positive contours are plotted, but only one negative level, to distinguish positive and negative signals.

In multiexperiment mode, for the first plot the experiment with the 1D spectrum should be specified (at least if it is not in `exp1`). From then on, the 1D spectrum will be stored *within* the experiment with the 2D spectrum, which allows much faster switching between the spectra and also frees the other (1D) experiment for other tasks. Because of this internal storage, the `exp1D` argument is not required for subsequent plots.

Arguments `levels` is the number of contour levels. The default is 7.

`spacing` is the spacing between the contours. The default is 2.

`exp1D` is a number from 1 to 9 for the experiment in which the 1D spectrum resides. The spectrum can be a full 1D spectrum but the referencing must be the same as for the 2D. A negative number will suppress the 1D trace. The default is 1 (for `exp1`).

`'pos'` specifies only plot positive contours.

`'neg'` specifies only plot negative contours.

Examples
```
plhom2dj
plhom2dj(25,1.2)
plhom2dj('pos',7,2,3)
plhom2dj(7,2,-1)
```

See also *NMR Spectroscopy User Guide*

plhxcor Plot X,H-correlation 2D spectrum (M)

Syntax
```
plhxcor(<'pos'|'neg'><,><levels<,spacing
    <,exp1D_H<,exp1D_X>>>>)
```

Description Automatically plots 2D spectra of type HETCOR, COLOC, HMQC, HMBC (direct and indirect detection). Features include the following:

- Keeps the orientation (f_1, f_2) of the spectrum on the screen.
- Plot area is optimized.
- Number of contour levels and their spacing can be selected.
- Negative or positive contours can be suppressed.
- 1D proton and X traces can be plotted along both axes; such 1D traces are taken from full (or reduced) 1D spectra in other experiments or subfile within the current experiment.
- Works correctly for expansions.
- 1D traces can be suppressed, allowing a larger area for the 2D spectrum.
- 1D spectra can be in any experiment.

Arguments `'pos'` is a keyword to plot only positive contours.

`'neg'` is a keyword to plot only negative contours.

`levels` is the number of contour levels. The default is 7.

`spacing` is the spacing between the contours. The default is 2.

`exp1D_H` is a number from 1 to 9 of the experiment in which the proton 1D spectrum resides; this can be a full 1D spectrum, but the referencing must be the same as for the 2D. A negative number will suppress the proton trace. The default is a subfile in the current experiment.

`exp1D_X` is a number from 1 to 9 of the experiment in which the X 1D spectrum resides. A negative number suppresses the X trace. the default is a subfile in the current experiment.

Examples
```
plhxcor(12,1.5)
plhxcor(7,2,3)
plhxcor(7,2,1,3)
plhxcor('pos',7,2,-1,3)
plhxcor(7,2,-1,-1)
plhxcor('neg')
```

See also *NMR Spectroscopy User Guide*

Related `hetcor` Set up parameters for HETCOR pulse sequence (M)

`pll` Plot a line list (M)

Syntax `pll<(x,y,minimum_y)>`

Description Produces a columnar line list on a plotter, similar to what would appear on a printer. `pll` is quite different from the alternative method of plotting peak frequencies using `ppf`. The output of `pll` is automatically formatted into multiple columns, depending on the number of lines.

Arguments `x` is the *x* position of the upper left of the line list.

`y` is the *y* position of the upper left of the line list.

`minimum_y` is the minimum *y* at which to reset back to top.

Examples
```
pll
pll(20,150)
pll(5,wc2max*.8,wc2max*.5)
```

See also *NMR Spectroscopy User Guide*

Related `ppf` Plot peak frequencies over spectrum (M)

`pllogo` Plots Logo

Description Plots a logo.

Syntax pllogo(x,y)

Examples pllogo(32,220)

Related `pllogo, pltext, pltime, pap, ppa, pll, plexpinfo`

`pll2d` Plot results of 2D peak picking (C)

Syntax `pll2d<(options)>`

Description Plots the results of applying the `ll2d` command to pick 2D peaks in a 2D spectrum or a 2D plane of a 3D spectrum. Refer to the description of `ll2d` for a description of the process and the options available.

See also *NMR Spectroscopy User Guide*

Related `ll2d` Automatic and interactive 2D peak picking (C)

Plock Sets Protection Bit for a Parameter

Description Sets the protection bit for a parameter given as an argument. This causes the specified parameter to be read from the appropriate `parlib` entry upon experiment set up, rather than inherited from the current workspace.

Syntax Plock(parameter)

Examples Plock('samplename')

plockport Port number to use to lock out multiple ProTune processes (P)

Syntax `plockport=<value>`

Description The parameter must be created as a real local parameter before it can be used. The parameter is used to override a default port number that is used internally in ProTune to prevent two Java ProTune process from running simultaneously.

Related `protune` Macro to start ProTune (M)
`create` Create new parameter in a parameter tree (C)

plot Automatically plot spectra (M)

Description A universal plotting macro normally called through the `procplot` macro (which by itself serves as processing and plotting facility for automatic experiments). `plot` can also be used directly by the user who then doesn't have to remember specific plotting macros. Of course, the specialized macros can still be called directly if the user know their names.

The main purpose of `plot` is to automatically call the correct specialized plotting macro, depending on the user definition or otherwise on the type of data in the experiment. A plotting macro is selected automatically as follows:

APT spectra:	`plapt`
other, non-arrayed 1D data:	`plot1d`
DEPT type arrayed spectra:	`pldept`

other arrayed 1D spectra:	plarray
J-resolved 2D spectra:	pl2dj
homonuclear correlation 2D spectra:	plcosy
heteronuclear correlation 2D spectra:	plhxcor

Other types of 2D spectra (mostly multiple-quantum 2D spectra such as 2D-INADEQUATE) are not plotted automatically at this time. For phase-sensitive 2D spectra, automatic plotting is only provided if they were acquired using the method described by States, Haberkorn, and others; TPPI spectra are not covered.

Note that plot macros in general should not adjust the phase, the vertical scale, or change the integral size and reset points; these are assumed to be adjusted either by hand or by a suitable processing macro like `procplot` and the macros called therein. The plotting macros only make adjustments in order to make spectrum and parameters fit onto the page the desired way.

See also *NMR Spectroscopy User Guide*

Related

apptype	Application type (P)
execpars	Set up the exec parameters (M)
execplot	Execute plotting macro (P)
plapt	Plot APT spectra (M)
plarray	Plot arrays (M)
plcosy	Plot homonuclear 2D correlation spectra (M)
pldept	Plot DEPT type spectra (M)
plhxcor	Plot heteronuclear correlation spectra (M)
plot1d	Plot 1D spectra (M)
plt2Darg	Plot 2D arguments (P)
procplot	Automatically process FIDs (M)

plot_<pslabel>

Description

plot1d Plotting macro for simple (non-arrayed) 1D spectra (M)

Description A generic macro for plotting non-arrayed 1D spectra using a set of standard macros. `plot1d` is called by the `plot` macro, but can also be used directly. `plot1d` first tries to find a specific macro (e.g., `plh`, `plc`, `plp`) for the current observe nucleus. If such a macro exists, it is called. If a nucleus-specific macro is not found in the command path, a “minimal” 1D plot is produced.

See also *NMR Spectroscopy User Guide*

Related	`plc`	Plot carbon spectrum (M)
	`plh`	Plot proton spectrum (M)
	`plp`	Plot phosphorus spectrum (M)
	`plot`	Automatically plot spectra (M)

plot2D — Plot 2D spectra (M)

Syntax
```
plot2D('pos'|'neg'|'both',levels,spacing,   \
   'top'|'notop'|'proj','side'|'noside'|'proj')
```

Description Checks for the presence of appropriate proton or carbon high-resolution spectra in the directory `userdir+'/data/'+sample` and decides to plot high resolution spectra or a projection depending on whether or not the proton or carbon spectrum exists.

Arguments The `plot2D` macro accepts the following arguments:

`'pos'`	keyword to plot positive contours
`'neg'`	keyword to plot negative contours.
`'both'`	keyword to plot both positive and negative contours.
`levels`	number of levels to be plotted.
`spacing`	spacing between contour levels.
`'top'`	keyword to plot a high-resolution spectrum on the top.
`'notop'`	keyword to plot a non-high-resolution spectrum or projection.
`'proj'`	keyword to plot a projection on top.
`'side'`	keyword to plot a high-resolution spectrum on the side.
`'noside'`	keyword to plot a non-high-resolution spectrum or projection.
`'proj'`	keyword that plots a projection on the side.

Examples `plot2D('pos',2,5,'top','side')`

See also *NMR Spectroscopy User Guide*

Related	`plot`	Automatically plot spectra (M)
	`plotside`	Plot spectrum on side (M)
	`plottop`	Plot spectrum on top (M)
	`plottopside`	Plot spectrum on top and side (M)

plotfile — Plot to a file (M)

Syntax `plotfile('argument')`

Description plots automatically to a file. Supported output formats are: `ps`, `pdf`, `jpg`, `pcl`, `hpgl`. and `png`.

Arguments `auto` — plots automatically.

`manual` — plots contents of printer queue to a file.

`Path and file name` — plots to specified file in the directory specified. Plots to the data directory using the supplied name if no path is specified.

Examples `plotfile('xxx.fid/myplotfile.PDF')` plots will go into saved data directory.

`plotfile('myplotfile.PDF')` - plots will go to `vnmrsys/plots` if FID has not been saved.

`plothiresprep` High resolution plot output preparation (M)

Description Required for the operation of the "Plot HiRes..." popup window to interactively use plottop/plotside of spectra in work spaces EXPn - creates necessary variables.

`plotlcnmr` An LC-NMR plotting macro (M)

Syntax

Applicability VnmrJ 3.1

Description The NMR data for a particular peak can be plotted using plotlcnmr with the number of the peak as an argument. While this can also be accomplished with the `pl` command, plotlcnmr labels the plot with the LC retention time of the peak and the Cascade file name associated with the LC data.

`plotmanual` Plot manually (M)

Description Makes correct choice of printer (for preview) and correct alignment with respect to parameter output, resets back screen to original size & position based on selections made on the Plot page.

`plotlogo` Plots a logo (M)

Description Plots a logo Varian logo using image file located in `/vnmr/iconlib/varianlogo.gif` or a custom logo from location specified in the parameter `plotlogo`.

Reads value for `doplotlogo` (n/y), `plotlogox` (x dimension image), and `plotlogoy` (y dimensions image), and image file in `iconlib`.

plotpreview Creates temporary plots of the current plot output (M)

Syntax `plotpreview<('argument')>`

Description Creates preview of the output from auto-plotting the current spectrum and starts an Acrobat PDF reader. The preview output can be saved in PS, PDF, PCL, HPGL, JPG or PNG formats.

Arguments `no argument` — creates preview of whatever is ready to send to the plotter.

`auto` — creates preview of auto-plot based upon plot macro

`manual` — creates preview of the contents of the print queue.

plotside Plot spectrum on side (M)

Description Plots projection or high-resolution spectrum on the side of a 2D spectrum. `plotside` is used with `plot2D` and is not useful by itself.

See also *NMR Spectroscopy User Guide*

Related `plot2D` Plot 2D spectra (M)

plotter Plotter device (P)

Description Sets the plotter in use on the system.

Values A string with entries such as `'DraftPro'`, `'ThinkJet_96'`, `'LaserJet_300'`, `'jim'`, `'varian1'`, and `'Laser1'`.

See also *NMR Spectroscopy User Guide*

Related `setplotdev` Return characteristics of a named plotter (C)
`showplotter` Show list of currently defined plotters and printers (M)

plottop Plot spectrum on top (M)

Description Plots projection or high resolution spectra on the top of a 2D spectrum. `plottop` is used with `plot2D` and is not useful by itself.

See also *NMR Spectroscopy User Guide*

Related `plot2D` Plot 2D spectra (M)

plottopside Plot spectrum on top and side (M)

Description Plots projection or high-resolution spectrum on the top and side of a 2D spectrum. `plottopside` is used with `plot2D` and is not useful by itself.

See also *NMR Spectroscopy User Guide*

Related `plot2D` Plot 2D spectra (M)

plp Plot phosphorus spectrum (M)

Syntax `plp<(pltmod)>`

Description Plots a phosphorus spectrum based on the parameters `pltmod` (the options `'off'`, `'full'`, and `'fixed'` are implemented) and `intmod` (`'off'`, `'full'`, and `'partial'` are implemented). Peak frequency labels, in ppm, are usually plotted.

Arguments `pltmod` is an alternate value of `pltmod` for this macro only. The value of the `pltmod` parameter is not changed.

Examples `plp`
`plp('full')`

See also *NMR Spectroscopy User Guide*

Related `intmod` Integral display mode (P)
`plh` Plot proton spectrum (M)
`pltmod` Plotter display mode (P)

plplanes Plot a series of 3D planes (M)

Syntax `plplanes(start_plot,stop_plot<,'pos'|'neg'>`
`<,number_levels><,spacing>)`

Description Creates the 2D contour plots for a subset of the 3D planes specified by the parameter `plane`.

Arguments `start_plot` specifies the number, greater than 0, of the 3D plane with which plotting is to begin.

`stop_plot` specifies the number of the 3D plane with which plotting is to end. If `start_plot` is greater than `stop_plot`, only the first plane, whose number is `start_plot`, is plotted. The range of `stop_plot` depends on the value of the parameter `plane`:

- if `plane='f1f3'`, `stop_plot` is between `0` and `fn2/2`
- if `plane='f2f3'`, `stop_plot` is between `0` and `fn1/2`
- if `plane='f1f2'`, `stop_plot` is between `0` and `fn/2`

'pos' is a keyword specifying that phase-sensitive spectra plot positive peaks only. The default is to plot both positive and negative peaks.

'neg' is a keyword specifying that phase-sensitive spectra plot negative peaks only. The default is to plot both positive and negative peaks.

levels is maximum number of contour levels to plot. The default is 4.

spacing is relative intensity of successive contour levels. The default is 2.

Note that the optional arguments 'pos'|'neg', number_levels, and spacing are for the VnmrJ plotting command pcon.

Examples
```
plplanes(1,3)
plplanes(2,3,'pos',4)
```

See also *NMR Spectroscopy User Guide*

Related

dplane	Display a 3D plane (M)
dproj	Display a 3D plane projection (M)
dsplanes	Display a series of 3D planes (M)
getplane	Extract planes from 3D spectral data set (M)
nextpl	Display the next 3D plane (M)
path3d	Path to currently displayed 2D planes from a 3D data set (P)
pcon	Plot contours on a plotter (C)
plane	Currently displayed 3D plane type (P)
prevpl	Display the previous 3D plane (M)

plt2Darg Plot 2D arguments (P)

Applicability Liquids

Description Specifies options for contours and 1D projections on 2D plots, used by the plot2D macro. The plot options are selected on the Defaults page in the Acquire folder for most 2D sequences.

Related plot2D Plot 2D spectra (M)

pltext Plot text file (M)

Syntax
```
pltext<(<file><,x<,y<,width>>>)>
  <:$x_next,$y_next,$y_increment>
```

Description Plots a text file.

Arguments file is the name of a text file. The default is the current experiment text file.

x and y are coordinates, in mm, of the first line of text. This positions the location of the output. The default is the upper left-hand corner of the page.

width is the maximum column text width, in characters. pltext uses a word wrap to make the text fit into the width specified.

$x_next and $y_next are the coordinates where the start of the next line would have been plotting. This is useful for subsequent character plotting.

$y_increment is the vertical increment between lines.

Examples
```
pltext
pltext(wcmax-70)
pltext(userdir+'/exp3/text')
pltext(100,100)
pltext(userdir+'/exp4/text',200,200,24)
pltext:$x,$y,$dy
```

See also *NMR Spectroscopy User Guide*

Related

dtext	Display a text file in the graphics window (C
ptext	Print out a text file (M)
text	Display text or set new text for current experiment (C)
userdir	User directory (P)

pltmod Plotter display mode (P)

Description Controls plotting of a proton, carbon, or phosphorus spectrum.

Values 'off' sets no plotting.

'fixed' takes sp and wp as is.

'full' adjusts sp and wp to plot the full spectrum.

'variable' adjusts sp and wp to plot only the region of interest.

See also *NMR Spectroscopy User Guide*

Related

plc	Plot carbon spectrum (M)
plh	Plot proton spectrum (M)
plp	Plot phosphorus spectrum (M)
sp	Start plot (P)
wp	Width of plot (P)

plvast Plot VAST Data in a stacked 1D-NMR matrix format

Applicability VnmrJ 3.1

Description If an array of 1D spectra have been acquired (in particular if a block of 96 spectra have been acquired using VAST automation, especially in a microtiter-plate format), and if these spectra have been glued into a

reconstructed 2D dataset (see vastglue), this macro will arrange and plot them (on the plotter) in a convenient 8 × 12 sample format (as a matrix of 1D spectra).
Uses a file (template) created by plate_glue to display a matrix of data. The number of spectra displayed, and their order, are controlled by the template file. Each "little spectrum" is labeled with its respective alphanumeric coordinates. The modulo number controls how many spectra appear per row.

Examples plvast(<display order>, <modulo>)

See also dsvast
dsvast2d
plvast
plvast2d
intvast
pintvast
plateglue
vastglue
vastget

plvastget Plot VAST spectral data in a vertical stacked plot mode

Applicability VnmrJ 3.1

Description This macro selects and plots the spectra from any arbitrary well or wells using the label(s) as an argument. The spectra are displayed in a dss stacked plot.

Examples vastget("B6","B7","C11","G3") will display four spectra.

See also dsvast
dsvast2d
plvast
plvast2d
intvast
pintvast
plateglue
vastglue
vastget

plvast_replot Replot VAST spectral data one spectrum per page of paper (M)

Applicability VnmrJ 3.1

Description This macro plots all the spectra in a glued dataset, one spectrum per page of paper. This mimics the plots obtained automatically during data acquisition, but allows the data to be rephased or reprocessed.

Examples plvast_replot(96) will replot all 96 spectra

See also dsvast
dsvast2d
plvast
plvast2d
intvast
pintvast
plateglue
vastglue
vastget

plvast2d Plot VAST data in a stacked pseudo-2D format (M)

Applicability Systems with the VAST accessory.

Syntax `plvast2d<(number)>`

Description If an array of 1D spectra have been acquired (in particular if a block of 96 spectra has been acquired using VAST automation, especially in a microtiter-plate format) and if these spectra have been glued into a reconstructed 2D dataset (see vastglue), `plvast2d` will arrange and plot them (on the plotter) in a convenient pseudo-2D format (almost like an LC-NMR chromatogram). Well labels are not attached to the spectra and spectra are plotted with 12 spectra per row.

Arguments `number` specifies that only spectra from 1 through `number` should be plotted. The default is to plot all the spectra (from 1 through arraydim).
An optional argument (plvast(##)) allows one to specify that only spectra from 1 through ## should be plotted.

See also *NMR Spectroscopy User Guide*
plvast2d
pintvast

Related

dsast2d	Display VAST data in a pseudo-2D format (M)
dsvast	Display VAST data in a stacked 1D-NMR matrix (M)
plvast	Plot VAST data in a stacked 1D-NMR matrix (M)

plww Plot spectra in whitewash mode (C)

Syntax `plww<(start,finish,step><,'all'>)>`

Description Plots one or more spectra in whitewash mode (after the first spectra, each spectra is blanked out in regions in which it is behind an earlier spectra).

Arguments `start` — index of the first spectra when plotting multiple spectra. It is also the index number of a particular trace to be plotted when plotting arrayed 1D spectra or 2D spectra. The default is to plot all spectra.

`finish` — index of the last spectra when plotting multiple spectra.

`step` — increment for the spectral index when plotting multiple spectra, default is 1.

`'all'` — (default) keyword to plot all spectra in the array.

See also *NMR Spectroscopy User Guide*

Related		
	`dss`	Display stacked spectra (C)
	`dsww`	Display spectra in whitewash mode (C)
	`pl`	Plot spectra (C)

pmode — Processing mode for 2D data (P)

Description Specifies the type of 2D spectral data that the 2D Fourier transform (FT) will yield. `pmode` is in the processing group.

Values `''` (null string, shown by two single quotes with no space in between) specifies a processing mode in which it is not possible to change either the f_2 or f_1 display mode after the 2D FT. If the f_2 display mode has been set to phased (`dmg='ph'`), each f_2 spectrum is phase rotated using the phase constants `rp` and `lp` prior to the FT along the second dimension. If the f_2 display mode has been set to power (`dmg='pwr'`) or absolute-value (`dmg='av'`), however, the f_2 spectrum is not processed any further after the first FT. The complex t_1 interferograms are handled in a similar manner. If the f_1 display mode has been set to phased (`dmg1='ph1'`), each f_1 spectrum is phased using the phase constants `rp1` and `lp1`. If the display mode has been set to power (`dmg1='pwr1'`) or to absolute value (`dmg1='av1'`), the appropriate magnitude calculation is performed, with the result being placed in the real part of the appropriate complex datum and a 0 being placed in the imaginary part. At the end of the 2D transform, the spectral data file `datdir/data` is reduced from complex data to real data ("VnmrJ REDUCE" display message).

`'partial'` specifies a processing mode in which it is not possible to change the f_2 display mode after the 2D FT. It is possible, however, to select between the three f_1 display modes without having to reprocess the 2D data. If the f_2 display mode has been set to phased (`dmg='ph'`), each f_2 spectrum is phase rotated using the phase constants `rp` and `lp` prior to FT along the second dimension. If the f_2 display mode is set to power (`dmg='pwr'`) or absolute value (`dmg='av'`), the f_2 spectrum is not processed any further after the first FT. Regardless of the requested f_1 display mode, no further processing is performed by `ft2d` on the f_1 spectra after the second FT. The calculations on 2D spectral data necessary to achieve the requested f_1 display mode are performed by `dcon` or `dconi`. If `pmode` does not exist, it is assigned a value of `'partial'` internal to VnmrJ.

`'full'` specifies a processing mode in which it is possible to select between the three display modes for each dimension without having to reprocess the 2D data. Regardless of any requested display mode, no display mode processing is performed by `ft2d` on the f_2 spectra after the first or second FT.

The hypercomplex data structure for the 2D time domain data is:

```
{Re(t1)Re(t2), Re(t1)Im(t2), Im(t1)Re(t2),
   Im(t1)Im(t2)}
```

and is experimentally composed by the pulse sequence generation arraying mechanism. The hypercomplex data structure for the t_1 interferograms is:

```
{Re(t1)Re(F2), Re(t1)Im(F2), Im(t1)Re(F2),
   Im(t1)Im(F2)}
```

where `Re` represents the real part and `Im` represents the imaginary part. A hypercomplex FT along t_1 yields a hypercomplex 2D spectrum with the following data structure per hypercomplex point:

```
{Re(F1)Re(F2), Re(F1)Im(F2), Im(F1)Re(F2),
   Im(F1)Im(F2)}
```

Note that if `pmode='full'`, the `ft2d` program will require an array index or coefficients for the construction of the t_1 interferograms.

See also *NMR Spectroscopy User Guide*

Related

`av`	Set abs. value mode in directly detected dimension (C)
`av1`	Set abs. value mode in 1st indirectly detected dimension (C)
`dcon`	Display noninteractive color intensity map (C)
`dconi`	Interactive 2D data display (C)
`dmg`	Data display mode in directly detected dimension (P)
`dmg1`	Data display mode in 1st indirectly detected dimension (P)
`ft1d`	Fourier transform along f_2 dimension (C)
`ft2d`	Fourier transform 2D data (C)
`ph`	Set phased mode in directly detected dimension (C)
`ph1`	Set phased mode in indirectly detected dimension (C)
`pwr`	Set power mode in directly detected dimension (C)
`pwr1`	Set power mode in 1st indirectly detected dimension (C)
`wft1d`	Weight and Fourier transform 2D data (C)
`wft2d`	Weight and Fourier transform 2D data (C)

poly0 Display mean of the data in regression.inp file (M)

Description Calculates and displays the mean of data in the file `regression.inp`.

See also *User Programming*

Related

`averag`	Calculate average and standard deviation of input (C)
`expl`	Display exponential or polynomial curves (C)

powerfit

Fits the diffusional attenuation calcuated by decay_gen to the exponential of a power series in the calibration of the non-uniformity of pulsed field gradients.

Syntax `powerfit()`

powerfit(ncoef)

Applicability VnmrJ 3.1

Description Used in the calibration of non-uniform field gradients to fit the diffusional decay calculated by `decay_gen` to the exponential of a power series.

Arguments powerfit has one optional argument, the number of coefficients in the power series. The default is 8.

See also `decay_gen`

`gradfit`

`nugcalib`

`profile_int`

pp

Decoupler pulse length (P)

Description Sets the decoupler pulse length for use by pulse sequences such as DEPT, HET2DJ, and HETCOR.

See also *NMR Spectroscopy User Guide*

Related

`AC1-AC9`	Automatic calibration (M)
`Dept`	Set up parameters for DEPT experiment
`dhp`	Decoupler high-power control with class C amplifier (P)
`dpwr`	Power level for first decoupler with linear amplifier (P)
`hetcor`	Set up parameters for HETCOR pulse sequence (M)
`p1`	First pulse width (P)
`pw`	Pulse width (P)

ppa

Plot a parameter list in plain English (M)

Syntax `ppa<(x<,y>)>`

Description Plots parameters in plain English (instead of in a table with parameter names and their values as plotted by the parameter `pap`).

Arguments `x` controls the x offset, in mm, from the lower left of the plot to the starting position (upper left) of the parameter list. The default is a preset position on the page (upper left corner).

`y` controls the y offset, in mm, from the lower left of the plot to the starting position (upper left) of the parameter list. Default is a preset position on the page (upper left corner).

Examples ppa
ppa(10)
ppa(wcmax-80,wc2max*.9)

See also *NMR Spectroscopy User Guide*

Related

bpa	Plot boxed parameters (M)
hpa	Plot parameters on special preprinted chart paper (C)
pap	Plot out "all" parameters (C)
pltext	Plot a text file (M)

ppcal Proton decoupler pulse calibration (M)

Description Proton decoupler pulse calibration for DEPT, HETCOR, INEPT, etc.

See also *NMR Spectroscopy User Guide*

Related

AC1S-AC11S	Automatic calibration (M)
d2pul	Set up parameters for D2PUL pulse sequence (M)
Dept	Set up parameters for DEPT experiment
hetcor	Set up parameters for HETCOR pulse sequence (M)
inept	Set up parameters for INEPT pulse sequence (M)

ppf Plot peak frequencies over spectrum (C)

Syntax (1) ppf<(<'noll'><,'pos'><,noise_mult><,'top'>)>
(2) ppf<(<'noll'><,'pos'><,noise_mult><,'leader'>
<,length>)>

Description Plots peak frequencies, in units specified by the axis parameter, in the plotter device. Only those peaks greater than th high are selected. Two basic modes of label positioning are available: labels placed at the top, with long "leaders" extending down to the tops of the lines (syntax 1 using the 'top' keyword), or labels positioned just above each peak, with short leaders (syntax 2 using the 'leader' keyword). The default is short leaders.

Arguments 'noll' is a keyword to plot frequencies using the last previous line listing.

'pos' is a keyword to plot positive peaks only ('noneg' is the same as 'pos').

noise_mult is a numerical value that determines the number of noise peaks plotted for broad, noisy peaks. The default is 3. A smaller value results in more peaks, a larger value results in fewer peaks, and a value of 0.0 results in a line listing containing all peaks above the threshold th. Negative values of noise_mult default to 3. The noise_mult argument is inactive when the 'noll' keyword is specified.

`'top'` is a keyword to plot labels at the top with long leaders. In this mode, the height of labels is varied by changing the parameter `wc2`.

`'leader'` is a keyword to plot labels positioned just above each peak with short leaders.

`length` specifies the leader length, in mm, if labels are positioned just above each peak. The default length is 20 mm.

Examples
```
ppf('pos')
ppf('leader',30)
ppf('top','noll')
ppf('pos',0.0,'leader',30)
```

See also *NMR Spectroscopy User Guide*

Related

`axis`	Axis label for displays and plots (P)
`dpf`	Display peak frequencies over spectrum (C)
`dpir`	Display integral amplitudes below spectrum (C)
`dpirn`	Display normalized integral amplitudes below spectrum (M)
`pir`	Plot integral amplitudes below spectrum (C)
`pirn`	Plot normalized integral amplitudes below spectrum (M)
`th`	Threshold (P)

pph Print pulse header (M)

Syntax `pph(file)`

Description Prints out the shape file header (i.e., all lines starting with #).

Arguments `file` is the name of the shape file, including the extension.

Examples `pph('shgrad.GRD')`

See also *NMR Spectroscopy User Guide*

Related

`Pbox`	Pulse shaping software (U)

ppmm Resolution on printers and plotters (P)

Description An internal software parameter, selected automatically based on the plotter configuration, that contains the resolution in dots/mm on raster graphics printers. On pen plotters, `ppmm` contains the resolution of points drawn. On PostScript printers, `ppmm` adjusts linewidths.

pprofile Plot pulse excitation profile (M)

Syntax `pprofile<(axisflag<,profile<,shapefile>>)>`

Description Plots the X, Y and Z excitation (inversion) profile for a pulse shape that has been generated with the Pbox software. If shape names is not provided, the last simulation data stored in the `shapelib/pbox.sim` file are plotted.

Arguments The `axisflag` and `profile` arguments can be given in any order.

`axisflag` is `'y'` to display the full spectrum and a frequency scale, or `'n'` to suppress the scale and spectrum. The default is `'n'`.

`profile` is a character string identifying the desired profile. `'xyz'` selects X, Y, and Z (inversion) profiles; `'xy'` selects only the excitation (transverse) profiles; `'x'` selects only the X transverse excitation profile; and `'z'` selects only the inversion profile. The default is `'xyz'`.

`shapefile` is the name of a *.RF or *.DEC file, including the extension.

Examples

```
pprofile
pprofile('y','x')
pprofile('xy','n','softpls.RF')
```

See also *NMR Spectroscopy User Guide*

Related

dprofile	Display pulse excitation profile (M)
Pbox	Pulse shaping software (U)

pps Plot pulse sequence (C)

Syntax `pps<(file<,x,y,width,height>)>`

Description Plots pulse sequences. The plotted picture consists of three to five parts. At the top is the transmitter pulse sequence. Below that is the decoupler pulse sequence. Next is the second decoupler pulse sequence or gradients, depending on the program. At the bottom is the status.

The parameter of each pulse is plotted if its length is less than 30 letters. The value of each pulse is also plotted. If its value is less than zero, a question mark “?” is plotted. The time units are displayed as letters (`s`, `m`, or `u`). The height of pulses are plotted according to their power level.

Arguments `file` specifies the pulse sequence to be plotted. The default is `seqfil`.

`x,y` specifies the start of the plotting position with respect to the lower-left corner of the plotter.

`width,height` are in proportion to `wcmax` and `wc2max`.

Examples

```
pps
pps('s2pul')
pps(3,50)
```

See also *NMR Spectroscopy User Guide*

Related

dps	Display pulse sequence (C)
seqfil	Pulse sequence name (P)
wcmax	Maximum width of chart (P)
wc2max	Maximum width of chart in second direction (P)

prealfa — Specify a delay for longer ring down (P)

Applicability: Systems with Varian, Inc. Cold Probes

Description: Specify a delay to be used in situations when there is a longer ring down of rf following the last rf pulse.

This parameter is only active when `qcomp='y'`. `prealfa` should be created as a local parameter of type `pulse` or `delay`. This parameter must be created as a local parameter of the type `pulse` for SpinCad Sequences.

If it is desired to use the software computed value for this delay, destroy the `prealfa` parameter.

Values: User set `prealfa` value that may be slightly adjusted by the software to better optimize the DSP parameters.

preAmpConfig — Set the band of the preamp, high or low, connected to each transmitter channel.

Syntax

Applicability: VnmrJ 3.1

Description: Sets the band of the preamp, high or low, connected to each transmitter channel. This global parameter is a string whose entries are the characters "H" "L" and"X" separated by commas. The number of characters must equal the number of channels, numrfch. The characters from left to right refer to the transmitter channels "1","2","3".. etc, which for VNMRS correspond to preamps or RF cable outputs from the Front End and fom the transmitters in the RF card cage, right to left. Set: 'H' for a highband preamp, 'L' for a lowband preamp and 'X' for no preamp.

`probeConnect` and `preAmpConfig` are required for all experiments that use transmitters "3" and "4" as Obs or Dec. Create `probeConnect` and `preAmpConfig` as global parameters on the commandline with:

create('probeConnect','string','global')

create('preAmpConfig','string','global')

If `probeConnect` is present it will overide the transmitter settings in the 'current' parameter rfchannel.

If `probeConnect` and `preAmpConfig` are not created and rfchannel is not present the default transmitters are:

Obs (highband) "1" Dec (lowband) "2"

Obs (lowband) "2" Dec (highband "1"

Dec2 (highband or lowband) "3"

Dec3 (highband or lowband) "4"

If `probeConnect` and `preAmpConfig` are not created the default preamps are presumed to be:

"1" highband

"2" lowband

"3" lowband

"4" lowband

probeConnect and preAmpConfig must both exist or both be absent. If they exist both must have correct values. An empty string or incorrect string in either parameter will cause errors in channel selection.

Examples probeConnect = 'H1 C13 F19 N15', preAmpConfig = 'HLHL', numrfch = 4, tn = 'H1', dn = 'C13', dn2 = 'N15' causes:

Obs on channel "1"

Dec on channel "2"

Dec2 on channel "4"

probeConnect = 'H1 N15 F19 C13', preAmpConfig = 'HLHL', numrfch = 4, tn = 'H1', dn = 'C13', dn2 = 'N15' causes:

Obs on channel "1"

Dec on channel "4"

Dec2 on channel "2"

probeConnect = 'H1 N15 F19 C13', preAmpConfig = 'HLHL', numrfch = 4, tn = 'C13',dn = 'H1', dn2 = 'N15' causes:

Obs on channel "2"

Dec on channel "1"

Dec2 on channel "4"

prep Run prepare acquisition macro (M)

Applicability Imaging

Description Run the prepare acquisition macro specified by the `execprep` parameter. Usually only called from panels.

Related execprep Execute prepare macro (P)

Presat Set up parameters for presat ^{1}H experiment (M)

Description Set up parameters for presat ^{1}H experiment with solvent suppression.

prevpl Display the previous 3D plane (M)

Description Displays 2D color map of the previous 3D plane in the set of planes defined by the parameters plane and path3d. For example, if dplane(40) has just been executed, prevpl results in the display of 3D plane 39 of that set. (If prevpl immediately follows the command dproj, an error results because there is no 3D plane whose number is –1.) prevpl is more efficient than dplane or dproj because the 3D parameter set (procpar3d) is not loaded into VnmrJ. It is assumed to have already been loaded by, for example, dplane or dproj.

See also *NMR Spectroscopy User Guide*

Related

dplane	Display a 3D plane (M)
dproj	Display a 3D plane projection (M)
dsplanes	Display a series of 3D planes (M)
getplane	Extract planes from a 3D spectral data set (M)
nextpl	Display the next 3D plane (M)
path3d	Path to currently displayed 2D planes from a 3D data set (P)
plane	Currently displayed 3D plane type (P)
plplanes	Plot a series of 3D planes (M)

prescan Study queue prescan (P)

Description This parameter keeps track of the type and status of the prescans in the study queue.

Related

cqexp	Load experiment from protocol (M)
cqrset	Reset study queue parameters (M)
sqexp	Load experiment from protocol (M)
sqreset	Reset study queue parameters for imaging (M)

prescan_CoilTableRead or update the CoilTable File (M)

Syntax prescan_CoilTable(action,rfcoil)

Description Manages the CoilTable file in ~/vnmrsys. Reads information about rfcoil into the global parameter coil_param; updates/adds information for rfcoil from coil_param; removes the rfcoil entry from CoilTable.

Arguments actions for the specified rfcoil are:

read
add
update
remove

Examples prescan_CoilTable('read','main')

prescan_tn — Return tn string for a given atomic number (M)

Syntax `prescan_tn(number):str`

Description Returns tn string for a given atomic number; for H1, c13, F19, P31, Na23, Xe129 only.

Arguments `Number` is the atomic number.

`str` is a string that can be assigned to tn.

Examples `prescan_tn(23):tn`

presig — Preamp Signal Level Selection Parameter (parameter)

Syntax

Applicability VnmrJ 3.1

Description This parameter is to be used with systems that support large signal handling at the `preamp`. It allows the user to select high signal handling "`presig=`'h'" or low signal handling "`presig=`'l'". Currently there are two types of preamps that support this capability.

UnityPlus Spectrometers with Selectable Large-Signal Mode Preamps support this capability by allowing a current increase the preamp. This allows larger signals, and the overall signal level will be slightly higher.

UnityPlus SIS Imaging Spectrometers support this capability using attenuation and a current increase. This allows larger signals and results in a lower overall signal level.

The use of this parameter to control the hardware depends on the Magnet Leg Driver Board Configuration ID being set to 16 for SIS Imaging Systems or 1 for UnityPlus Spectrometers with Selectable Large-Signal Mode Preamp.

Arguments 'h' signifies high signal mode at the preamp.

'l' signifies low signal mode at the preamp.

'n' signifies "not used" and will default to low signal mode at the preamp if the hardware is present.

Related gain

printer — Printer device (P)

Description Selects the printer in use on the system.

Values A string with entries such as `'ThinkJet_96'`, `'LaserJet_300'`, `'jim'`, `'varian1'`, and `'Laser1'`.

See also *NMR Spectroscopy User Guide*

Related showplotter Show list of currently defined plotters and printers (M)

printfile Path to the print-to-file image (P)

Description Defines the path where an image is saved if it is printed to a file.

printformat Format of saved-to-file image (P)

Description The format of the image to be printed to a file.

Values `'jpeg'`, `'gif'`, `'tiff'`, `'bmp'`

printlayout Layout of printed image (P)

Description The layout of the printed image.

Values `'portrait'` or `'layout'`

printoff Stop sending text to printer and start print operation (C)

Syntax `printoff<('clear'|file)>`

Description Stops redirection of output to printer caused by the `printon` command and starts the print operation. **The command `printoff` must be entered to obtain output on the printer.** Actual printing is controlled by the `vnmrprint` script in the `bin` subdirectory of the system directory. `printoff` can also clear the data in the current print file or save data to a specified file name (i.e., print or plot to a file).

Arguments `'clear'` is a keyword to clear the print file made so far.

`file` specifies the name of a file to save the printout. If the file already exists, it is overwritten.

Examples
```
printoff
printoff('clear')
printoff('vnmrsys/papers/peaks.list')
```

See also *NMR Spectroscopy User Guide*

Related
`printon`	Direct text output to printer (C)
`vnmrprint`	Print text files (U)

printon Direct text output to printer (C)

Description Sends information to the printer that is normally displayed in the text window. After using `printon`, output from commands that use the text window, such as `dg` and `cat`, is sent to the printer and does not appear on the screen. The value of the parameter printer is used to select which printer is used.

See also *NMR Spectroscopy User Guide*

Related cat	Output one or more files to output text window (C)
dg	Display group of acquisition/processing parameters (C)
printer	Printer device (P)
printoff	Stop sending text to printer and start print operation (C)

printregion Screen region to be printed (P)

Description The region of the screen to be printed or saved to a file.

Values 'vnmrj' -- entire VnmrJ interface.
'graphics' -- the graphics area of the VnmrJ interface.
'frames' -- selected frames from the graphics area.

printsize Size of printed image (P)

Description The size of the printed image.

Values 'quarterpage', 'halfpage', 'page'

printsend Defines where image will print (P)

Description Defines whether the selected image will sent to a file or a printer.

Values 'file' or 'printer'

probe Probe type (P)

Description Contains a string with the name of the probe currently in the magnet. This parameter is set automatically when the addprobe macro is entered. The getparam and setparams macros use probe to retrieve and write parameters into the current probe file.

See also *NMR Spectroscopy User Guide*

Related addnucleus	Add new nucleus to existing probe file (M)
addprobe	Create new probe directory and probe file (M)
getparam	Receive parameter from probe file (M)
setparams	Write parameter to current probe file (M)

`probeConnect` Specify which nucleus can be acquired on each RF channel (P)

Applicability VNMRS and 400 MR

Syntax `probeConnect = 'nuc1 nuc2 nuc3...'`

Description Global string parameter that does not exist by default. If present, PSG uses it to determine which RF channel to connect to a given nucleus. The string consists of a series of space-separated nuclei. A nucleus 'X' may be used only once in the string to match any nucleus. The parameter must match the hardware connections. If the parameter does not match the hardware connections or does not exist, default settings are used. Default settings are to use the first channel for `tn` for high band observe, and the second channel for `tn` for low band observe.

Values Any nucleus name used for `tn`, or 'X'.

Examples `create('probeConnect','string','global')`

`probeConnect = 'H1 C13'` maps H1 to channel 1, C13 to channel 2

`probeConnect = 'H1 P31 X'` maps H1 to channel 1, P31 to channel 2, any nucleus to channel 3.

See also *VnmrJ User Programming*

Related

`tn`	Nucleus for observe transmitter (P)
`dm`	Nucleus for first decoupler (P)
`dm2`	Nucleus for second decoupler (P)
`dm3`	Nucleus for third decoupler (P)

`Probe_edit` Edit probe for specific nucleus (U)

Syntax (UNIX) `Probe_edit probe nucleus`

Description Opens a dialog box showing all the parameters related to a specific nucleus from the probe table.

Arguments `probe` is the name of the probe.

`nucleus` is the specified nucleus from the probe table.

Examples `Probe_edit 5mmSW H1`

Related `probe_edit` Edit probe for specific nucleus (M)

`probe_edit` Edit probe for specific nucleus (M)

Syntax `probe_edit(probe,nucleus)`

Description Opens a dialog box showing all the parameters related to a specific nucleus from the probe table.

Arguments `probe` is the name of the probe.

nucleus is the specified nucleus from the probe table.

Examples probe_edit('5mmSW','H1')

probe_edit(probe,tn)

Related Probe_edit Edit probe for a specific nucleus (U)

probe_protection Probe protection control (P)

Description Controls the power check for probe protection.

See also *NMR Spectroscopy User Guide*

proc Type of processing on np FID (P)

Description Specifies the type of data processing to be performed upon the np (t_2) FID. Similarly, parameters proc1 and proc2 specify the type of data processing on the ni (t_1) and ni2 interferograms, respectively.

All Varian data must be processed along np with a complex Fourier transform (FT). Sequentially sampled Bruker data (the usual case) must be processed along this dimension with a real FT, while simultaneously sampled Bruker data must be processed with a complex FT.

Pure absorptive 2D data collected by the States-Haberkorn (hypercomplex) method must be processed along ni or ni2 with a complex FT.

Pure absorptive 2D data collected by the TPPI method on a Varian spectrometer can be processed in one of two ways, depending upon how the data was collected:

phase=3	Complex FT, i.e., proc1='ft' (standard way)
phase=1,4	Real FT, i.e., proc1='rft' (new way)
phase2=3	Complex FT, i.e., proc2='ft'
phase2=1,4	Real FT, i.e., proc2='rft'

Pure absorptive 2D data collected by TPPI method on a Bruker spectrometer must be processed along ni with a real FT (i.e., proc1='rft').

Values 'ft' specifies complex FT data processing.

'rft' specifies real FT data processing.

'lp' specifies linear prediction processing on complex data. If 'lp' is selected, additional parameters must be set to fully define how the time-domain data is to be processed; see the description of the addpar command.

See also *NMR Spectroscopy User Guide*

Related
addpar Add selected parameters to the current experiment (M)
ni Number of increments in 1st indirectly detected dimension (P)
np Number of data points (P)
parlp Create parameters for linear prediction (C)
phase Phase selection (P)
phase2 Phase selection for 3D acquisition (P)
proc1 Type of processing on ni interferogram (P)
proc2 Type of processing on ni2 interferogram (P)

proc1 Type of processing on ni interferogram (P)

Description Specifies the type of data processing to be performed upon the ni (t_1) interferogram (2D). Refer to the description of proc for further information.

Values 'ft' specifies complex Fourier transform (FT) data processing.
'rft' specifies real FT data processing.
'lp' specifies linear prediction processing on complex data. If 'lp' is selected, additional parameters must be set to fully define how the time-domain data is to be processed; see the description of the addpar command.
'ht' specifies Hadamard transform processing. If 'ht' is selected, additional parameters must be set with the addpar command. In addition, the data set must be acquired using a Hadamard pulse sequence.

See also *NMR Spectroscopy User Guide*

Related
addpar Add selected parameters to the current experiment (M)
ni Number of increments in 1st indirectly detected dimension (P)
proc Type of processing on np FID (P)

proc1d Processing macro for simple (non-arrayed) 1D spectra (M)

Description A generic macro for processing non-arrayed 1D spectra using a set of standard macros. proc1d is called by the procplot macro, but can also be used directly. proc1d first tries to find a macro of the form {tn}p with the name of the observe nucleus in lower case (e.g., h1p, c13p). If such a macro exists, it is called. If such a nucleus-specific macro is not found in the command path, minimal 1D processing is performed (the intent is to provide a well-processed spectrum in most cases): Fourier transformation (using pre-set weighting functions), automatic phasing (aphx macro), automatic integration (integrate macro), vertical scale adjustment (vsadj macro), avoiding excessive noise (noislm macro), and threshold adjustment (thadj macro). proc1d does not work with arrayed 1D spectra: use deptproc (for DEPT-type spectra) or procarray (for all other arrayed 1D data).

See also *NMR Spectroscopy User Guide*

Related

aphx	Perform optimized automatic phasing (M)
c13p	Process 1D carbon spectra (M)
deptproc	Process arrayed dept type spectra (M)
h1p	Process 1D proton spectra (M)
integrate	Automatically integrate 1D spectrum (M)
noislm	Avoids excessive noise (M)
procarray	Process arrayed 1D spectra (M)
procplot	Automatically process FIDs (M)
thadj	Adjust threshold (M)
vsadj	Adjust vertical scale (M)

proc2 — Type of processing on ni2 interferogram (P)

Description Specifies the type of data processing to be performed upon the ni2 interferogram (3D). Refer to the description of proc for further information.

Values 'ft' specifies complex Fourier transform (FT) data processing.

'rft' specifies real FT data processing.

'lp' specifies linear prediction processing on complex data. If 'lp' is selected, additional parameters must be set to fully define how the time-domain data is to be processed; see the description of the addpar command.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
ni2	Number of increments in 2nd indirectly detected dimension (P)
proc	Type of processing on np FID (P)

proc2d — Process 2D spectra (M)

Description A general 2D processing macro that tries to do the appropriate processing for as many types of 2D experiments as possible. It uses wft2da for phase-sensitive spectra, wft2d for absolute-value 2D spectra, wft2d('ptype') for HOM2DJ and COSYPS (absolute value). Symmetric homonuclear correlation spectra (fn=fn1, sw=sw1) in absolute-value mode is symmetrized using foldt. The resulting spectrum is then normalized (adjustment of vs and th) using nm2d and displayed (if not in background mode). proc2d is called as part of the procplot macro, but can also be used directly by the user.

See also	*NMR Spectroscopy User Guide*	
Related	fn	Fourier number in the directly detected dimension (P)
	fn1	Fourier number in 1st indirectly detected dimension (P)
	foldt	Fold COSY-like spectrum along diagonal axis (C)
	nm2d	Normalize intensity of 2D spectrum (M)
	procplot	Automatically process FIDs (M)
	sw	Spectral width in the directly detected dimension (P)
	sw1	Spectral width in the 1st indirectly detected dimension (P)
	th	Threshold (P)
	vs	Vertical scale (P)
	wft2d	Weight and Fourier transform 2D data (C)
	wft2da	Weight and Fourier transform for pure absorption 2D data (M)

`procarray` Process arrayed 1D spectra (M)

Description A generic macro for processing arrayed 1D data. It is called within the `procplot` macro, but can also be called directly. It transforms all traces, phase the trace with the largest signal, scale the traces appropriately, and set up the display parameters such that the data can be plotted directly. The plotting is done in a separate macro `plarray` that is also called in the `procplot` macro.

For the display setup, `procarray` distinguishes between arrays with 6 or less elements, which are stacked vertically (no horizontal offset), and spectra with greater than 6 elements, which are stacked horizontally by default, unless there are too many lines, in which case a diagonally stacked display is chosen.

Horizontal stacking is mostly adequate for pulse and power calibrations, where there are usually only a few lines. Diagonally stacked displays and plots are frequently chosen for T_1 and T_2 experiments on entire spectra, often with many lines. The automatic stacking mode can be overridden by creating and setting a string parameter `stackmode` in the startup macro, or before calling `procplot` or `procarray`. Possible values for `stackmode` are `'horizontal'`, `'vertical'`, and `'diagonal'`. DEPT-type spectra can, in principle, be also processed with `procarray` but, of course, no DEPT editing occurs.

See also	*NMR Spectroscopy User Guide*	
Related	deptproc	Process arrayed dept type spectra (M)
	plarray	Plot arrayed 1D spectra (M)
	proc1d	Processing macro for simple (non-arrayed) 1D spectra (M)
	procplot	Automatically process FIDs (M)

stack	Set stacking control parameter (M)
stackmode	Stack control for processing arrayed 1D spectra (P)

process — Generic automatic processing (M)

Description: Processes a wide range of data types. If the apptype parameter is set, it runs the execprocess macro if it exists. If the apptype parameter is not set it selects a macro depending on the type of data. For simple 1D spectra, process looks for a macro of form {tn}p with the observe nucleus in lower case (e.g., h1p, c13p, f19p). If no such macro is found, process calls proc1d, a generic processing macro for 1D spectra. For DEPT type data, deptproc is called. For other arrays of 1D spectra, procarray is called. For 2D spectra, proc2d is called. process by itself is called within the procplot macro.

See also: *NMR Spectroscopy User Guide*

Related:

apptype	Application type (P)
c13p	Processing of 1D carbon spectra (M)
deptproc	Process array of DEPT spectra (M)
execpars	Set up the exec parameters (M)
execprocess	Execute processing macro (P)
f19p	Processing of 1D fluorine spectra (M)
h1p	Processing of 1D proton spectra (M)
proc1d	Automatically process non-arrayed 1D fids (M)
proc2d	Process 2D spectra (M)
procarray	Process arrayed 1D spectra (M)
procplot	Automatically process FIDs (M)
tn	Nucleus for observe transmitter (P)

procplot — Automatically process FIDs (M)

Syntax: `procplot<(pltmod_value)>`

Description: Universal FID processing macro called usually with wexp='procplot' by automatic acquisition macros such as h1, c13, hcapt, and hcosy. The purpose of procplot is not the data processing itself, but rather the selection of the appropriate processing macro for a given data set.

First, procplot calls a macro process that calculates spectra; that macro by itself then selects an appropriate processing macro, like proc1d for non-arrayed 1D spectra. Depending whether the parameter pltmod is set to 'none' or not, procplot then calls plot, a universal plotting macro. The setting of the parameter pltmod can be temporarily overridden by specifying an alternative value as argument to procplot.

One of the concepts behind procplot is that the user should never have to modify any processing macro for customizing the processing or the output of automatic experiments or processing; this outcome can happen by selecting a parameter in the calling macro or before calling `procplot`.

Arguments `pltmod_value` is an alternate value for the parameter `pltmod` that is only used for the current call. The values `'none'` and `'off'` suppress plotting. The range of possible (active) values for `pltmod_value` depends on the plotting macros. Often, the parameter `pltmod` has no effect other than turning on or off plotting. Note that if only the calculation of a spectrum is desired, it is usually easier to call the `process` macro.

Examples
```
procplot
procplot('none')
```

See also *NMR Spectroscopy User Guide*

Related

`deptproc`	Process arrayed dept type spectra (M)
`plot`	Automatically plot spectra (M)
`pltmod`	Determine plot mode (P)
`proc1d`	Processing macro for simple (non-arrayed) 1D spectra (M)
`proc2d`	Process 2D spectra (M)
`procarray`	Process arrayed 1D spectra (M)
`process`	Automatically calculate spectra (M)

`profile` Set up pulse sequence for gradient calibration (M)

Applicability Systems with the pulsed field gradients (PFG) module.

Description Performs an rf and gradient echo sequence that gives a high quality profile of the sample. This sequence is used with the macro `setgcal` to provide gradient strength calibration.

See also *Performa I Pulsed Field Gradient Module Installation; Pulsed Field Gradient Modules Installation; User Programming*

Related

`gcal`	Gradient calibration constant (P)
`setgcal`	Calibrate gradient strength from measured data (M)

`profile_int` Normalise the experimental signal profile during calibration of non-uniform pulsed gradients.

Syntax `profile_int(lowfrq,highfrq)`

Applicability VnmrJ 3.1

Description Integrates the signal in the file Signal_profile, normalises it and writes it to the file Normalised_profile.

Arguments `profile_int` takes two arguments: lowfrq is the lower frequency limit of the profile, highfrq is the high frequency limit of the profile.

See also decay_gen
gradfit
nugcalib
powerfit

proj Project 2D data (C)

Syntax `proj(exp_number<,'sum'><,start<,width>>)`

Description Projects 2D data onto the axis parallel to the screen x-axis, which can be f_1 or f_2, depending upon the parameter trace. Two projections are available:

- *Summing projection*. The data at each frequency are summed and the result becomes the projection.
- *Skyline projection*. The data are searched and the maximum intensity at any given frequency becomes the intensity in the projection (similar to looking at the skyline of a city where only the largest building along any given line of sight is visible).

Phase-sensitive data can be projected, but the resulting projection can only be displayed in an absolute-value mode

Arguments `exp_number` is the number of the experiment, from 1 through 9, in which the resulting spectrum is stored.

`'sum'` is a keyword to use the summing projection. The default is skyline.

`start` defines the starting trace, in Hz. The default is to project all data.

`width` defines the width of the traces, in Hz, to be projected. The default is to project all data. If `width` is supplied as zero, a single trace corresponding to the `start` frequency will be stored.

Examples
```
proj(3)
proj(5,'sum')
proj(4,3*sfrq,6*sfrq)
```

See also *NMR Spectroscopy User Guide*

Related trace Select mode for 2D data display (P)

proshimhelp Proshim help (C)

Applicability VnmrJ 3.2

Description Use to bring up help for the Proshim window.

Proton — Set up parameters for ^{1}H experiment (M)

Description Set up parameters for ^{1}H experiment.

protune — Macro to start ProTune (M)

Applicability Liquids, Walkup, Automation

Syntax

```
protune(freq1 <, match1 <, freq2 <, match2>>>)
protune('argument',<$nucleus,<$target>>)
protune('exec', command1 <, command2, ...>)
```

Description Tunes to frequency freq1 MHz if the first argument is the frequency in MHz.

Executes a sequence of arbitrary tuning commands if the first argument is the keyword exec. Any command that can be typed into the command line box in the ProTune GUI display is allowed.

Arguments First case:

`freq1` MHz — first tuning frequency in MHz

`match1` — % of optimum for the first frequency, 5% is the default

`freq2` MHz — optional second tuning frequency in MHz

`match2` — % of optimum for the second frequency, 5% is the default.

Second case:

`'argument'` may have the following values:

no argument or `'popup'`	opens Tune Probe dialog for probe tuning. Select the nucleus to tune and how coarse to tune using the buttons and menus in the dialog box.
`'calibrate'`	open ProTune calibration interface.
`'nucleus'`	tune using specified nucleus – `$nucleus` must be specified.

`$nucleus` — Nucleus to tune to, 'H1', 'C13' ...

`$target` — Tune target level, 0.1(finest) to 100 (coarsest), defaults to 5 if no value is specified.

Third case:

`exec` — keyword that precedes a command or string of commands.

Examples `protune('exec', 'setTuneFrequency 0 599.96e6')`

Tunes the probe to 599.96 MHz.

See also User Guide Liquids and VnmrJ Walkup

Related

`atune`	ProTune present (P)
`protunegui`	Macro to start ProTune in graphical user interface (M)

plockport	Port number to use to lock out multiple ProTune processes (P)
probeConnect	Specify which nucleus can be tuned on each RF channel (P)
settune	set up tune parameters for automation
showprotunegui	show the graphical interface while tuning (P)
tchan	RF channel number used for tuning (P)
tugain	Receiver gain used in tuning (P)
tunehf	Tune both H1 and F19 on an HFX probe (M)
tunesw	Width of the tuning sweep in Hz (P)
tunematch	Default match target, in percent of optimum (P)
tupwr	Transmitter power used in tuning (P)
tuneResult	Message indicating how well the tuning succeeded (P)
tunemethod	Method to use for tuning (P)
wtune	Specify when to tune (P)
wtunedone	What to do after tuning is done (P)
xmtune	Check tune parameter during automation (M)

protune — Shell script for start ProTune operation (U)

Applicability: Automation

Description: Starts and stops ProTune. Usually called from Protune macros.

See also: *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related: protune (M — Macro to start ProTune (M)

protunegui — Macro to start ProTune in graphical user interface (M)

Applicability: Liquids, VnmrJ Walkup, Automation

Syntax: `protune('argument',<$nucleus,<$target>>)`

Description: Starts ProTune in graphical mode.

Arguments: see protune (M)

See also: *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related: protune — Macro to start ProTune (M)

prune — Prune extra parameters from current tree (C)

Syntax: `prune(file)`

Description: Destroys parameters in the current parameter tree that are not also defined in the supplied parameter file. `prune` is used to remove leftover parameters from previous experimental setups. Recalling a new parameter set into an experiment has a similar effect and, in general, `prune` is not required.

Arguments `file` is the path of a parameter file.

Examples
```
prune(systemdir+'/parlib/cosyps.par/procpar')
prune('/vnmr/par400/stdpar/H1.par/procpar')
prune(userdir+'/exp3/curpar')
```

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`destroy`	Destroy a parameter (C)
`display`	Display parameters and their attributes (C)
`fread`	Read parameters from file and load them into a tree (C)
`fsave`	Save parameters from a tree to a file (C)

pscale — Plot scale below spectrum or FID (C)

Syntax `pscale<(<rev><,axis><,label><,vp0><,sp0><,color><,pen>)>`

Description Plots a scale under a spectrum or FID.

Arguments `rev` – reverses the direction of the scale. That is, the smaller numbers will be at the left side of the scale. If used, `'rev'` must be the first argument.

`axis` – If the letter `p`, `h`, `k`, etc. is supplied, it will be used instead of the current value of the parameter `axis`. For an FID scale, if the letter `s`, `m`, or `u` is supplied, it will be used instead of the current value of the parameter `axisf`.

`label` – If a string of 2 or more characters is supplied, it will be used as the axis label.

`vp0` – This is supplied as the first real number. It defines the vertical position where the scale is drawn. The default is 5 mm below the current value of the parameter `vp`.

`sp0` – This is supplied as the second real number. It is a modified start of plot. If, for example, the display is from 347 to 447 hz, but the scale is desired to read 0 to 100 hz., `sp0` would be input as 0.

`wp0` – This is supplied as the third real number. It is a modified width of plot. If, for example, the display is from 347 to 447 hz, but the scale is desired to read 0 to 550 Units. sp0 would be input as 0, `wp0` would be 550, and the label would be 'Units'.

An optional color or pen number can be supplied to `dscale` or `pscale`. The available colors and pens are: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, `'white'`
`'pen1'`, `'pen2'`, `'pen3'`,..., `'pen8'`

Examples
```
pscale
pscale(20)
pscale('h',0,'pen2')
```

```
pscale('fid','m')
pscale('h',vp-10,0)
```

See also *NMR Spectroscopy User Guide*

Related	axis	Axis label for displays and plots (P)
	axisf	Axis label for FID displays and plots (P)
	dscale	Display scale below spectrum or FID (C)
	vp	Vertical position of spectrum (P)

pseudo   Set default parameters for pseudo-echo weighting (M)

Syntax `pseudo<(C1,C2,C3,C4)>`

Description Generates an initial guess at good weighting parameters for absolute-value 2D experiments. To generate modified guesses, four coefficients are allowed to set the values of the weighting functions.

Arguments `C1` sets `lb=-0.318/(C1*at)`. The default value of `C1` is 0.0625.

`C2` sets `gf=C2*at`. The default value of `C2` is 0.25.

`C3` sets `lb1=-0.318/(C3*(ni/sw1))` but is used with 2D experiments only. The default value of `C3` is 0.0625.

`C4` sets `gf1=C4*(ni/sw1)` but is used with 2D experiments only. The default value of `C4` is 0.25.

Examples
```
pseudo
pseudo(.1,.4,.2,.5)
```

See also *NMR Spectroscopy User Guide*

Related	sinebell	Select default parameters for sinebell weighting (M)

psg   Display pulse sequence generation errors (M)

Description Helps identify the problem if, after entering `go` or `su`, etc., the message is returned that pulse sequence generation (PSG) aborted abnormally. Any parameters that are not found are listed. This information is stored in the user's directory (`vnmrsys`) in a text file named `psg.error`. If the message “Maximum communication retries exceeded, Experiment unable to be sent” is displayed, a program communications problem is indicated. Consult the system operator for assistance.

See also *User Programming*

psggen   Compile a user PSG object library (M,U)

Description A user PSG (pulse sequence generation) kit is supplied that allows editing low-level pulse sequence code. `psggen` compiles these edits so that subsequent pulse sequence generation with the seqgen command uses the customized pulse sequence source.

See also *User Programming*

psgset Set up parameters for various pulse sequences (M)

Syntax `psgset(file,par1,par2,...,parN)`

Description Sets up parameters for various pulse sequences using information in a `parlib` file. Rather than returning the entire parameter file, `psgset` returns the parameters listed. `psgset`, in general, is never entered from the keyboard but is used as part of experiment setup macros.

Arguments `file` is the file from the user or system `parlib` that provides information on setting up the parameters listed. The parameters `seqfil` and `pslabel` are set to the supplied file name.

`par1,par2,...,pN` are 1 to 11 parameters to be returned from `parlib`.

Examples `psgset('cosy','dg','ap','ss','d1','axis','phase')`

See also *User Programming*

Related
- `pslabel` Pulse sequence label (P)
- `seqfil` Pulse sequence name (P)

psgupdateon Enable update of acquisition parameters (C)

Description Permits the interactive updating of acquisition parameters.

See also *SpinCAD*

Related
- `psgupdateoff` Prevent update of acquisition parameters (C)
- `updtparam` Update specified acquisition parameters (C)

psgupdateoff Prevent update of acquisition parameters (C)

Description Prevents the interactive updating of acquisition parameters.

See also *SpinCAD*

Related
- `psgupdateon` Enable update of acquisition parameters (C)
- `updtparam` Update specified acquisition parameters (C)

pshape — Plot pulse shape or modulation pattern (M)

Syntax `pshape<(pattern.ext)>`

Description Plots the real (X) and imaginary (Y) components of a shaped pulse. Any type of waveform (.RF, .DEC or ,GRD) can be plotted.

Arguments `pattern` is the name of a shape or pattern file specified by an absolute file name, relative file name, or a simple pattern file name. `ext` is a file name extension that specifies the file type. In the case of a simple file name, `dshape` searches for the file in the local directory, then in the user's `shapelib`, and finally in the directory `/vnmr/shapelib`. If `pattern.ext` is not given, `pshape` displays the last created waveform stored in the `pbox.fid` file.

Examples
```
pshape
pshape('my_shape.DEC')
```

See also *NMR Spectroscopy User Guide*

Related

`dshape`	Display the last created pulse shape (M)
`Pbox`	Pulse shaping software (U)

pshapef — Plot the last created pulse shape (M)

Description Plots real (X) and imaginary (Y) components of the last created shaped pulse.

See also *NMR Spectroscopy User Guide*

Related

`dshape`	Display the last created pulse shape (M)
`Pbox`	Pulse shaping software (U)

pshr — PostScript High Resolution plotting control (P)

Applicability ALL

Syntax `pshr=<value>`

Description Global parameter that controls whether a 1D spectrum is plotted in hi-resolution mode or not. A hi-resolution plot is one in which every data point is represented in the plot. The standard resolution plot determines maximum and minimum values over small regions and plots those. The parameter pshr can have the values 1 for hi-res and 0 for standard plot.

Values `0` for standard resolution

`1` for high resolution.

Related

`pl`	Plot spectra (C)
`pslw`	PostScript Line Width control (P)

pslabel — Pulse sequence label (P)

Description: Contains the text to be displayed in the `Seq:` field on the top line of the screen. This string may be different from the pulse sequence name selected with `seqfil`. However, the string in `seqfil` is the name of the pulse sequence searched for when an experiment is started. Generally `seqfil=pslabel`, and when `seqfil` is set, the system sets `pslabel` to the same string.

See also: *NMR Spectroscopy User Guide*

Related: `seqfil` — Pulse sequence name (P)

<pslabel>_setup — Experiment-Specific Setup Macro (M)

Syntax:

Applicability: VnmrJ 3.1

Description: Macro is executed to set up sequence-specific parameters.

Examples: User Guide: Automation-User Space Customization

Related: `execpslabel('setup')` — Pulse sequence name (P)

pslw — PostScript Line Width control (P)

Applicability: ALL

Syntax: `pslw=<value>`

Description: Global parameter that adjusts the line width of PostScript plots.

Values: `0` (narrowest) to `100` (widest) line width.

Related:
`pl` — Plot spectra (C)
`pshr` — PostScript High Resolution plotting control (P)

psMain — Prescan controlling macro

Syntax

Applicability: VnmrJ 3.1

Description: Prior to acquiring data, a number of operations may be performed to condition the data acquisition. These may include probe tuning, acquiring a lock, shimming, adjusting receiver gain, and performing an equilibration delay. These operations are collectively referred to as prescan operations. The order of executing the various prescans, and the name of the macro to call for a specific prescan, is defined in the "templates/vnmrj/choicefiles" application directory by the prescanInfo file.

The `psMain` macro is the controlling macro that executes each prescan. The individual prescans are controlled by macros, conventionally named `psX`, where X is Gain, Lock, etc.

Arguments Calling the `psMain` macro with no arguments will execute all defined prescans, in the order given in the prescanInfo file. Calling psMain with the name of a specific prescan, or a list of specific prescans, will execute those. For example, psMain('psGain') will execute the autogain prescan. psMain('psTune psLock') will tune the probe and then autolock.

The prescan process can also be executed in steps. psMain('setup') initializes the prescans, but does not start the process. At this point, the setup may be customized. For example, a specific prescan could be removed from the list with the command psCmd('remove','psTune'). The command `psMain('start'):$ret` starts the execution. Depending an what specific prescans are requested, a data acquisition may or may not be started. Depending on whether the prescans start an acquisition or not, the $ret value will be set to 'psAcquiring' or 'psDone', respectively. In the case of 'psAcquiring', you can schedule the post-prescan acquisition with psMain('acquireAfterPs'). See the `cpgo` macro for an example.

`pssl` Plot Arrayed Numbers (C)

Syntax `pssl(<options>)`

Description Plots a label for each element in a set of stacked spectra. The label is an integer value from 1 up to the number of spectra in the display.

Arguments `options` can be any of the following:

- `'all'` is a keyword to display all of the spectra.
- `'int'` is a keyword to display only the integral, independently of the value of the parameter `intmod`
- `'top'` or `'side'` are keywords that cause the spectrum to be displayed either above or at the left edge, respectively, of a contour plot. This assumes that the parameters `sc`, `wc`, `sc2`, and `wc2` are those used to position the contour plot.
- `'dodc'` is a keyword for all spectra to be drift corrected independently.
- `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'black'`, and `'white'` are keywords that select a color.
- `'pen1'`, `'pen2'`, `'pen2'` ... are keywords that pens.
- `'nopars'` — prevents the display commands from drawing the parameters at the bottom of the graphics screen.
- `'custom'` — uses the parameters `shownumx` (x position) and `shownumy` (y position), counting from bottom left of every spectrum.

- 'reverse' — rotate the text by 90° - useful if the arrayed parameter values are long with respect to the width of the individual sub-spectra.
- 'value' —The values of up to two simultaneous arrays are displayed. Diagonal arrays are allowed. The second parameter is shown in different color). The name of the arrayed parameter(s) is also shown. If used on a one-dimensional array representation of a 2D spectrum, ni and phase (in case of phase sensitive 2Ds) parameters are shown.
- 'list=xxx' produces a display of the values contained in the arrayed parameter xxx.
- 'format=yyy' uses the format yyy to control the plot of each label. See the write command for information about formats.

Examples
```
pssl
pssl('top','left')
pssl('value','format=%3.1f')
```

See also *NMR Spectroscopy User Guide*

Related

dssl	Label a display of stacked spectra (M)
write	Write formatted text to a device (C)

ptext — Print out a text file (M)

Syntax ptext(file)

Description Prints out a text file.

Arguments file is the name of the text file.

Examples
```
ptext('/vnmr/maclib/ptext')
ptext(curexp+'/dept.out')
```

See also *NMR Spectroscopy User Guide*

Related

curexp	Current experiment directory (P)
dtext	Display a text file in the graphics window (C)
lookup	Look up words and lines from a text file (C)
pltext	Plot a text file (C)
text	Display text or set new text for current experiment (C)
textvi	Edit text file of current experiment (M)
vi	Edit text file with vi text editor (C)

ptspec3d — Region-selective 3D processing (P)

Description Sets whether region-selective 3D processing occurs. If ptspec3d does not exist, it is created by the macro par3d. ptspec3d is functional at this time only for the f_3 dimension. If ptspec3d='ynn', only the currently displayed region of f_3 is retained as non-zero values after the f_3 transform in the 3D FT. A larger f_3 region may be kept to ensure that the number of hypercomplex f_3 points is a power of 2; but that portion of the f_3 spectrum that is retained outside of the currently displayed region contains only zeroes. This 3D utility can reduce the fully transformed 3D data size by factors of 2 to 4, especially in some of the triple resonance experiments.

Values A three-character string such as `'nnn'`, `'nny'`, `'nyn'`, etc. The default is `'nnn'`. The first character refers to the f_3 dimension (`sw`, `np`, `fn`); the second character, to the f_1 dimension (`sw1`, `ni`, `fn1`); and the third character, to the f_2 dimension (`sw2`, `ni2`, `fn2`). Each character may take one of two values: `'n'` for no region-selective processing in the relevant dimension, or `'y'` for region-selective processing in the relevant dimension.

See also *NMR Spectroscopy User Guide*

Related		
	`fiddc3d`	3D time-domain dc correction (P)
	`fn`	Fourier number in directly detected dimension (P)
	`fn1`	Fourier number in 1st indirectly detected dimension (P)
	`fn2`	Fourier number in 2nd indirectly detected dimension (P)
	`ft3d`	Perform a 3D Fourier transform (M)
	`ni`	Number of increments in 1st indirectly detected dimension (P)
	`ni2`	Number of increments in 2nd indirectly detected dimension (P)
	`np`	Number of data points (P)
	`ntype3d`	N-type peak selection in f_1 or f_2 (P)
	`par3d`	Create 3D acquisition, processing, display parameters (C)
	`specdc3d`	3D spectral drift correction (P)
	`sw`	Spectral width in directly detected dimension (P)
	`sw1`	Spectral width in 1st indirectly detected dimension (P)
	`sw2`	Spectral width in 2nd indirectly detected dimension (P)

`ptsval` PTS frequency synthesizer value (P)

Description Configuration parameter for the frequency of the PTS synthesizer on each channel. Every broadband system is equipped with a PTS frequency synthesizer as part of broadband frequency generation. The frequency of the unit is marked on its front panel. The value is set for each channel using the Synthesizer label in the Spectrometer Configuration window.

Values `0` (Not Present choice in Spectrometer Configuration window); 160, 200, 250, 320, 500, 620, 1000 (PTS 160, PTS 200, PTS 250, PTS 320, PTS 500, PTS 620, PTS 1000 choices in Spectrometer Configuration window, respectively).

See also *VnmrJ Installation and Administration.*

Related

`config`	Display current configuration and possibly change it (M)
`latch`	Frequency synthesizer latching (P)
`overrange`	Frequency synthesizer overrange (P)

`pulseinfo` Shaped pulse information for calibration (M)

Syntax `pulseinfo<(shape,pulse_width<,reference_power>)>`
`:width,power`

Description Returns or prints a table with the bandwidth and predicted pulse power settings for a given pulse shape. No parameter settings are changed. The necessary data is contained in the file `shapeinfo` in the system `shapelib` subdirectory.

Arguments `shape` is the name of the pulse shape. The default is the system interactively prompts the operator for the name of the shape and the duration of the pulse and then prints a table containing the bandwidth of that pulse and the predicted pulse power settings.

`pulse_width` is the duration of the pulse, in µs.

`reference_power` is a value, in dB, for power calculations. The default is 55. This value replaces the assumption used for power calculation that `pw90` is set for a `tpwr` of 55.

`width` returns the bandwidth of that pulse, in Hz.

`power` returns the predicted 90° pulse power settings.

Examples `pulseinfo('gauss',1000):bw,pwr`

See also *User Programming*

Related

`bandinfo`	Shaped pulse information for calibration (M)
`pw90`	90° pulse width (P)
`tpwr`	Observe transmitter power level with linear amplifiers (P)

`pulsetool` RF pulse shape analysis (U)

Syntax `pulsetool <-shape filepath>`

Description Enables examination of shaped rf pulses. It is started from a UNIX window.

Arguments The optional `-shape filepath` specifies the name of an rf pulse template file that is displayed when `pulsetool` is started.

Examples
```
pulsetool
pulsetool -shape /vnmr/shapelib/sinc.RF
```

See also *NMR Spectroscopy User Guide*

purge — Remove macro from memory (C)

Syntax `purge<(file)>`

Description Removes one or more macros from memory, freeing extra memory space.

Arguments `file` is the name of a macro file to be removed from memory. The default is to remove all macros that have been loaded into memory.

CAUTION The purge command with no arguments should never be called from a macro. The `purge` command with an argument should never be called by the macro being purged.

Examples `purge`
`purge('_sw')`

See also *User Programming*

Related `macrold` Load a macro into memory (C)

puttxt — Put text file into a data file (C)

Syntax `puttxt(file)`

Description Copies text from current experiment into a data file.

Arguments `file` is the name of a data file (i.e., a directory with a `.fid` or `.par` suffix). Do not include the suffix in the name provided to `file`.

Examples `puttxt('mydata')`

See also *NMR Spectroscopy User Guide*

Related `gettxt` Get text file from another file (C)

putwave — Write a wave into Pbox.inp file (M)

Syntax `putwave(sh,bw,pw,ofs,st,ph,fla,trev,d1,d2,d0)`

Description Sets up a single excitation band in the `Pbox.inp` file. An unlimited number of waves can be combined by reapplying `putwave`.

Arguments 1 to 11 wave parameters in the following predefined order:

`sh` is the name of a shape file.

`bw` is the bandwidth, in Hz.

`pw` is the pulsewidth, in sec.

`ofs` is the offset, in Hz.

`st` is a number specifying the spin status: 0 for Mz, or 1 for Mxy.

`ph` is the phase (or phase cycle, see `wavelib/supercycles`).

`fla` is the flip angle. Note that `fla` can override the default flip angle.

`trev` concerns time reversal. It can be used to cancel time reversal if spin status (`st`) is set to 1 for Mxy.

`d1` is the delay, in sec, prior the pulse.

`d2` is the delay, in sec, after the pulse.

`d0` is a delay or command prior to `d1`. If `d0=a`, the wave is appended to the previous wave.

Examples
```
putwave('eburp1')
putwave('GARP',12000.0)
putwave('esnob',600,-1248.2,1,90.0,'n','n',0.001)
```

See also *NMR Spectroscopy User Guide*

Related		
	Pbox	Pulse shaping software (U)
	setwave	Write a wave definition string into the Pbox.inp file (M)

pw — Enter pulse width pw in degrees (C)

Syntax `pw(flip_angle,<90_pulse_width>)`

Description Calculates the flip tim, in µs, given a desired flip angle and 90° pulse. The value is entered into the parameter `pw`.

Arguments `flip_angle` is the desired flip angle, in degrees.

`90_pulse_width` is the 90° pulse length, in µs. The default is the value of parameter `pw90`, if it exists.

Examples
```
pw(30)
pw(90,12.8)
```

See also *NMR Spectroscopy User Guide*

Related		
	ernst	Calculate the Ernst angle pulse (C)
	pw	Pulse width (P)
	pw90	90° pulse width (P)

pw — Pulse width (P)

Description Length of the final pulse in the standard two-pulse sequence. In “normal” 1D experiments with a single pulse per transient, this length is the observe pulse width.

Values 0, 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also *NMR Spectroscopy User Guide*

Related		
	p1	First pulse width (P)
	pw	Enter pulse width parameter `pw` in degrees (C)

pw90 — 90° pulse width (P)

Description: Length of the 90° pulse. `pw90` is not used by pulse sequences directly, but is used by a number of commands to assist in setting up special experiments. `pw90` is also used by certain output programs to be able to print the value of the pulse width in degrees instead of microseconds. Note that this parameter must be updated by the user and is not automatically determined or magically correct under all circumstances.

Values: 0, 0.1 µs to 8190 sec, smallest value possible is 0.1 µs, finest increment possible is 12.5 ns.

See also: *NMR Spectroscopy User Guide*

Related:

`AC1S-AC11S`	Autocalibration macros (M)
`pw`	Enter pulse width parameter `pw` in degrees (C)

pwd — Display current working directory (C)

Syntax: `pwd<:directory>`

Description: Displays the path of the current working directory.

Arguments: `directory` is a string variable with the path of the current directory.

Examples: `pwd:$name`

See also: *NMR Spectroscopy User Guide*

Related:

`cd`	Change working directory (C)
`dir`	List files in current directory (C)
`lf`	List files in current directory (C)
`ls`	List files in current directory (C)

pwpat — Shape of refocusing pulse (P)

Applicability: Systems with imaging capabilities.

Description: Specifies the shape of the refocusing pulse `pw` in imaging experiments

Values: `'hard'`, `'sinc'`, `'gauss'`, `'sech'`, `'sine'`, or any shape resident in the system pulse shape library or libraries.

See also: *VnmrJ Imaging NMR*

Related:

`p1pat`	Shape of an excitation pulse (P)
`pw`	Pulse width (P)

pwr Set power mode in directly detected dimension (C)

Description Selects the power spectra display mode by setting dmg='pwr'. In the *power mode,* each real point in the displayed spectrum is calculated as the sum of the squares of the real and imaginary points comprising each respective complex data point. All information, including noise, is positive and the relationship between signal and noise is non-linear.

For multidimensional data, pwr has no effect on data prior to the second Fourier transform. If pmode='full', pwr acts in concert with the commands ph1, av1 or pwr1 to yield the resultant contour display for the 2D data.

See also *NMR Spectroscopy User Guide*

Related

av	Set abs. value mode in directly detected dimension (C)
av1	Set abs. value mode in 1st indirectly detected dimension (C)
dmg	Data display mode in directly detected dimension (P)
ft	Fourier transform 1D data (C)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
pa	Set phase angle mode in directly detected dimension (C)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
ph	Set phased mode in directly detected dimension (C)
ph1	Set phased mode in 1st indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr1	Set power mode in 1st indirectly detected dimension (C)
pwr2	Set power mode in 2nd indirectly detected dimension (C)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 of 2D data (M)
wft2d	Weight and Fourier transform 2D data (M)

pwr1 Set power mode in 1st indirectly detected dimension (C)

Description Selects the power spectra display mode along the first indirectly detected dimension by setting dmg1='pwr1'. If the parameter dmg1 does not exist, pwr1 creates it and sets it to 'pwr1'. In the *power mode*, each real point in the displayed trace is calculated as the sum of the squares of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the real-real and imaginary-real points from each respective hypercomplex data point are used in the summation. In this mode, all information, including noise, is positive and the relationship between signal and noise is non-linear.

The pwr1 command is only needed if mixed-mode display is desired. If the parameter dmg1 does not exist or is set to the null string, the display mode along the first indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter dmg). For the contour display of multidimensional data, the result of pwr1 is the same as for traces, provided that pmode='partial' or pmode=''.

See also *NMR Spectroscopy User Guide*

Related

dmg1	Data display mode in 1st indirectly detected dimension (P)
pa	Set phase angle mode in directly detected dimension (C)
pa1	Set phase angle mode in 1st indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr	Set power mode in directly detected dimension (C)
pwr2	Set power mode in 2nd indirectly detected dimension (C)

pwr2 Set power mode in 2nd indirectly detected dimension (C)

Description Selects the power spectra display mode along the second indirectly detected dimension by setting dmg2='pwr2'. If dmg2 does not exist or is set to the null string, pwr2 will create dmg2 and set it equal to 'pwr2'. In the *power mode*, all information, including noise, is positive and the relationship between signal and noise is non-linear. Each real point in the displayed trace is calculated as the sum of the squares of the real and imaginary points comprising each respective complex data point. For hypercomplex data, the real-real and imaginary-real points from each respective hypercomplex data point are used in the summation.

The pwr2 command is only needed if mixed-mode display is desired. If the parameter dmg2 does not exist or is set to the null string, the display mode along the second indirectly detected dimension defaults to the display mode of the directly detected dimension (characterized by the parameter dmg). For the contour display of multidimensional data, the result of pwr2 is the same as for traces, provided that pmode='partial' or pmode=''.

See also *NMR Spectroscopy User Guide*

Related

av2	Set abs. value mode in 2nd indirectly detected dimension (C)
dmg2	Data display mode in 2nd indirectly detected dimension (P)
ft1d	Fourier transform along f_2 dimension (C)
ft2d	Fourier transform 2D data (C)
ph2	Set phased mode in 2nd indirectly detected dimension (C)
pmode	Processing mode for 2D data (P)
pwr	Set power mode in directly detected dimension (C)

pwsadj Adjust pulse interval time (M)

Applicability Systems with waveform generators.

Syntax `pwsadj(shape_file,pulse_parameter)`

Description Adjusts the pulse interval time so that the pulse interval for the specified shape is an integral multiple of 100 ns. This ensures there is no time truncation error in executing the shaped pulse by waveform generators.

Arguments `shape_file` is a file name of a shaped pulse file. The name can be specified with or without the `.RF` file extension. `pwsadj` first looks for the file name specified by `shape_file` in the user's `shapelib` directory. If the file specified is not found there, `pwsadj` then looks in the system `shapelib` directory.

`pulse_parameter` is a string containing the adjusted pulse interval time.

Examples `pwsadj('pulse12','pulseparam')`

See also *User Programming*

Related		
	dmfadj	Adjust decoupler tip-angle resolution time (M)
	dmf2adj	Adjust second decoupler tip-angle resolution time (M)

pwxcal Decoupler pulse calibration (M)

Description Provides an interactive method of selecting the decoupler (first, second, or third) and the nucleus (^{13}C, ^{15}N, or ^{31}P) to calibrate. The `pwxcal` pulse sequence determines the pulse width characteristics of the probe's decoupler channel(s) in indirect detection or triple resonance experiments. `pwxcal` can also be used to determine the rf field homogeneity of the decoupler.

The parameter `pwx1` is arrayed to calibrate the 90° pulse width on the first decoupler. If a second decoupler is present, the parameter `pwx2` is arrayed to calibrate the 90° pulse width on that decoupler. If a third decoupler is present, the parameter `pwx3` is arrayed to calibrate the 90° pulse width on that decoupler. Other parameters include: `jC13` is the ^{13}C-^{1}H coupling, constant, `jN15` is the ^{15}N-^{1}H coupling constant, `jP31` is the ^{31}P-^{1}H coupling constant, and `jname` is a selected calibration nucleus.

See also *System Administration*

pxbss — Bloch-Siegert shift correction during Pbox pulse generation (P)

Description: A flag to enable or disable Bloch-Siegert shift correction during the creation of Pbox pulses.

Values: 'y' enable Bloch-Siegert shift correction
'n' disable Bloch-Siegert shift correction
Default value is 'y'.

See also: *NMR Spectroscopy User Guide*

Related: `htfrq1` Hadamard frequency list in `ni` (P)

pxrep — Flag to set the level of Pbox reports (P)

Description: A flag to set the level of Pbox debug messages displayed at the start of acquisition.

Values: 'y' shows all Pbox reports.
'h' shows the Hadamard matrix.
'n' shows no reports.
Default value is 'nnn'.

See also: *NMR Spectroscopy User Guide*

Related: `htfrq1` Hadamard frequency list in `ni` (P)

pxset — Assign Pbox calibration data to experimental parameters (M)

Syntax: `pxset<(file.ext)>`

Description: Retrieves experimental settings from a file and assigns them to corresponding experimental parameters using a dialog form. If no file name is provided, `pxset` extracts data from the `Pbox.cal` file that contains the output data of the last created waveform

Arguments: `file.ext` is the name of a shape or pattern file.

Examples:
```
pxset
pxset('Pbox.RF')
```

See also: *NMR Spectroscopy User Guide*

Related:
`Pbox` Pulse shaping software (U)
`pboxget` Extract Pbox calibration data (M)

pxshape Generates a single-band shape file (M)

Syntax
```
pxshape('sh bw/pw ofs st ph fla trev   \
   d1 d2 d0',name,disp)
```

Description Generates a single-band waveform based on wave definition provided as a single string of wave parameters.

Arguments A single string of 1 to 12 wave parameters in predefined order. Note that a single quote is required at the start and the end of the entire string, but no single quotes are required surrounding characters and strings inside the entire string.

`sh` is the name of a shape file.

`bw/pw` is either the bandwidth, in Hz, or the pulsewidth, in sec.

`ofs` is the offset, in Hz.

`st` is a number specifying the spin status: 0 for Mz, or 1 for Mxy.

`ph` is the phase (or phase cycle, see `wavelib/supercycles`).

`fla` is the flip angle. Note that `fla` can override the default flip angle.

`trev` is a time reversal. This can be used to cancel time reversal if spin status (`st`) is set to 1 for Mxy.

`d1` is the delay, in sec, prior the pulse.

`d2` is the delay, in sec, after the pulse.

`d0` is a delay or command prior to `d1`. If `d0=a`, the wave is appended to the previous wave.

`name` is the output file name. An extension is optional and can be used to override an internally defined shape type.

`disp` is the shape is displayed by default in the graphics window. If `disp` is set to `'n'`, the shape is not displayed.

Examples
```
pxshape('eburp1','myshape.RF')
pxshape('GARP 12000.0','shape2','y')
pxshape('esnob 600.0 -1248.2 n 180.0 n n 0.001','xxx')
```

See also *NMR Spectroscopy User Guide*

Related		
	`Pbox`	Pulse shaping software (U)

Pxsim Simulate Bloch profile for a shaped pulse (U)

Syntax
```
Pxsim file <simtime <num_steps <add/sub>>>
```

Description Used by the `dprofile` macro to simulate a Bloch profile for a shaped pulse. `Pxsim` extracts the information necessary for simulation from the shape header. Only shape files containing this information can be processed.

Arguments `file` is the name of a shape or pattern file including an .RF or .DEC extension. `Pxsim` searches for the file in the user's `shapelib`

(~/vnmrsys/shapelib), and if not found there, it searches in the system shapelib (vnmr/shapelib).

simtime is the maximum simulation time (in sec) that can be provided.

num_steps is the number of steps in the profile.

add/sub is add (a) or subtract (s) from the previous simulation.

Examples Pxsim myshape.RF

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

Pxspy Create shape definition using Fourier coefficients (U)

Syntax Pxspy file

Description An interactive program that converts shaped pulse files into a Fourier series and produces an output file pbox.cf in the user's shapelib (~/vnmrsys/shapelib), which can be used to create a wave definition file in the wavelib directory. Pxspy can also be used to convert hard pulse decoupling sequences into soft (“cool”) decoupling waveforms. The resulting Fourier coefficients can depend on the number of points in the waveform.

Arguments file is the name of a shape or pattern file, including an .RF, .DEC, or .GRD extension. The name can be given as a relative name, absolute name, or as a simple name (i.e., with a path). If given as a simple name, Pxspy searches for the file in the user’s shapelib (~/vnmrsys/shapelib), and then if not found there, it searches in the system shapelib (vnmr/shapelib).

Examples
```
Pxspy myshape.RF
Pxspy /vnmr/shapelib/myshape.RF
Pxspy ~vnmrsys/shapelib/myshape.RF
```

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

<pslabel>_plot Experiment-Specific Plot Macro

Description The <pslabel>_plot macro, if it exists, is executed at set-up and is used to configure plotting and display features on a pslabel-specific basis.

Related pl_<pslabel>

<pslabel>_process Experiment-Specific Processing Macro

Description The <pslabel>_process macro, if it exists, is executed at set-up and is used to configure processing parameters on a pslabel-specific basis.

<pslabel>_setup Experiment-Specific Setup Macro

Description The <pslabel>_setup macro is executed to set up sequence-specific parameters.

Related cpsetup

Q

qcomp — Longer dead time for longer ring down (P)

Applicability Systems with Varian, Inc. Cold Probes

Description Global parameter to handle longer ring down times following the rf pulse. This is only active if `dsp='i'` or if `dsp='r'` and `fsq='y'`. The dead time is calculated by the software and the DSP parameters are appropriately adjusted for flat baseline and good phase properties. If it is necessary to use a user specified delay, create the `prealfa` parameter. `qcomp` is not effective in explicit acquisition experiments. Not compatible with `srof2`.

Values `qcomp='y'` triggers a longer dead time before the receiver is gated on for the acquisition.

Related

prealfa	Specify a delay for longer ring down (P)
dsp	Type of DSP for data acquisition (P)

QKexp — Set up quick experiment (M)

Syntax `QKexp(arguments)`

Description Set up parameters for quick experiment for a chained acquisition. Multiple arguments can be given to define the chain. Default parameter values are used by the macro and or the probe file is used.

Examples QKexp('PROTON','COSY','HMQC')
QKexp('PROTON','CARBON','HETCOR','gCOSY')

qtune Tune probe using swept-tune graphical tool (C)

Syntax qtune<(gain<,power>)>

Description Displays a real-time graph showing reflected power versus frequency for tuning probes. If the acquisition system has been recently rebooted, enter su before running qtune. Refer to the manual *NMR Spectroscopy User Guide* for a detailed description of this tool.

Arguments gain specifies the gain value, typically 20 to 50. The default is 50.
power specifies the power value, typically 60 to 70. The default is 60.

Examples qtune
qtune(20)
qtune(38,65)

See also *NMR Spectroscopy User Guide*

Related

tugain	Amount of receiver gain used by qtune (P)
su	Submit a setup experiment to acquisition (M)
tune	Assign frequencies (C)

? Display the value of an individual parameter (C)

Syntax parameter_name<[index]>?

Description The question mark displays the current numerical or string value of a parameter when the parameter name is followed by a question mark. No change is made to the value of the parameter. To display an individual element of an parameter array, provide the index in square brackets (e.g., nt[3]? might display "nt[3]=2")

Certain parameters can be "turned off" by setting the parameter to 'n'. The display of a parameter that is turned off will be the phrase "Not Used" followed by the actual value in parentheses. For example, if lb is set to 1.5 and then set to 'n', entering lb? will display lb= Not Used (1.5). Such a parameter can be "turned on" by setting it to 'y'. It will then have its prior value.

To show a parameter's array of values or learn about its attributes, use the display command.

Arguments index is the integer for a selected member of an arrayed parameter.

Examples lb?
sw?
pw[2]?

See also *NMR Spectroscopy User Guide*

Related

display	Display parameters and their attributes (C)
getvalue	Get value of a parameter in a tree (C)

quadtt Prints differences in wideline receiver channels

Syntax `quadtt`

Applicability VnmrJ 3.1

Description Prints differences in wideline receiver channels.

See also See Wideline Accessory Installation Manual Pub. No. 87-178257-00 Rev B788 or later. Used with pulse sequence `s2pulq`.

quantcalhelp Quantification calibration help file

R

r	Recall display parameter set (M)
r(n)	Recall some display parameters (C)
r1-r7	Real-value storage for macros (P)
ra	Resume acquisition stopped with sa command (C)
random	Return a random number
rcvrwt	Weighting for different receivers (P)
react	Recover from error conditions during werr processing (M)
readallshims	Read all shims from hardware (M)
readbrutape	Read Bruker data files from 9-track tape (U)
readfile	Read the contents of a text file into two parameters (C)
readhw	Read current values of acquisition hardware (C)
readlk	Read current lock level (C)
readparam	Read one of more parameters from a file (C)
readultra	Read shim coil setting for Ultra•nmr shim system (M)
real	Create a real variable without a value (C)
recon_all	Reconstruct images from 2D MRI fid data (C)
record	Record keyboard entries as a macro (M)
redor1	Set up parameters for REDOR1 pulse sequence (M)
redosy	Restore 2D DOSY display from sub experiment (M)
reff1	Reference f2 Indirect Dimension from Observe Dimension (M)
reff2	Reference f2 Indirect Dimension from Observe Dimension (M)
reffrq	Reference frequency of reference line (P)
reffrq1	Reference freq. of reference line in 1st indirect dimension (P)
reffrq2	Reference freq. of reference line in 2nd indirect dimension (P)
refpos	Position of reference frequency (P)

refpos1	Position of reference frequency in 1st indirect dimension (P)
refpos2	Position of reference frequency in 2nd indirect dimension (P)
refsource1	Center frequency in 1st indirect dimension (P)
refsource2	Center frequency in 2nd indirect dimension (P)
region	Divide spectrum into regions (C)
relayh	Set up parameters for RELAYH pulse sequence (M)
rename	Move and/or rename a file (C)
reorder3D	Reorders array elements in arrayed phase sensitive 2D experiment (M)
reqparcheck	Flag which enables/disables required parameters (P)
reqparclear	Clears the parameters in required parameter list (M)
reqparlist	List of required parameters (P)
reqpartest	Tests whether required parameters are set (M)
res	Display lineshape of largest peak in the current spectrum (M)
resetf3	Reset parameters after a partial 3D Fourier transform (M)
resetplotter	Reset plotter to system plotter (M)
resetsampglobal	Clears sample global parameters
resolv	Set resolution enhancement parameters (M)
restorenuctable	Calculate and (Re-)store accurate nuctable (M)
resume	Resume paused acquisition queue (C)
return	Terminate execution of a macro (C)
rev	System software revision level (P)
revdate	System software preparation date (P)
rfband	RF band in use (P)
rfblk	Reverse FID block (C)
rfchannel	Independent control of rf channel selection (P)
rfchtype	Type of rf channel (P)
rfdata	Reverse FID data (C)
rfl	Reference peak position in directly detected dimension (P)
rfl1	Reference peak position in 1st indirectly detected dimension (P)
rfl2	Reference peak position in 2nd indirectly detected dimension (P)
rfp	Reference peak frequency in directly detected dimension (P)
rfp1	Reference peak freq. in 1st indirectly detected dimension (P)

rfp2	Reference peak freq. in 2nd indirectly detected dimension (P)
rftempcomp	RF Transmitter Board Temperature Compensation (P)
rftrace	Reverse FID trace (C)
rftype	Type of rf generation (P)
rfwg	RF waveform generator (P)
right	Set display limits to right half of screen (C)
rights	Determine an operator's specified right (C)
rinput	Input data for a regression analysis (M)
rl	Set reference line in directly detected dimension (M)
rl1	Set reference line in 1st indirectly detected dimension (M)
rl2	Set reference line in 2nd indirectly detected dimension (M)
rm	Delete file (C)
rmdir	Remove directory (C)
rmsAddData	Add transformed data files with weighting (U)
Roesy	Convert the parameter to a ROESY experiment (M)
Roesy1d	Convert the parameter set to a Roesy1d experiment (M)
rof1	Receiver gating time preceding pulse (P)
rof2	Receiver gating time following pulse (P)
rof3	Receiver gating time following T/R switch (P)
rotate	Rotate 2D data (C)
rotorsync	Rotor synchronization (P)
rp	Zero-order phase in directly detected dimension (P)
rp1	Zero-order phase in 1st indirectly detected dimension (P)
rp2	Zero-order phase in 2nd indirectly detected dimension (P)
rt	Retrieve FIDs (M)
rtcmx	Return Spinsight data into current experiment (C)
rtp	Retrieve parameters (M)
rts	Retrieve shim coil settings (C)
rttmp	Retrieve experiment data from experiment subfile (M)
rtv	Retrieve individual parameters (C)
rtx	Retrieve parameters based on rtx rules (C)

r — Recall display parameter set (M)

Syntax (1) `rset_number`
(2) `r(set_number)`

Description Recalls the parameters sp, wp, sp1, wp1, sp2, wp2, sc, wc, sc2, wc2, ho, vo, vs, and ai/nm of a selected display parameter set. Not recalled are phase parameters, drift correction parameters, integral reset parameters, and reference parameters. This allows, for example, saving a set of display parameters, adjusting the phase or drift correction, and later recalling the display parameters without und0oing the new phase or drift correction.

Arguments `set_number` is the number, from 1 to 9, of a display parameter set.

Examples `r2`
`r(3)`

See also *NMR Spectroscopy User Guide*

Related

ai	Select absolute intensity mode (C)
fr	Full recall of a display parameter set (M)
ho	Horizontal offset (P)
nm	Select normalized intensity mode (C)
s	Save display parameters as a set (M)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
sp	Start of plot in directly detected dimension (P)
sp1	Start of plot in 1st indirectly detected dimension (P)
sp2	Start of plot in 2nd indirectly detected dimension (P)
vo	Vertical offset (P)
vs	Vertical scale (P)
wc	Width of chart (P)
wc2	Width of chart in second direction (P)
wp	Width of plot in directly detected dimension (P)
wp1	Width of plot in 1st indirectly detected dimension (P)
wp2	Width of plot in 2nd indirectly detected dimension (P)

r(n) — Recall some display parameters (C)

Applicability All

Syntax `r(n<,noupdate>)`

Description `r(n)` recalls only the following parameters: sp, wp, sp1, wp1, sp2, wp2, sc, wc, sc2, wc2, ho, vo, vs, and ai/nm.

`noupdate` — as a second argument prevents the automatic update of interactive programs.

Arguments `n=1 to 9`

See also *User Programming*

Related
`fr(n)`	Recall all the parameters of the specified display parameter set (C)
`s(n)`	Save a copy of the current values of all display parameters (C)

`r1-r7` Real-value storage for macros (P)

Description The seven parameters `r1`, `r2`, `r3`, `r4`, `r5`, `r6`, and `r7` are available in each experiment for macros to store a real value.

See also *User Programming*

Related
`dgs`	Display group of special/automation parameters (M)
`n1,n2,n3`	Name storage for macros (P)

`ra` Resume acquisition stopped with sa command (C)

Description Resumes an experiment acquisition that was stopped with the `sa` command. `ra` is not permitted after any parameters have been brought into the stopped experiment with the `rt` or `rtp` macros. The parameters `dp` and `np` may not be altered.

`ra` applies to the experiment that you are joined to at the time the command is entered. If experiment 1 has been previously stopped with `sa`, you must be joined to experiment 1 for ra to resume that acquisition. If you are in experiment 2, entering `ra` has no effect on experiment 1.

If an experiment has been stopped with `sa`, you can increase the number of transients `nt` and resume the acquisition with `ra`. You cannot, however, increase `nt` and enter `ra` if the experiment had completed in a normal fashion (i.e., it was not stopped with `sa`).

Note that the completion time and remaining time shown in the Acquisition Status window are not accurate after `ra` is executed.

See also *NMR Spectroscopy User Guide*

Related
`dp`	Double precision (P)
`np`	Number of data points (P)
`nt`	Number of transients (P)
`rt`	Retrieve FID (M)
`rtp`	Retrieve parameters (M)
`sa`	Stop acquisition (C)

random — Return a random number

Syntax `random<(max <,'real'>)>:val`

Applicability VnmrJ 3.1

Description Return a random number. By default, it returns a random integer between 0 and 2^31-1. (2^31-1 is 2147483647 or, in hexadecimal, 0x7fffffff). If an optional number is supplied, the returned value will be between 0 and that value. If an optional keyword 'real' is supplied, the random number will be returned as a real number.

Arguments The difference between random(10) and random(10,'real') is that the first will only return whole numbers between 0 to 10. The second call, with the 'real' option, will return fractional numbers such as 2.342, 7.324, etc.

If a max value is supplied, the conversions are slightly different if a real number or integer is returned. This is to avoid truncation problems with integer math.

For real numbers:

```
double val = random();
val = val / 2147483647.0;  /* results in value from 0.0 to 1.0 */
val = val * max; /* scales from 0.0 to max (max may be negative) */
```

For integers:

```
long val = random();
long range = 2147483647 / (abs(max) + 1);  /* determine size of max + 1 ranges of integers */
val = val / range;
if (max < 0)
   val = -val;
```

Arguments

Examples To return a random real number between 0.0 and 1.0, use

random(1,'real'):$val

rcvrwt — Weighting for different receivers (P)

Applicability Systems with multiple receivers.

Description An array of real numbers giving weighting factors to use when combining multiple receiver data. The i'th array element is used to weight data from the i'th receiver. Applying a weight factor is like increasing the gain of the receiver by the same factor (but the weights are specified as numerical factors rather than in dB).

Examples `rcvrwt=10,12,8`

react Recover from error conditions during werr processing (M)

Syntax `react<('wait')>`

Description When an acquisition error occurs, any action specified by the werr parameter is executed. The react macro is a prototype for handling these errors. This macro can be invoked for error handling by setting werr='react'. The acqstatus parameter is provided so that react can determine which specific error has occurred.

Arguments 'wait' is a keyword for a special type of error handling during an automation run. The react macro always uses the 'next' option when it calls the command au. Under certain conditions, it is also appropriate to use the 'wait' option. react checks to see if an argument was passed to it; that is, werr='werr(\'wait\')' to determine whether to use the 'wait' option of au.

See also *NMR Spectroscopy User Guide*

Related

acqstatus	Acquisition status (P)
au	Submit experiment to acquisition and process data (C)
werr	Specify action when error occurs (C)
werr	When error (P)

readallshims Read all shims from hardware (M)

Description Reads all shims from the hardware and sets the values into the shim parameters in the current parameter tree. The shims used depend on the shimset configuration. For the shim set on the Ultra•nmr shim system, readallshims is active only if hardware-to-software shim communication is enabled.

See also *NMR Spectroscopy User Guide*

Related

load	Load status of displayed shims (P)
readhw	Read current values of acquisition hardware (C)
setallshims	Set all shims into hardware (M)
sethw	Set values for hardware in acquisition system (C)
shimset	Type of shim set (P)
su	Submit a setup experiment to acquisition (M)

readbrutape Read Bruker data files from 9-track tape (U)

Syntax (From UNIX) `readbrutape file <number_skipped>`

Description A shell script that reads one file from a Bruker tape into a UNIX file with the name specified. Bruker tapes are likely to be made at 1600 bpi, although 1600 bpi is not a requirement.

Arguments file is the name of the file read into UNIX. For identification, the .bru extension is added to the file name.

`number_skipped` is the number of files skipped and *includes* the header file (which is assumed to be the first file on the tape). The default is the script reads the first file after the header file. If `number_skipped` equals 0, there is no rewinding and the first file (or the next file) on the tape is read.

See also *NMR Spectroscopy User Guide*

Related `convertbru` Convert Bruker data (M,U)

`readfile` Read the contents of a text file into two parameters (C)

Examples `readfile (path, par1, par2, <,cmpstr <,tree> >):num`

Description `readfile` reads the contents of a file and puts the contents into two supplied parameters. The first word on each line in the file is placed in the first parameter. The remainder of the line is placed in the second parameter. An optional fourth argument specifies a string which is used to match the first word of the line. For example, if the file contained:

```
H1pw 10
H1pwr 55
C13pw 14
C13pwr 50
```

and the comparison string was set to `H1`, only the lines starting with `H1` would be put into the parameters. Namely, `H1pw` and `H1pwr`.

Arguments `path` is the path name of the file to read.

`par1` is the name of the parameter to hold the first word of the line.

`par2` is the name of the parameter to hold the remainder of each line.

`cmpstr` is the optional comparison string for matching the first word.

`tree` is an optional parameter to select the `tree` for `par1` and `par2`. The possibilities are `current`, `global`, and `local`. `Current` is the default. `Local` is used if the parameters are $macro parameters. If `tree` is used, the `cmpstr` must also be supplied. If `cmpstr` is `''`, then it is ignored.

The `par1` and `par2` parameters must already exist. If `par1` or `par2` are defined as a real parameter, as opposed to a string parameter, then if the value does not have a number as the first word, a zero will be assigned.

`num` will be set to the number of items in the arrayed parameters `par1` and `par2`.

Lines that only contain white space are not added to the parameters. Lines that start with a `#` are not added to the parameters. Lines which start with a `#` can be used as comment lines. If a line only contains a single word, that word is put into the first parameter. The corresponding array element of the second parameter will be set to an empty string. The `readfile` will return the number of lines added to the parameters.

Examples

Examples using a prototype file containing the following:

```
# A readfile test case
# Proton values
H1pw 10
H1pwr 55
# Carbon values
C13pw 14
C13pwr 50
H1macro ft f full aph vsadj
End
```

```
readfile(systemdir+'/probes/testcase','attr','vals')
```

This sets the `attr` and `vals` parameters to arrays of six strings.

```
attr='H1pw','H1pwr','C13pw','C13pwr','H1macro','End'
vals='10','55','14','50','ft f full aph vsadj',''
```

```
readfile(systemdir+'/probes/testcase','attr','vals','H1
')
```

This sets the `attr` and `vals` parameters to arrays of three strings.

```
attr='H1pw','H1pwr','H1macro'
vals='10','55','ft f full aph vsadj'
```

The `readfile` command might be used in conjunction with the `teststr` command. The `teststr` command can be used to search an arrayed parameter to determine the index of a specified element.

For example,

```
teststr(attr,'H1pwr'):$e
vals[$e] will be the value of 'H1Pwr'
```

readhw — Read current values of acquisition hardware (C)

Syntax

`readhw(“param1”,”param2”,...)<:r1,r2,...>`

`readhw(“keword”):$res1,...`

Description

Returns or displays the current values of the lock system parameters `lockpower`, `lockgain`, `lockphase`, `lock`, `temp`, `loc`, and `z0`.

The values of the shims can also be obtained. The particular shims that can be read depends upon the type of shim hardware present in the system. See the description of `shimset` for a list of the shim names for each type of shim hardware.

Shim DACs read by readhw:

- Axial shim: `z1, z2, z3, z4, z1c, z2c`
- Non-axial shims: `x1, y1, xz, yz, xy, x2y2, x3, y3`
- Special Oxford magnets shims: `z5, xz2, yz2, zx2y2, zxy`

Arguments

`param1,param2,...` parameter to read — maximum of 10 parameters.

`r1,r2,...` Vnmr variables hold the returned results

no variables supplied — results are displayed in the text panel

Keywords:

`loc` —sample changer location.

`temp` — returns the sample temperature, controller status, and set point. Results are displayed in the text panel if no variables are supplied

Returned value	Status
0	Regulation off
1	Regulated
2	Not regulated
3	No controller

`status` — returns the systems status as an integer. The returned values are:

Returned value	Status
10	IDLE
15	PARSE
16	PREP
17	SYNCED
20	ACQUIRE
25	PAD
30	VTWAIT
40	SPINWAIT
50	AGAIN
60	ALOCK
61	AFINDRES
62	APOWER
63	APHASE
70	SHIMMING
80	SMPCHANGE
81	RETRIEVSMP
82	LOADSMP
90	INTERACTIVE
100	TUNING
0	INACTIVE
Error messages	
-1	Available on spectrometer only (i.e. system = 'datastation')
-2	acquisition not active (acquisition communication programs are not running try running `su acqproc`).
-7	console powered down or not connected

Results are displayed in the text panel if no variables are supplied.

`readhw` cannot be used when an acquisition is in progress or when `acqi` is connected to the acquisition system.

Arguments `param1, param2,...` are the names of the parameters to be read. `value1,value2,...` are return variables to store the settings of the parameters specified. The default is to display the setting in the status window.

Examples

```
readhw('z1c','z2c','z1','z2')
readhw('z1c','z2c','z1','z2'):r1,r2,r3,r4
readhw('temp'):$t sets $t
```

See also *NMR Spectroscopy User Guide*

Related

`lockgain`	Lock gain (P)
`lockphase`	Lock phase (P)
`lockpower`	Lock power (P)

readallshims	Read all shims from hardware (M)
sethw	Set values for hardware in the acquisition system (C)
shimset	Type of shim set (P)

readlk Read current lock level (C)

Syntax `readlk<:lock_level>`

Description Returns the same information as would be displayed on the digital lock display using the manual shimming window. `readlk` can be used in developing automatic shimming methods such as shimming via grid searching. It *cannot* be used during acquisition or manual shimming.

Arguments `lock_level` returns the current lock level.

Examples `readlk`
`readlk:$levell`

See also *User Programming*

Related alock Automatic lock status (P)

readparam Read one of more parameters from a file (C)

Syntax `readparam(file,parlist[,tree[,type]])-`

Description The `readparam` command will read one or more parameters from a specified file. The first argument is the name of the file. The second argument is a list of the names of the parameters to be read. It is a string parameter and the names can be separated either by a space or a comma. If a parameter in the list is not present in the file being read, no error is generated. The optional third argument is the tree into which the parameters are read. The variable trees are 'current', `'global'`, `'processed'` and `'systemglobal'`. The optional fourth argument controls the behavior of the `readparam` command. The options are `'read'`, `'replace'`, and `'add'`. The default type is `'read'`.

Examples In order to specify the type, the tree must also be specified. The behaviors are best illustrated with specific examples. Lets say that there is a temporary file containing only the parameters a and b. We are going to use the readparam command to read parameters into a current tree which contains the parameters a and c but does not contain the parameters b and d. This can be summarized as:

Parameters in mypar: a=1 b=2

Initial parameters in current tree: a=4 c=8 (b and d do not exist)

`readparam(curexp+'/mypar','a b c d','current','read')`

Parameter in a current tree is replaced with parameter from `mypar`. Parameter b in current tree is read in from mypar Parameter c in current tree is unaltered Parameter d in current tree still does not exist. Final parameters in current tree: a=1 b=2 c=8 (d does not exist).

```
readparam(curexp+'/mypar','a b c
d','current','replace')
```

Parameter in a current tree is replaced with parameter from `mypar`. Parameter b in current tree still does not exist. Parameter c in current tree is deleted. Parameter d in current tree still does not exist. Final parameters in current tree: a=1 (b c and d do not exist).

```
readparam(curexp+'/mypar','a b c d','current','add')
```

Parameter in a current tree is unaltered. Parameter b in current tree is read in from `mypar` Parameter c in current tree is unaltered. Parameter d in current tree still does not exist. Final parameters in current tree: a=4 b=2 c=8 (d does not exist).

This command may be used to read temporary values which have been saved with the `writeparam` command.

More Examples:

```
readparam(curexp+'/mypar','in')
```

reads the parameter in from the file mypar in the current experiment directory.

```
readparam(curexp+'/mypar','sw ct np','processed')
```

reads the parameters sw, ct, and np into the processed tree from the file mypar in the current experiment directory.

`readultra` Read shim coil setting for Ultra•nmr shim system (M)

Applicability Systems with the Ultra•nmr shim system.

Syntax `readultra<(file_number)>`

Description Reads shim set files for a Ultra•nmr shim system from a Sun floppy disk into VnmrJ. The floppy disk for Ultra•nmr contains up to 63 shim sets named `file1.dac` to `file63.dac`.

Arguments `file_number` is the number of the shim set file, from 1 to 63. The default is to read all of the shim set files.

Examples `readultra`
`readultra(6)`

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)
`svs` Save shim coil settings (C)

`real` Create a real variable without a value (C)

Syntax `real(variable)`

Description Creates a real variable without a value.

Arguments `variable` is the name of the variable to be created.

Examples `real('realval1')`

See also *User Programming*

Related

create	Create a new parameter in a parameter tree (C)
string	Create a string variable (C)

recon_all Reconstruct images from 2D MRI fid data (C)

Applicability Imaging Systems

Syntax `recon_all(acqstring,<pc option>)`
or
`recon_all(acqstring,<image directory>,<pc option>)`
or
`recon_all`

Description Produces 2D images (in fdf format) from FID data acquired with most 2D imaging sequence, including sems, gems, fsems, and epi. Supported features:

- Compressed/Standard/Arrayed experiments supported (relevant VNMR parameter: seqcon)
- Capable of running concurrently with acquisition (set acqstring to acq after first wnt; empty or dummy string initially).
- Disable image display (relevant parameter: recondisplay. Create in processed tree as a real variable and set it to 0)
- Display every N images (relevant parameter: recondisplay. Create in processed tree as a real variable and set it to N)
- DC removal (relevant parameter: dcrmv)
- Image shifting (relevant VNMR parameter: lsfrq, lsfrq1)
- Multi-shot/sorting (relevant parameters: petable, etl, and/or nseg)
- Multi-slice (interleaved) acquisitions (relevant VNMR parameter: ns)
- Separate output from multiple receivers (relevant VNMR parameter: rcvrout, a string. Set to i, will yield either raw- (if VNMR parameter raw is set) or image-domain magnitude and phase images for separate coils)
- Multi-echo imaging support (sems, epi) (relevant VNMR parameter: ne)
- Multiple receiver data (magnitude sum) (relevant parameter: rcvrs)
- Weighting (through VnmrJ panel selections) (relevant parameter: ftproc)
- Zero filling (through VnmrJ panel selections) (relevant parameters: fn and/or fn1)
- Output magnitude and/or phase raw data components. (relevant (optional) parameter: raw. Create in processed tree as a string which can be set to 'm' (magnitude), 'p' (phase), or 'b' (both))
- Partial k-space conjugation. Relevant parameters are fract_kx and fract_ky, which denote the number of points/echoes acquired beyond the intended N/2. Example: nv=80, fract_ky=16 results in the central 32 echoes used as a correction map prior to conjugate

synthesis. Resulting image has 128 (2*(80-16)) lines in the phase encoded direction.

- Phase correction (relevant parameters: image, `epi_pc`). Implemented for epi sequences. Phase of transformed imaging data (`image=1`) is corrected by phase of transformed reference data (`image=0`). Accepted values for `pc option` in command string or for the optional parameter `epi_pc` are:

  `POINTWISE` (the default; direct use of the phase of profile)

  `LINEAR` (1st order fit of phase of profile)

  `QUADRATIC` (2nd order fit of phase of profile)

  `CENTER_PAIR` (even/odd pair at center of echo train used for all even/odd echoes)

  `PAIRWISE` (even/odd pair phase differences along echo train used)

  6.`FIRST_PAIR` (1st and 2nd echoes used for even/odd correction)

- Navigator Echo correction. Requires acquisition of *echo train* data (`fsems`, `epi`), some of which are not phase encoded. Adjusts phase of encoded echoes according to the phase of navigator echoes of the same echo train, relative to the first such navigator echo. Relevant parameters are:
- `navigator` (can be string set to `'y'` or `'n'`, or array of integers giving navigator echo positions within the echo train (i.e., `navigator=1,2`).)
- `nav_type` (optional; string, set to `'off'` to disable correction or `'POINTWISE'` (default)).

  Order of operation per echo in block:

  1. DC removal
  2. echo reversal if necessary
  3. raw data output if requested
  4. windowing if necessary
  5. read direction Fourier transform
  6. phase correction if necessary
  7. sorting if necessary

Order of operation per slice:

1 navigator correction if necessary
2 windowing in phase direction if necessary
3 partial Fourier correction if necessary
4 phase direction Fourier transform
5 accumulation of multi-receiver data
6 write `fdf` output file

Arguments

acqstring	Set to `'acq'` to indicate concurrent reconstruction; performs no initialization. Any other value can be used for retrospective reconstruction or the first pass through concurrent reconstruction (initialization is performed).
pc option	Optional argument to specify phase correction method (see description of phase correction below).
image directory	Optional argument to specify the directory which will contain produced `fdf` files.
NB	`recon_all` accesses parameters in the PROCESSED tree for control of some features. It is in the PROCESSED tree that variables should be created and/or modified for effectiveness with `recon_all`.
Input/Output	`recon_all` reads the FID file in the `acqfil` subdirectory of the current experiment, and creates `fdf` files that are written to the `recon` subdirectory of the current experiment when run in standalone mode, or to the study tree when run in study mode. If raw data output is selected, the resulting fdf files are written to the `rawmag` or `rawphs` subdirectory of the current experiment. If phase images are optionally generated, the resulting fdf files are written to the `reconphs` subdirectory of the current experiment's directory.

Examples
recon_all('','/usr/home/myimages')
recon_all('','/usr/home/myimages','CENTERPAIR')
recon_all('ignorethis','LINEAR')
recon_all('acq')

See also *VnmrJ Imaging User’s Guide*

`record` Record keyboard entries as a macro (M)

Syntax `record<(file|'off')>`

Description Records keyboard entries and stores the entries as a MAGICAL macro in the user’s `maclib` directory. To start recording keyboard entries, enter `record`. You are prompted for a macro name (you can also give the name as an argument to `record`). The command line prompt then becomes “Command?” to indicate that the `record` macro is active. Type the MAGICAL commands to be recorded on the keyboard.

Function keys can be included by entering `F1` to `F8` for function keys 1 to 8, respectively. Enter `off` or `record('off')` to finish the recording.

Arguments `file` is the name of the macro file in which the entries are saved. The default is that the user is prompted for a file name. If the macro file name already exists, the user is asked if the file should be overwritten.

`'off'` is a keyword to stop recording the entries.

Examples
```
record
record('mymacro')
record('off')
```

See also *User Programming*

redor1 Set up parameters for REDOR1 pulse sequence (M)

Applicability Three-channel systems with a triple-tuned MAS solids probe.

Description Sets up a parameter set, obtained withXPOLAR1, for REDOR (rotational echo double-resonance) experiment.

See also *User Guide: Solid-State NMR*

Related		
	`xpolar1`	Set up parameters for XPOLAR1 pulse sequence (M)

redosy Restore 2D DOSY display from sub experiment (M)

Description Restores the previous 2D DOSY display (if one exists) by recalling the data stored by the `dosy` macro in the file `subexp/dosy2Ddisplay` in the current experiment. `undosy` and `redosy` enable easy switching between the 1D DOSY data (spectra as a function of `gzlvl`) and the 2D DOSY display (signal as a function of frequency and diffusion coefficient).

See also *NMR Spectroscopy User Guide*

Related		
	`dosy`	Process DOSY experiments (M)
	`undosy`	Restore original 1D NMR data from subexperiment (M)

reff1 Reference f1 Indirect Dimension from Observe Dimension (M)

Syntax `reff1<(refsource1)>`

Description Macros uses the ratio of the Ξ values for the relevant nuclei from `refsource1` or the reference source specified to determine the reference frequency in the f1 indirect dimension directly from the reference frequency in the observe dimension using the formula:

```
reffrq1 = (reffrq / Ξ[tn]) * Ξ[nucf1]
rfp1=0
rfl1 = sw1/2 - (frq[f1] - reffrq1)*1e6
```

Ξ is the normalized frequency such that the ^{1}H signal from TMS is 100.00 MHz.

Referencing in the observe dimension using `setref` and this method is same as using `setref1` (apart from minor round-off errors).

Referencing the observe dimension to an internal reference standard as proposed by IUPAC references all dimensions to that single reference signal and not the lock as with `setref`, `setref1`, and `setref2`.

Limitations: the macro works with data recalled from an archive or acquired on an other system provided the data was acquired using VNMR6.1C or newer.

Referencing is based on `nuctables/nuctabrefBio` if `bioref='y'` (global or local). Setting `bioref='n'` (global or local) or if the flag does not exist the standard IUPAC / organic chemistry referencing (`nuctables/nuctabref`) is used.

See `/vnmr/nuctables/nuctabref`.

Arguments: No argument — reference source is determined from `refsource1`. If the relevant parameter is missing, the macro tries to determine the (indirect) reference source from the `axis` parameter.

`'sfrq'`, `'dfrq'`, `'dfrq2'`, `'dfrq3'`, or `'dfrq4'` as a reference source

Examples: `reff1 reff1('sfrq')`

Related:

`reff2`	Reference f2 Indirect Dimension from Observe Dimension (M)
`setref`	Set Frequency Referencing for Proton Spectra (M)
`setref1`	Set Frequency Referencing for f1 Evolution Dimension (M)
`setref2`	Set Frequency Referencing for f2 Evolution Dimension (M)
`mref`	Set Referencing Based on Spectrum from the same sample (M)
`bioref`	Flag for Bio-NMR Referencing (P)

`reff2` Reference f2 Indirect Dimension from Observe Dimension (M)

Syntax: `reff2<(refsource2)>`

Description: Macros uses the ratio of the Ξ values for the relevant nuclei from `refsource1` or the reference source specified to determine the reference frequency in the f1 indirect dimension directly from the reference frequency in the observe dimension using the formula:

```
reffrq1 = (reffrq / Ξ [tn]) * Ξ [nucf1]
rfp1=0
rfl1 = sw1/2 - (frq[f1] - reffrq1)*1e6
```

Ξ is the normalized frequency such that the ^{1}H signal from TMS is 100.00 MHz.

Referencing in the observe dimension using `setref` and this method is same as using `setref1` (apart from minor round-off errors).

Referencing the observe dimension to an internal reference standard as proposed by IUPAC references all dimensions to that single reference signal and not the lock as with `setref`, `setref1`, and `setref2`.

Limitations: the macro works with data recalled from an archive or acquired on an other system provided the data was acquired using VNMR6.1C or newer.

Referencing is based on `nuctables/nuctabrefBio` if `bioref='y'` (global or local). Setting `bioref='n'` (global or local) or if the flag does not exist the standard IUPAC / organic chemistry referencing (`nuctables/nuctabref`) is used.

See `/vnmr/nuctables/nuctabref`.

Arguments No argument — reference source is determined from `refsource2`. If the relevant parameter is missing, the macro tries to determine the (indirect) reference source from the `axis` parameter.

`'sfrq'`, `'dfrq'`, `'dfrq2'`, `'dfrq3'`, or `'dfrq4'` as a reference source

Examples `reff2('dfrq3')`

Related

`reff1`	Reference f2 Indirect Dimension from Observe Dimension (M)
`setref`	Set Frequency Referencing for Proton Spectra (M)
`setref1`	Set Frequency Referencing for f1 Evolution Dimension (M)
`setref2`	Set Frequency Referencing for f2 Evolution Dimension (M)
`mref`	Set Referencing Based on Spectrum from the same sample (M)
`bioref`	Flag for Bio-NMR Referencing (P)

`reffrq` Reference frequency of reference line (P)

Description Reference frequency, in MHz, of the reference line. This parameter is set by the `rl` macro. By defining `reffrq` as the conversion factor between Hz and ppm using the `unit` command, ppm calculations can be made.

If referencing is on (i.e., `refpos` is not set to `'n'`), the `go`, `ga`, and `au` macros calculate values of `rfl` and `rfp` based on `reffrq` and `refpos`. If referencing is off, `go`, `ga`, and `au` set `reffreq` to `sfrq`.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (M)
crl	Clear reference line in directly detected dimension (M)
ga	Submit experiment to acquisition and FT the result (M)
go	Submit experiment to acquisition (M)
reffrq1	Ref. frequency of reference line in 1st indirect dimension (P)
reffrq2	Ref. frequency of reference line in 2nd indirect dimension (P)
refpos	Position of reference frequency (P)
rfl	Reference peak position in directly detected dimension (P)
rfp	Reference peak frequency in directly detected dimension (P)
rl	Set reference line in directly detected dimension (M)
sfrq	Transmitter frequency of observe nucleus (P)
unit	Define conversion units (C)

reffrq1 Reference freq. of reference line in 1st indirect dimension (P)

Description Reference frequency, in MHz, of the reference line in the first indirect dimension of a nD experiment. This parameter should be used as the conversion factor between hertz and ppm in the first indirect dimension.

See also *NMR Spectroscopy User Guide*

Related

crl1	Clear reference line in 1st indirectly detected dimension (M)
reffrq	Reference frequency of reference line (P)
refpos1	Position of reference frequency in 1st indirect dimension (P)

reffrq2 Reference freq. of reference line in 2nd indirect dimension (P)

Description Reference frequency, in MHz, of the reference line in the second indirect dimension of a 2D experiment. This parameter should be used as the conversion factor between hertz and ppm in the second indirect dimension.

See also *NMR Spectroscopy User Guide*

Related

crl2	Clear reference line in 2nd indirectly detected dimension (M)

reffrq	Reference frequency of reference line (P)
refpos2	Position of reference frequency in 2nd indirect dimension (P)

refpos Position of reference frequency (P)

Description Position of reference frequency, set by the setref and rl macros. Setting refpos='n' indicates that referencing has been turned off. The crl macro turns referencing off.

Values Because all spectra are (by definition) referenced to a frequency at 0 ppm, refpos is either 0 or "not used".

See also *NMR Spectroscopy User Guide*

Related

crl	Clear reference line in directly detected dimension (M)
reffrq	Reference frequency of reference line (P)
refpos1	Position of reference frequency in 1st indirect dimension (P)
refpos2	Position of reference frequency in 2nd indirect dimension (P)
rl	Set reference line indirectly detected dimension (M)
setref	Set frequency referencing (M)

refpos1 Position of reference frequency in 1st indirect dimension (P)

Description Position of reference frequency in the first indirect dimension of a nD experiment, set by setref1 and rl1 macros. Setting refpos1='n' indicates that f1 referencing has been turned off. The crl1 macro turns f1 referencing off.

Values Because all spectra are (by definition) referenced to a frequency at 0 ppm, refpos1 is either 0 or "not used".

See also *NMR Spectroscopy User Guide*

Related

crl1	Clear reference line in 1st indirectly detected dimension (M)
reffrq1	Ref. frequency of reference line in 1st indirect dimension (P)
refpos	Position of reference frequency (P)
rl1	Set reference line in 1st indirect dimension (M)
setref1	Set frequency referencing for 1st indirectly detected dimension (M)

refpos2 Position of reference frequency in 2nd indirect dimension (P)

Description
Position of reference frequency in the second indirect dimension of a 3D experiment, set by setref2 and rl2 macros. Setting refpos2='n' indicates that f2 referencing has been turned off in 3D spectra. The crl2 macro turns f2 referencing off.

Values
Because all spectra are (by definition) referenced to a frequency at 0 ppm, refpos2 is either 0 or "not used".

See also
NMR Spectroscopy User Guide

Related

crl2	Clear reference line in 2nd indirectly detected dimension (M)
reffrq2	Ref. frequency of reference line in 2nd indirect dimension (P)
refpos	Position of reference frequency (P)
rl2	Set reference line in 2nd indirect dimension (M)
setref2	Set frequency referencing for 2nd indirectly detected dimension (M)

refsource1 Center frequency in 1st indirect dimension (P)

Description
Holds a parameter name to be used as the center frequency in the first indirect dimension of 2D experiments. If refsource1 does not exist, the default is 'sfrq'.

For 2D experiments, the second dimension may be related to sfrq if it is a homonuclear experiment. The second dimension may also be related to dfrq if it is a heteronuclear experiment. refsource1 would then be set as refsource1='sfrq' and refsource1='dfrq', respectively.

See also
NMR Spectroscopy User Guide

Related

dfrq	Transmitter frequency of first decoupler (P)
refsource2	Center frequency in 2nd indirect frequency (P)
sfrq	Transmitter frequency of observe nucleus (P)

refsource2 Center frequency in 2nd indirect dimension (P)

Description
Holds a parameter name to be used as the center frequency in the second indirect dimension. refsource2 is analogous to refsource1

See also
NMR Spectroscopy User Guide

Related

refsource1	Center frequency in 1st indirect dimension (P)

region Divide spectrum into regions (C)

Syntax `region<(tail_length,relative_number,threshold,`
`number_points,tail_size)><:number_regions >`

Description Breaks a spectrum up into regions containing peaks.

Arguments `tail_length` is the length from 0.0 to `sw`, in Hz, that is added to the start and end of each calculated peak region; default value is `sw/10`. The default value is used if a negative number is entered for this argument. If the addition of these wings would cause overlap between adjacent regions, the wings are reduced until the regions no longer overlap.

`relative_number` is a number that, in combination with other factors, governs the relative number of regions to be found. The default is 12, which is used if 0 is entered for this argument. `relative_number` is used as part of a test to determine whether two spectral areas containing peaks are close enough together to be represented as a single region. There are no strict rules that associate the value of `relative_number` to the total number of regions that will be found. In general, increasing this number decreases the number of regions that will be found and increases the size of an individual region. A value of `1` would give more regions; a value of `100` would give fewer regions.

`threshold` is a sensitivity factor used to decide if a data point is large enough, relative to the noise level, to qualify it as part of a peak. The default value is `0.6`, which is used if `0` is entered for this argument. Smaller values of `threshold` make peak selection more sensitive; larger values make peak selection less sensitive.

`number_points` governs the number of successive data points, normally from `7` to `40`, that must qualify as part of a peak (see the description of `threshold` above) in order for that spectral area to be considered a real peak. The default value is a function of `fn`, `sw`, weighting functions, and other values. The default is used if `0` is entered for this argument. For carbon spectra with large spectral windows, experimental peaks often contain only one or two data points. Adjust `number_points` to `1` or `2` in those cases.

`tail_size` is a number that, in combination with `relative_number` and other factors, governs whether two spectral areas that contain peaks are close enough together to be represented as a single region. The default value is used if `0` is entered for this argument.

`number_regions` is the total number of regions determined by `region`.

Examples
```
region
region:$1
region(50,0,1)
region(-1,0,0,2):r1
```

See also *NMR Spectroscopy User Guide*

Related		
	`fn`	Fourier number in directly detection dimension (P)
	`sw`	Spectral width in directly detected dimension (P)

relayh — Set up parameters for RELAYH pulse sequence (M)

Description Sets up parameters for absolute-value COSY, or a single or double RELAY-COSY pulse sequence.

See also *NMR Spectroscopy User Guide*

Related `Cosy` Set up parameters for COSY pulse sequence (M)
`cosyps` Set up parameters for phase-sensitive COSY (M)
`Dqcosy` Set up parameters for double quantum filtered COSY (M)

rename — Move and/or rename a file (C)

Syntax `rename(from_file,to_file)`

Description Renames and/or moves a file or directory. `rename` is identical in function to the command `mv`.

Arguments `from_file` is the name of the file to be moved to renamed.

`to_file` is the name of the file after moving or renaming it. If the `from_file` argument has an extension such as `.fid` or `.par`, be sure the `to_file` argument has the same extension.

Examples
```
rename('/home/vnmr1/vnmrsys/seqlib/d2pul',
   '/vnmr/seqlib/d2pul')
```

See also *NMR Spectroscopy User Guide*

Related `copy` Copy a file (C)
`cp` Copy a file (C)
`delete` Delete a file, parameter directory, or FID directory (C)
`mv` Move and/or rename a file (C)
`rm` Delete file (C)

reorder3D — Reorders array elements in arrayed phase sensitive 2D experiment

Syntax `reorder3D`

Applicability VnmrJ 3.1

Description Exchanges the order of the two arrayed parameters in an arrayed phase sensitive 2D experiment. Useful if `3D DOSY` data are acquired with array='phase,gzlvl1' instead of array='gzlvl1,phase'.

See also `dosy`

reqparcheck Flag which enables/disables required parameters (P)

Syntax `reqparcheck= 'y' or 'n'`

Description:

Description The parameter reqparcheck is a flag with the possible values of 'y' or 'n'. Only if it is set to 'y' are actual parameters compared to the file. If it is set to 'n', reqpartest will always return 0.

Values `'y'` or `'n'`, indicating whether required parameters are to be checked.

Related

callacq	Utility macro to call Acq command (M)
reqparlist	List of required parameters (P)
reqparclear	Clears the parameters in required parameter list (M)
reqpartest	Tests whether required parameters are set (M)

reqparclear Clears the parameters in required parameter list (M)

Syntax `reqparclear`

Description Clears the parameters listed in reqparlist. If for some reason reqparlist has been destroyed, then this macro exits without a message. The parameter is cleared on the current tree, if it exists there, or on the global tree, if it exists there. If it exists in neither place, a message is printed and the routine moves on to the next parameter in reqparlist.

The definition of "clear" is that real parameters are turned "off" and string parameters are set to the empty string ''.

There is a known issue with this macro, which due to its obscurity will remain as "user beware." The issue is that if a parameter of the same name exists in both the 'global' and 'current' trees, and if that parameter is part of reqparlist, then it will be cleared in the 'current' tree but not in the `global` tree. Users should just not be doing this.

Also note that while this macro checks for reqparlist='', if it is an array and any element in the array is '' then it assumes '' is a parameter and reports a "does not exist" message.

Related

callacq	Utility macro to call Acq command (M)
reqparcheck	Flag which enables/disables required parameters (P)
reqparlist	List of required parameters (P)
reqpartest	Tests whether required parameters are set (M)

reqparlist List of required parameters (P)

Description The parameter reqparlist holds the parameter names. It is an array of strings. It will not array the experiment.

Related

callacq	Utility macro to call Acq command (M)
gettoken	Utility macro to separate a string into tokens (M)
reqparcheck	Flag which enables/disables required parameters (P)
reqparclear	Clears the parameters in required parameter list (M)
reqpartest	Tests whether required parameters are set (M)

reqpartest Tests whether required parameters are set (M)

Syntax `reqpartest<('showtext'|'showgui'<,callback_string>)>`

Description If the parameter `reqparcheck='y'`, then this macro examines the list of parameter names in reqparlist and if all of them exist and are properly set, returns 0. Properly set is defined as a non- empty string for string parameters, or the active bit set (parameter is `'on'`) for real parameters.

This macro also checks the string which is the concatenation of `autoname + globalauto + sqname` for any parameters in that string. Parameters in this string are delimited by $.

For convenience, this macro will return different values depending on the specific non-true condition, as defined in the following table (X is "don't care").

All parameters exist	T	X	F	T	F
All parameters set	T	X	T	F	F
reqparcheck='y	'T	F	T	T	T
return value	0	-1	1	2	3

Also note that the non-existence of either reqparcheck or reqparlist is equivalent to reqparcheck not set to 'y'.

Parameters are checked in the current tree first for existence, and if that parameter exists there, then that tree is checked for whether it is set. If it does not exist in the current tree, then the global tree is checked. If and only if it exists in neither tree is it considered to not exist.

If the argument to this macro is `'showtext'` then if one or more parameters do not exist or are not properly set, then they are listed on the alphanumeric (text) screen.

If the argument to this macro is `'showgui'`, then an entry popup is displayed for both creation (of non-existing parameters) and value entry. The return value is not affected by the fact that the values are now being entered - in other words, the return value is to be interpreted as 'did not exist' or 'was not set' prior to running the macro.

The comprehensive list to check is `reqparlist+autoname+globalauto+sqname`. Some duplicates may occur, and this macro checks and eliminates duplicates.

The argument `callback_string` is an optional argument that gets passed onto VnmrJ, and then gets passed back to `vnmrbg` when the required parameters entry popup closes. VnmrJ and vnmrbg are not otherwise synchronized, so this allows for re-entrance.

Arguments `'showgui'|showtext'`

`'showgui'` displays an entry popup in the required parameter is not set;

`'show text'` displays information about the required parameters in the text window

`callback_string` — optional callback to `vnmrbg` from VnmrJ (ignored in `'showtext'` option)

See also *VnmrJ User Programming*

Related		
	`callacq`	Utility macro to call Acq command (M)
	`reqparcheck`	Flag which enables/disables required parameters (P)
	`reqparclear`	Clears the parameters in required parameter list (M)

`res` Display lineshape of largest peak in the current spectrum (M)

Description The `res` macro finds the tallest peak in a spectrum, within the current values of `sp` and `wp`. It determines the widths at half-height, 0.55% and 0.11% and writes these lineshape values on the graphics screen.

Related `lres` Used to plot lineshape values. (M)

`resetf3` Reset parameters after a partial 3D Fourier transform (M)

Description Restores the acquisition parameter `sw`, the processing parameter `fn`, and the display parameters `sp`, `wp`, `rfl`, and `rfp` in the 3D parameter set, which are read into VnmrJ by either the `select` command or the `dplane` or `dproj` macros. These parameters were modified due to the selection of regional f_3 processing (`ptspec3d ='ynn'`). The original value for each of these parameters is stored in the parameter `$sv`, where `$` represents `sw`, `fn`, `sp`, `wp`, `rfl`, or `rfp` (e.g., `swsv`).

If a 2D plane into VnmrJ is retrieved from a 3D transformed data set that was processed with regional f_3 processing, `resetf3` must be run before executing `ft3d` in that particular VnmrJ environment.

See also	*NMR Spectroscopy User Guide*	
Related	dplane	Display a 3D plane (M)
	dproj	Display a 3D plane projection (M)
	fn	Fourier number in directly detected dimension (P)
	ft3d	Perform a 3D Fourier transform (M)
	ptspec3d	Region-selective 3D processing (P)
	rfl	Ref. peak position in directly detected dimension (P)
	rfp	Ref. peak frequency in directly detected dimension (P)
	select	Select a spectrum or 2D plane without displaying it (C)
	sp	Start of plot (P)
	sw	Spectral width in directly detected dimension (P)
	vnmrjcmd()	Commands to invoke the GUI popup (C)
	wp	Width of plot (P)

resetplotter Reset plotter to system plotter (M)

Description Command to reset a (temporarily chosen) plotter back to the system plotter sysplotter. Command is called by all plotfile/plotpreview and plot/autoplot buttons on plot panels.

resetsampglobal Clears sample global parameters

Description Clears sample global parameter values in the current workspace.

Examples `resetsampglobal`

Related getsampglobal, resetsampglobal, savesampglobal, mvsampglobal, showsampglobal

resolv Set resolution enhancement parameters (M)

Syntax `resolv<(a,b)>`

Description Calculates a default resolution enhancement function, setting up lb and gf based on the acquisition time at. “Zero-filling” is also accomplished, if possible, by making fn ≥>=2*np.

Arguments a sets a value of lb using lb=-0.318/(a*sw). The default for a is 0.1.

b sets a value of gf using gf=b*sw. The default for b is 0.3.

Examples
```
resolv
resolv(.2,.4)
```

See also *NMR Spectroscopy User Guide*

Related		
Related	at	Acquisition time (P)
	fn	Fourier number in directly detected dimension (P)
	gf	Gaussian function in directly detected dimension (P)
	lb	Line broadening in directly detected dimension (P)
	np	Number of data points (P)
	sw	Spectral width in directly detected dimension (P)

`restorenuctable` Calculate & store accurate nuctable for current system (M)

Syntax `restorenuctable`

Description The `setref` contribution is a generic nucleus table, `/vnmr/nuctables/nuctable`, based on a standard proton frequency of 1000.0 MHz. All standard nucleus tables in the `/vnmr/nuctables` are symbolic links pointing to a generic table.

The `restorenuctable` is used to replace the standard links with specific links that to files containing proper and accurately calculated nucleus tables. Problems arising with custom macros and third party software that are not aware of the symbolic links pointing to a generic table can be fixed using this macro.

Commands and utilities that do not scale nuctable entries to the actual proton frequency (as they should) will work better than with the standard tables.

Limitations: `restorenuctable` is not compatible with `qtune` and certain commands in current software.

Examples `restorenuctable`

Related `nuctable` Display nucleus table for a given H1 frequency (M)

`resume` Resume paused acquisition queue (C)

Description Enables continuing submitting experiments to the acquisition system. For experiments initiated with the command `au('wait')`, the acquisition is paused during the time of data processing in order to prevent the acquisition from submitting new experiments that might be queued. `resume` then allows the data processing macro to initiate another acquisition with `au('next')`, which is then performed immediately instead of at the end of the queue.

See also *NMR Spectroscopy User Guide*

Related `au` Submit experiment to acquisition and process data (C)

return Terminate execution of a macro (C)

Syntax `return<(expression1,expression2,...)>`

Description Terminates the execution of a macro and optionally returns values to another calling macro. This is usually used after testing some condition. `return` is used only in macros and not entered from the keyboard.

Arguments `expression1,expression2,...` are return values to another calling macro.

See also *User Programming*

Related `abort` Terminate action of calling macro and all higher macros (C)

rev System software revision level (P)

Description Stores a string identifying the VnmrJ software version for the system. This parameter is not be entered by the user, but can be examined by entering `rev?`.

See also *VnmrJ Installation and Administration*

Related `revdate` System software preparation date (P)

revdate System software preparation date (P)

Description Stores a string identifying the date the current VnmrJ software version was prepared. This parameter is not be entered by the user, but can be examined by entering `revdate?`.

See also *VnmrJ Installation and Administration*

Related `rev` System software revision level (P)

rfband RF band in use (P)

Description Indicates which rf band of the amplifier is in use for each channel.

Values A string, such as `'hlc'`, in which the first channel is determined by the first character, the second channel is determined by the second character, and so forth. The following values are available for each channel:

`'h'` indicates the high rf band is in use on the channel.

`'l'` indicates the low rf band is in use on the channel.

`'c'` indicates the system software will calculate whether to use the high band or the low band for the channel.

See also *NMR Spectroscopy User Guide*

rfblk Reverse FID block (C)

Syntax rfblk(<src_expno>,src_blk_no,dest_expno,dest_blk_no)

Description Reverses and copies data from a source FID block specified by `src_blk_no` to a destination FID block specified by `dest_expno` and `dest_blk_no`, using memory-mapped input and output. The file header determines the size and type of data to reverse.

`rfblk` searches for the source and destination FID file in the directory `$vnmruser/expN/acqfil`; `N` is the requested experiment number or the current experiment number. If the FID file is not open, `rfblk` opens the file, copies the data, and closes the file. If a number of blocks need to be copied, explicitly opening and closing the files with the commands `mfopen` and `mfclose` can significantly speed up the data reformatting process.

`rfblk` can also be used to append blocks of data to a FID file by specifying that the `dest_blk_no` is greater than the number of blocks in a file.

Be aware that `rfblk` can modify data returned to an experiment with the `rt` command. To avoid modification, enter the following sequence of commands before running `rfblk`:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments `src_expno` specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

`src_blk_no` specifies the source block of data to be copied. Block numbers run from 1 to the number of blocks in a file.

`dest_expno` specifies the experiment number of the destination FID file.

`dest_blk_no` specifies the destination block to send the copied data.

Examples `rfblk(1,2,1)` reverses and copies block 1 from the current experiment to block 1 of experiment 2.

See also *User Programming*

Related

`mfblk`	Move FID block (C)
`mfclose`	Memory map close FID file (C)
`mfdata`	Move FID data (C)
`mfopen`	Memory map open FID file (C)
`mftrace`	Move FID trace (C)
`rfdata`	Reverse FID data (C)
`rftrace`	Reverse FID trace (C)

rfchannel Independent control of rf channel selection (P)

Description Gives override capability over the selection of rf channels. rfchannel does not normally exist but can be created by a user with the command create('rfchannel','flag').

The control of each rf channel is built around a collection of parameters and pulse sequence statements. The frequency of channel 1 is set by sfrq and tof, its power by tpwr and tpwrf. The first decoupler uses the corresponding parameters dfrq, dof, dpwr, and dpwrf, respectively. Furthermore, the decoupler can have modulation modes specified with the parameters dmf, dm, dmm, dres, and dseq,. The second decoupler has the same set of parameters as the first decoupler and they are distinguished by appending a 2 to each name. That is, the names aer dfrq2, dof2, dpwr2, dpwrf2, dmf2, dm2, dmm2, dres2, and dseq2. The third decoupler would use parameters with a 3 appended: dfrq3, dof3, dpwr3, dpwrf3, dmf3, dm3, dmm3, dres3, and dseq3. The rfchannel parameter provides a mechanism to override the default parameter usage.

Values A string of one to four characters in which the position of each character identifies the rf channel controlled.

- The first character selects which rf channel (1 to 4) the parameters sfrq, tof, tpwr, etc. control. The first character also identifies the rf channel used as the receiver.
- The second character selects which rf channel (1 to 4) the parameters dfrq, dof, dpwr, etc. control.
- The third character maps the parameter set dfrq2, dof2, dpwr2, etc. to an rf channel (1 to 4).
- The fourth character maps tdfrq3, dof3, dpwr3, etc. to an rf channel (1 to 4).

For example, rfchannel='132' would exchange control of the second and third rf channels from the default parameter usage.

The number of characters in the rfchannel parameter must match the number of real rf channels (defined by the parameter numrfch) and each rf channel must be selected by the parameter.

Besides remapping the parameters to different rf channels, pulse sequence statements are also remapped. For example, if rfchannel='132', then statements decpulse, decshaped_pulse, decoffset, decpower, decspinlock, and so on are applied on rf channel 3 and dec2pulse, dec2shaped_pulse, and so on are applied on rf channel 2.

An obvious use for this remapping is on systems with the decoupler set to U+ H1 Only in the Spectrometer Configuration window. On these systems, if multinuclear pulses are needed and ^{1}H needs to be observed, the parameter sets that assume a dual-broadband system can be used and the parameters remapped by setting rfchannel='21'. However, internal logic checks if the first decoupler is set to U+ H1 Only, tn is set to 'H1', and dn is not set to 'H1'. If these settings are the case, the parameter mapping for rf channels 1 and 2 is exchanged automatically.

See also *NMR Spectroscopy User Guide; User Programming*

Related

create	Create new parameter in parameter tree (C)
dfrq	Transmitter frequency for first decoupler (P)
dm	Decoupler mode for first decoupler (P)
dmf	Decoupler modulation frequency for first decoupler (P)
dmm	Decoupler modulation mode for first decoupler (P)
dn	Nucleus for first decoupler (P)
dof	Frequency offset for first decoupler (P)
dpwr	Power level for first decoupler with linear amplifier (P)
dpwrf	First decoupler fine power (P)
dres	Tip-angle resolution for first decoupler (P)
dseq	Decoupler sequence for first decoupler (P)
numrfch	Number of rf channels (P)
sfrq	Transmitter frequency for observe nucleus (P)
tn	Nucleus for observe transmitter (P)
tof	Frequency offset for observe transmitter (P)
tpwr	Observe transmitter power level with linear amplifiers (P)
tpwrf	Observe transmitter fine power (P)

rfchtype Type of rf channel (P)

Description Configuration parameter for type of rf on each channel. The value for a channel is set using the Type of RF label in the Spectrometer Configuration window. Pulse sequence programs check `rfchtype` to determine if indirect detection should be used for some experiments. Indirect detection occurs automatically if the decoupler is set to U+ H1 Only in the Spectrometer Configuration window, `tn` is set to `'H1'`, and `dn` is not set to `'H1'`.

Values The values of `rfchtype` parallel the `rftype` values. The only distinction is that the setting for `rftype` is `'d'` on the U+ Direct Synthesis and U+ H1 Only entries.

'U+ Direct Synthesis' is the setting for a system with direct synthesis (U+ Direct Synthesis in the Spectrometer Configuration window).

'U+ H1 Only' is a fixed-frequency proton system (U+ H1 Only in Spectrometer Configuration window).

'Deuterium Decoupler' is the setting for a system deuterium decoupler channel.

'Direct Synthesis' is the setting for direct synthesis (Direct Synthesis in the Spectrometer Configuration window).

'Broadband' is the setting for broadband (Broadband in the Spectrometer Configuration window).

'Fixed Frequency' is the setting for fixed frequency (Fixed Frequency in the Spectrometer Configuration window).

'SIS Modulator' is the setting for imaging modulator (SIS Modulator in the Spectrometer Configuration window).

See also *VnmrJ Installation and Administration*

Related

config	Display current configuration and possibly change it (M)
dn	Nucleus for first decoupler (P)
rftype	Type of rf generation (P)
tn	Nucleus for observe transmitter (P)

rfdata Reverse FID data (C)

Syntax
```
rfdata(<src_expno,>src_blk_no,src_start_loc,  \
  dest_expno,dest_blk_no,dest_start_loc,num_points)
```

Description Reverses and copies data specified by `src_start_loc` from a FID block specified by `src_blk_no` to a destination location specified by `dest_expno`, `dest_blk_no`, and `dest_start_loc`, using memory-mapped input and output. The data point locations and the `num_points` to be reversed are specified by data points corresponding to the `np` parameter, not bytes or complex points; however, when reversing the data, `rfdata` looks at the file header to determine the size and type of data to reverse.

`rfdata` searches for the source and destination FID file in the directory `$vnmruser/expN/acqfil`; `N` is the requested experiment number or the current experiment number. If the FID file is not open, `rfdata` opens the file, copies the data, and closes the file. If a number of blocks need to be copied, explicitly opening and closing the files with the commands `mfopen` and `mfclose` can significantly speed up the data reformatting process.

Be aware that `rfdata` can modify data returned to an experiment with the `rt` command. To avoid modification, enter the following sequence of commands before running `rfdata`:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments `src_expno` specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

`src_blk_no` specifies the source block of data to be copied. Block numbers run from 1 to the number of blocks in a file.

`src_start_loc` specifies the starting data location within the specified block to copy the data. Data locations start from 0 and are specified as data points corresponding to the `np` parameter.

`dest_expno` specifies the experiment number of the destination FID file.

`dest_blk_no` specifies the destination block to send the copied data.

`dest_start_loc` specifies the starting data destination location within the specified block to send the copied data.

Examples `rfdata(1,0,2,1,(nv-1)*np,np)` copies and reverses `np` points of data from the starting location 0 of block 1 of the current experiment to the data location `(nv-1)*np` of block 1 of experiment 2.

See also *User Programming*

Related

`mfblk`	Move FID block (C)
`mfclose`	Memory map close FID file (C)
`mfdata`	Move FID data (C)
`mfopen`	Memory map open FID file (C)
`mftrace`	Move FID trace (C)
`rfblk`	Reverse FID block (C)
`rftrace`	Reverse FID trace (C)

`rfl` Reference peak position in directly detected dimension (P)

Description Actual position of the reference line in the spectrum (i.e., the distance from the right edge of the spectrum to the reference line). If there is no reference line in the spectrum, `rfl` can be used to enter the frequency where the reference line would appear if the line were present in the spectrum.

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related

`rfl1`	Reference peak position in 1st indirectly detected dimension (P)
`rfl2`	Reference peak position in 2nd indirectly detected dimension (P)
`rfp`	Reference peak frequency in directly detected dimension (P)

`rfl1` Reference peak position in 1st indirectly detected dimension (P)

Description Analogous to the `rfl` parameter except that `rfl1` applies to the first indirectly detected dimension of a multidimensional data set. `rfl1` can either be set manually or be adjusted automatically when the macro `rl1` is used to assign a reference line.

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related

`rfl`	Reference peak position in directly detected dimension (P)
`rfl2`	Reference peak position in 2nd indirectly detected dimension (P)
`rfp1`	Reference peak frequency in 1st indirectly detected dimension (P)

`rfl2` Reference peak position in 2nd indirectly detected dimension (P)

Description Analogous to the `rfl` parameter except that `rfl2` applies to the second indirectly detected dimension of a multidimensional data set. `rfl2` can either be set manually or be adjusted automatically when the macro `rl2` is used to assign a reference line.

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related

`rfl`	Reference peak position in directly detected position (P)
`rfl1`	Reference peak position in 1st indirectly detected dimension (P)
`rfp2`	Reference peak frequency in 2nd indirectly detected dimension (P)

`rfp` Reference peak frequency in directly detected dimension (P)

Description Sets the frequency to be assigned to the reference line in the spectrum. `rfp` is always stored in Hz, but can be entered in ppm by using the `p` suffix (e.g., `rfp=2.1p`).

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related

`rfl`	Reference peak position in directly detected dimension (P)
`rfp1`	Ref. peak frequency in 1st indirectly detected dimension (P)
`rfp2`	Ref. peak frequency in 2nd indirectly detected dimension (P)
`rl`	Set reference line in directly detected dimension (M)

`rfp1` Reference peak freq. in 1st indirectly detected dimension (P)

Description Analogous to the `rfp` parameter except that `rfp1` applies to the first indirectly detected dimension of a multidimensional data set. `rfp1` can either be set manually or be assigned a value when `rl1` is called with an argument (e.g., `rl1(7.2p)` assigns the value of 7.2 ppm to `rfp1`).

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related

`rfl1`	Ref. peak position in 1st indirectly detected dimension (P)
`rfp`	Ref. peak frequency in directly detected dimension (P)
`rfp2`	Ref. peak frequency in 2nd indirectly detected dimension (P)
`rl1`	Set reference line in 1st indirectly detected dimension (M)

rfp2 Reference peak freq. in 2nd indirectly detected dimension (P)

Description Analogous to the rfp parameter except that rfp2 applies to the second indirectly detected dimension of a multidimensional data set. rfp2 can be set manually or be assigned a value when rl2 is called with an argument. For example, entering rl2(7.2p) assigns the value of 7.2 ppm to rfp2.

Values Number, in Hz.

See also *NMR Spectroscopy User Guide*

Related
- rfl2 Reference peak position in 2nd indirectly detected dimension (P)
- rfp Reference peak frequency in directly detected dimension (P)
- rfp1 Reference peak frequency in 1st indirectly detected dimension (P)
- rl2 Set reference line in 2nd indirectly detected dimension (C)

rftempcomp RF Transmitter Board Temperature Compensation (P)

Syntax

Applicability VnmrJ 3.1

Arguments If rftempcomp='n' temperature compensation on the RF transmitter board is turned off.

If rftempcomp='c' temperature compensation on the RF transmitter board is turned on continuously and will continuously update until it is turned off.

To create the rftempcomp parameter, enter:

```
create('rftempcomp','string','global')
```

rftrace Reverse FID trace (C)

Syntax
```
rftrace(<src_expno,src_blk_no,src_trace_no,   \
  dest_expno,<dest_blk_no,dest_trace_no)
```

Description Reverses and copies FID traces specified by src_trace_no from a FID block specified by src_blk_no to a destination location specified by dest_expno, dest_blk_no, and dest_trace_no, using memory-mapped input and output. The file header determines the size and type of data to be reversed.

rftrace searches for the source and destination FID file in the directory $vnmruser/expN/acqfil; N is the requested experiment number or the current experiment number. If the FID file is not open, rftrace opens the file, copies the data, and closes the file. If a number of blocks need to be copied, explicitly opening and closing the files with the commands mfopen and mfclose can significantly speed up the data reformatting process.

You cannot use rftrace to append data to a FID file. Its purpose is for moving around data.

Be aware that rftrace can modify data returned to an experiment with the rt command. To avoid modification, enter the following sequence of commands before running rftrace:

```
cp(curexp+'/acqfil/fid',curexp+'/acqfil/fidtmp')
rm(curexp+'/acqfil/fid')
mv(curexp+'/acqfil/fidtmp',curexp+'/acqfil/fid')
```

Arguments src_expno specifies the experiment number of the source FID file. The default is the FID file of the current experiment.

src_blk_no specifies the source block of data to be copied. Block numbers run from 1 to the number of blocks in a file.

src_trace_no specifies the source trace of data within the specified block to be copied. Trace numbers run from 1 to number of traces in a file.

dest_expno specifies the experiment number of the destination FID file.

dest_blk_no specifies the destination block to send the copied data.

src_trace_no specifies the destination trace of data within the specified block to be copied. Trace numbers run from 1 to the number of traces in a file.

Examples rftrace(1,1,2,1,nv) copies and reverses trace 1 from block 1 of the current experiment to trace nv of block 1 of experiment 2.

See also *User Programming*

Related

mfblk	Move FID block (C)
mfclose	Memory map close FID file (C)
mfdata	Move FID data (C)
mfopen	Memory map open FID file (C)
mftrace	Move FID trace (C)
rfblk	Reverse FID block (C)
rfdata	Reverse FID data (C)

rftype Type of rf generation (P)

Description Configuration parameter for type of rf generation on each rf channel. On other systems, the value is set using the Type of RF label in the Spectrometer Configuration window.

Values The values of rftype parallel the rfchtype values. The setting for rftype is 'd' on the entries U+ Direct Synthesis and U+ H1 Only.

`'d'` is the setting for a system with direct synthesis (U+ Direct Synthesis in the Spectrometer Configuration window) or a fixed-frequency proton system (U+ H1 Only in Spectrometer Configuration window).

`'l'` is the setting for a deuterium decoupler channel.

`'c'` is the setting for direct synthesis (Direct Synthesis in the Spectrometer Configuration window).

`'b'` is the setting for broadband (Broadband in the Spectrometer Configuration window).

`'a'` is the setting for fixed frequency (Fixed Frequency in the Spectrometer Configuration window).

`'m'` is the setting for imaging modulator (SIS Modulator in the Spectrometer Configuration window).

See also *VnmrJ Installation and Administration*

Related `config` Display current configuration and possibly change it (M)
`rfchtype` Type of rf channel (P)

`rfwg` RF waveform generator (P)

Description Configuration parameter for whether a waveform generator board is present or not on the current rf channel. The value for each channel is set using the Waveform Generator label in the Spectrometer Configuration window.

Values `'n'` is setting for no waveform generator board on the channel (Not Present choice in Spectrometer Configuration window).

`'y'` is setting for a waveform generation board on the channel (Present choice in Spectrometer Configuration window).

See also *VnmrJ Installation and Administration*

Related `config` Display current configuration and possibly change it (M)

`right` Set display limits to right half of screen (C)

Description Sets the horizontal control parameters, `sc` and `wc`, to produce a display (and subsequent plot) in the right portion of the screen (and page). For 2D data, space is left for the scales.

See also *NMR Spectroscopy User Guide*

Related `center` Set display limits for center of screen (C)
`full` Set display limits for a full screen (C)
`fullt` Set display limits for full screen with room for traces (C)
`left` Set display limits for left half of screen (C)

`sc`	Start of chart (P)
`wc`	Width of chart (P)

`rights` Determine an operator's specified right (C)

Applicability Walkup

Syntax `rights('right'<,'errval'>)<:$ret>`

Description The rights program queries the rights database to determine if the current operator has the specified right. This command is used by the interface designer to determine if and how certain options are presented. An operator does not typically use this command. The system administrator sets (restricts) the rights for an operator using VnmrJ administrator interface. By default, the rights command grants any requested right. Rights requested that are not in the rights database are granted. Granting a right means that the rights program returns a 1 to the calling macro.

Arguments `right` — a specific operator right, not case sensitive.

- 1 is returned by the command if the specified `right` is granted or the `right` is not in the rights data base
- 0 is default value returned by the command if the right is both in the database and the operator does not have the specified right.

`errval` — optional argument specifying return value if a right is both in the database and the operator does not have the specified right.

`$ret` — variable holding the return value from the `right` command.

Examples `rights('prioritySample',-1):$ok`

Sets `$ok` to -1 if the `prioritySample` right is not granted. A value of 1 is returned if the `prioritySample` is granted. Returning either a 0 or -1 if a right is not granted lets the interface designer choose to show or gray out a control.

See also *VnmrJ Installation and Administration* and *VnmrJ Walkup* manuals.

`rinput` Input data for a regression analysis (M)

Description Formats data for regression analysis and places the data into the file `regression.inp`. The program is interactive. If a `regression.inp` already exists, `rinput` starts by asking if you want to overwrite the file. Type *y* and press the Return key. It then asks for an x-axis title and a y-axis title. Enter the titles as asked (for no title, simply press Return). Next, `rinput` asks you to input the data in pairs. Separate each pair of values with a blank and press Return after the second value. At the end of the data set, press Return in response to the request for data. If you have another data set, type *y* and press Return to the question and then type in the data when it is asked for.

See also	*NMR Spectroscopy User Guide; User Programming*	
Related	`expl`	Display exponential or polynomial curves (C)
	`poly0`	Find mean of data in the file *regression.inp* (C)

`rl` Set reference line in directly detected dimension (M)

Syntax `rl<(frequency)>`

Description Sets the direct dimension reference line, taking into account any frequency scaling with the `scalesw` parameter.

Arguments `frequency` is a value, in Hz, to assign to the reference line. The default is the cursor position `cr`. To enter the value in ppm, add a `p` suffix.

Examples
```
rl
rl(0)
rl(7.2p)
```

See also *NMR Spectroscopy User Guide*

Related

`cr`	Current cursor position in directly detected dimension (P)
`crl`	Clear ref. line in directly detected dimension (C)
`reffrq`	Reference frequency of the reference line (P)
`rl1`	Set ref. line in 1st indirectly detected dimension (M)
`rl2`	Set ref. line in 2nd indirectly detected dimension (M)
`scalesw`	Scale spectral width in directly detected dimension (P)

`rl1` Set reference line in 1st indirectly detected dimension (M)

Syntax `rl1<(frequency)>`

Description Sets the first indirect dimension reference line, taking into account any frequency scaling with the `scalesw1` parameter.

Arguments `frequency` is a value, in Hz, to assign to the reference line. The default is the cursor position `cr1`. You can enter the suffixes `p`, `d`, or `k` to mean ppm, decoupler ppm, and kilo, respectively. These suffixes are exactly equivalent to using `*sfrq`, `*dfrq`, and `*1000`. Thus, if you are doing a 2D experiment in which the indirect axis is determined by the decoupler channel, you might enter, for example, `rl1(10d)`, which is equivalent to `rl1(10*dfrq)`.

Examples
```
rl1
rl1(0)
rl1(7.2p)
```

See also *NMR Spectroscopy User Guide*

Related

`cr1`	Cursor position in 1st indirectly detected dimension (P)
`crl1`	Clear ref. line in 1st indirectly detected dimension (M)
`dfrq`	Transmitter frequency of first decoupler (P)

refpos2	Position of reference frequency in 2nd indirect dimension (P)
rl	Set ref. line in directly detected dimension (M)
rl2	Set ref. line in 2nd indirectly detected dimension (M)
scalesw1	Scale spectral width in 1st indirectly detected dimension (P)
sfrq	Transmitter frequency of observe nucleus (P)

rl2 Set reference line in 2nd indirectly detected dimension (M)

Syntax `rl2<(frequency)>`

Description Sets the second indirect dimension reference line, taking into account any frequency scaling with the scalesw2 parameter.

Arguments `frequency` is a value, in Hz, to assign to the reference line. The default is the cursor position cr2. You can enter the suffixes `p`, `d`, or `k` to mean ppm, decoupler ppm, and kilo, respectively. These suffixes are exactly equivalent to using *sfrq, *dfrq, and *1000. Because there is no suffix for the second decoupler (i.e., the third channel), to reference the third axis using `rl2` you might enter (e.g., `rl2(45*dfrq2)`).

Examples

```
rl2
rl2(0)
rl2(7.2p)
```

See also *NMR Spectroscopy User Guide*

Related

cr2	Cursor position in 2nd indirectly detected dimension (P)
crl	Clear ref. line in directly detected dimension (C)
crl1	Clear ref. line in 1st indirectly detected dimension (C)
crl2	Clear ref. line in 2nd indirectly detected dimension (C)
dfrq	Transmitter frequency of first decoupler (P)
dfrq2	Transmitter frequency of second decoupler (P)
rl	Set ref. line in directly detected dimension (M)
rl1	Set ref. line in 1st indirectly detected dimension (M)
scalesw2	Scale spectral width in 2nd indirectly detected dimension (P)
sfrq	Transmitter frequency of observe nucleus (P)

rm Delete file (C)

Syntax `rm(file1<,file2,...>)`

Description Removes one or more files from the file system, functioning like the UNIX command of the same name. Because it allows wildcard characters (* and ?) in the command argument and recursive file deletion with the `-r` option, `rm` is very powerful. But it can be quite dangerous—without warning important files can be inadvertently deleted, even by experienced users. **Using `rm` to delete files in VnmrJ is not recommended**. The `delete` command is provided as a safer alternative.

Arguments `file1,file2,...` are names of files to delete.

See also *NMR Spectroscopy User Guide*

Related		
	`delete`	Delete a file, parameter directory, or FID directory (C)
	`delexp`	Delete an experiment (C)
	`exists`	Determine if a parameter, file, or macro exists (C)
	`mv`	Move and/or rename a file (C)
	`rename`	Move and/or rename a file (C)

`rmdir` Remove directory (C)

Syntax `rmdir(directory)`

Description Removes one or more empty directories (i.e., directories without files).

Arguments `directory` is the name of the directory to be removed.

Examples `rmdir('/home/dan/temp')`

See also *NMR Spectroscopy User Guide*

Related		
	`delete`	Delete a file, parameter directory, or FID directory (C)
	`dir`	List files in current directory (C)
	`lf`	List files in current directory (C)
	`ls`	List files in current directory (C)
	`mkdir`	Create new directory (C)

`rmsAddData` Add transformed data files with weighting (U)

Applicability Systems with multiple receivers.

Description This command is not normally executed directly by the user.

`Roesy` Convert the parameter to a ROESY experiment (M)

Description Convert the parameter to a rotating frame Overhauser effect spectroscopy (ROESY) experiment.

Roesy1d — Convert the parameter set to a Roesy1d experiment (M)

Description: Convert the parameter set to a 1D rotating frame Overhauser effect spectroscopy (Roesy1D) experiment.

See also: *NMR Spectroscopy User Guide*

Related:

Proton	Set up parameters for ^{1}H experiment (M).
sel1d	Selective 1D protocols to set up (M).

rof1 — Receiver gating time preceding pulse (P)

Description: Sets the period of time in most pulse sequences when the receiver is gated off before each pulse. This allows the amplifier to fully turn on before the start of the pulse. Systems are configured with linear amplifiers that are normally “blanked” to give the best possible signal-to-noise (i.e., the amplifiers are turned off when the receiver is turned on). The $^{1}H/^{19}F$ amplifiers have a short turn-on time, usually 1 to 5 µs following the removal of blanking by turning the receiver off. The low-frequency amplifier modules have a longer turn-on time, about 40 to 60 µs.

Values: Typically 2-5 microseconds.

See also: *NMR Spectroscopy User Guide*

Related:

rof2	Receiver gating time following pulse (P)

rof2 — Receiver gating time following pulse (P)

Description: Sets the time after the final pulse in each pulse sequence that the receiver is gated off before acquisition begins. If “pulse breakthrough” effects are seen (a spike in the beginning of the FID), increasing `rof2` can reduce or eliminate the problem, particularly for low-frequency nuclei.

Values: Typically 10 microseconds.

See also: *NMR Spectroscopy User Guide*

Related:

rof1	Receiver gating time preceding pulse (P)
setlp0	Set parameters for zero linear phase (M)

rof3 — Receiver gating time following T/R switch (P)

Description: Sets the time when the receiver is gated on following the T/R switch during the pulse. This allows for the elimination of pulse artifacts during the acquisition period.

rotate Rotate 2D data (C)

Syntax `rotate<(number_degrees)>`

Description Rotates a 2D spectrum. Both complex and hypercomplex 2D data will work.

Arguments `number_degrees` is the amount of counter-clockwise rotation, in degrees. The default is 45.

See also *NMR Spectroscopy User Guide*

Related
`foldcc` Fold INADEQUATE data about 2-quantum axis (C)
`foldj` Fold J-resolved 2D spectrum about *f1=0* axis (C)
`foldt` Fold COSY-like spectrum along diagonal axis (C)

rotorsync Rotor synchronization (P)

Applicability Systems with the solids rotor synchronization module.

Description Configuration parameter that identifies if the system has the optional solids rotor synchronization module. The value of `rotorsync` is set using the Rotor Synchronization label in the Spectrometer Configuration window. Rotor synchronization requires either the Acquisition Controller board (Part No. 969204) or the Pulse Sequence Controller board (Part No. 992560) in the system.

Values `1` is setting that system has solids rotor synchronization (Present choice in the Spectrometer Configuration window).

`0` is setting that system does not have solid rotor synchronization (Not Present choice in the Spectrometer Configuration window).

See also *VnmrJ Installation and Administration*

Related
`config` Display current configuration and possibly change it (M)

rp Zero-order phase in directly detected dimension (P)

Description Specifies the right phase-correction angles along the directly detected dimension according to

absorption spectrum(ω) =
real channel(ω) * *cos θ* + *imaginary channel(ω)* **sin θ*

where the phase angle θ is a function of frequency:

$\theta = \mathtt{rp} + (\omega - \omega_o)/\mathtt{sw} * \mathtt{lp}$

ω_o is defined as the right end of the spectrum. This dimension is referred to as the f_2 dimension in 2D data sets, f_3 dimension in 3D data sets, and so on.

Values –360 to +360, in degrees.

See also *NMR Spectroscopy User Guide*

Related

`aph`	Automatic phase adjustment of spectra (C)
`aph0`	Automatic phase of zero-order term (C)
`lp`	First-order phase in directly detected dimension (P)
`rp1`	Zero-order phase in 1st indirectly detected dimension (P)
`rp2`	Zero-order phase in 2nd indirectly detected dimension (P)
`setlp0`	Set parameters for zero linear phase (M)

rp1 — Zero-order phase in 1st indirectly detected dimension (P)

Description: Specifies the right phase parameter along the first indirectly detected dimension, in degrees, for the f_1 dimension of a multidimensional data set during the process of phase-sensitive 2D transformation.

See also *NMR Spectroscopy User Guide*

Related

`lp1`	First-order phase in 1st indirectly detected dimension (P)
`rp`	Zero-order phase in directly detected dimension (P)
`rp2`	Zero-order phase in 2nd indirectly detected dimension (P)

rp2 — Zero-order phase in 2nd indirectly detected dimension (P)

Description: Controls the zero-order phase constant along the second indirectly detected dimension during a `ds`, `dconi`, or equivalent display operation on the 2D data or a 1D trace therein. This dimension is often referred to as the f_2 dimension.

See also *NMR Spectroscopy User Guide*

Related

`dconi`	Interactive 2D contour display (C)
`ds`	Display a spectrum (C)
`lp2`	First-order phase in 2nd indirectly detected dimension (P)
`rp`	Zero order phase in directly detected dimension (P)

rt — Retrieve FIDs (M)

Syntax: `rt<(file<,'nolog'>)>`

Description: Retrieves FIDs from a file into the current experiment.

The `rt` macro does not copy the FID into the experiment. Instead, it links access to the original FID from the experiment. Most of the time, this behavior is desired, because the FID file is seldom changed. By making a link, disk space is also conserved. However, if the FID file in the experiment is written to, the data in the original file is also written to. It is best to make a copy of a FID file before altering it. The makefid command alters the FID file. The manual entry for makefid gives details on how to make a copy of the FID.

As another somewhat subtle point, because the FID in the experiment is a link to another .fid file, if that .fid file is removed, the link from the experiment may be gone. If you expect the FID in the experiment to be there, even if you delete the .fid file from where it was retrieved using `rt`, you should explicitly copy the file into the experiment.

Arguments `file` is the name of the file that, with the suffix `.fid` added, contains the FIDs to be retrieved. The default is that the system prompts for the name (in that case, the name can be given without single quotes). If `file.fid` does not exist and `file.par` does, `rt` retrieves the parameters from `file.par`.

`'nolog'` is a keyword specifying that the log file is not to be retrieved.

Examples
```
rt
rt('/vnmr/fidlib/fid1d')
```

See also *NMR Spectroscopy User Guide*

Related		
	fixpar	Correct parameter characteristics in experiment (M)
	makefid	Make a FID element using numeric text input (C)
	rtp	Retrieve parameters (M)
	rtv	Retrieve individual parameters (C)
	svf	Save FIDs in current experiment (M)

rtcmx Return Spinsight data into current experiment (C)

Syntax `rtcmx<(file)>`

Description Retrieves Spinsight data into the current experiment.

Arguments `file` is the name of the file. The default is that the macro prompts for the file name.

Alternate: Load button in the files program.

Examples
```
rtcmx
rtcmx('redor.data')
```

See also *NMR Spectroscopy User Guide*

Related		
	files	Interactively handle files (C)

rtp Retrieve parameters (M)

Syntax `rtp<(file)>`

Description Retrieves parameters from a file into the current experiment.

Arguments `file` is the name of the file that, with the suffix `.par` added, contains the parameters to be retrieved;. The default is that the system prompts for the name (in that case, the name can be given without single quotes). If `file.par` does not exist and `file.fid` does, `rtp` retrieves the parameters only from `file.fid`.

Examples `rtp`
`rtp('/vnmr/stdpar/P31')`

See also *NMR Spectroscopy User Guide*

Related		
	fixpar	Correct parameter characteristics in experiment (M)
	rt	Retrieve FIDs (M)
	rtv	Retrieve individual parameters (C)
	svp	Save parameters from current experiment (M)

rts Retrieve shim coil settings (C)

Syntax `rts(file)<:status>`

Description Locates a preexisting file of shim settings and copies the settings into the current parameter set of the current experiment and sets `load='y'` to facilitate subsequent loading of shims with `su` (or related commands or macros). If the shim file is not found, `rts` displays the file names it tried.

The `rts` command returns shims from a `.fid` file or a `.par` file, selecting the shim parameters from the parameters stored there.

Arguments `file` — the name of a file containing the shim coil settings to be retrieved. If the file name is an absolute path, `rts` uses it with no modifications. Otherwise, `rts` searches the applications directories.

`status` — the return variable with one of the following values after `rts` finishes searching for the shim coil settings file:

- 0 indicates that `rts` failed to find requested file.
- 1 indicates that `rts` found the requested file, either as an absolute path or in the `shims` directory of the first application directory.
- >=2 indicates that `rts` found the requested file in `shims` subdirectory of the second, third, or later application directory.

Examples `rts('acetone')`
`rts('bb10mm'):r1`

See also *NMR Spectroscopy User Guide*

Related		
	load	Load status of displayed shims (P)
	su	Submit a setup experiment to acquisition (M)
	svs	Save shim coil settings (C)

rttmp Retrieve experiment data from experiment subfile (M)

Syntax `rttmp(file)`

Description Retrieves experiment data—parameters, FID, and transformed spectrum—from the file specified in a subdirectory inside curexp+'/subexp'.

Arguments file is the name of the subfile from which to retrieve the experiment data.

Examples rttmp('H1')
rttmp('cosy')

See also *NMR Spectroscopy User Guide*

Related
Related	cptmp	Copy experiment data into experiment subfile (M)
	curexp	Current experiment directory (P)
	svtmp	Move experiment data into experiment subfile (M)

rtv Retrieve individual parameters (C)

Syntax rtv<(file,par1<,index1<,par2,index2...>>)><:val>

rtv('parmaster','noabort','parameter'):$pm

Description Retrieves one or more parameters from a parameter file. The file might have been made with svf or svp or sd commands, or it might be from another experiment. If no return argument is added, the parameters are copied into the experiment's current tree. If the parameter does not already exist in the current tree, it is created. If the returned parameter is an array, the entire array is returned.

rtv returns values into the macro if a return argument is added. This form of rtv command, in which values are passed only to macro variables, avoids the creation of additional parameters in the experiment's current tree.

Arguments file — name of the directory or a parameter file. If the supplied value for file is a directory (with or without the .fid or .par extension), the parameters are retrieved from the procpar file in that directory. If the supplied value does not correspond to a directory but rather is a parameter file, that file is used. The default is that rtv prompts for a file name. In that case, the file name can be given without single quotes.

par1,index1,par2,index2,... — name and array index of one or more parameters to be retrieved. The default for each array index argument is the first index. Including the array index for a parameter is only useful when returning values to the macro through a return argument.

val — return argument for values to return to the macro. If the requested parameter do not exist in the parameter file, rtv will abort.

noabort — keyword option must follow the 'parmaster' keyword and precede the parameter argument. This option applies to a single parameter. Command does not abort if the requested parameter does not exist.

parmaster — filename of the parameter set.

`parameter` – the parameter name.

Executing `rtv` without macro return values causes the `fixpar` macro run. The macro `fixpar` is not executed if return values are requested. rtv will prompt for a file name if the command is executed without an argument. The filename given in response to the prompt does not require single quotes.

In LC-NMR, `rt` will retrieve the `lcdata` (and `drunlog`) files if these files were saved along with the NMR data by using `svf`.

Examples
```
rtv
rtv('/vnmr/parlib/cosy.par','phase')
rtv('/vnmr/parlib/cosy.par','noabort','phase')
```

See also *NMR Spectroscopy User Guide* and *User Programming* manuals

Related
rt	Retrieve FIDs (M)
rtp	Retrieve parameters (M)
sd	Set first decoupler frequency to cursor position (M)
svf	Save FIDs in current experiment (M)
svp	Save parameters from current experiment (M)

rtx — Retrieve parameters based on rtx rules (C)

Syntax `rtx(filename <,tree <, keyword1 <, keyword2 >>>)`

Description The rtx command retrieves parameters from filename, based on the setting of the P_LOCK protection bit and using the rules below.

Arguments `tree` is `'current'`, `'processed'`, `'global'`, or `'systemglobal'`.

`keyword1` may be `'keep'` or `'rt'`. The default is `'keep'`.

`keyword2` may be `'clear'` or `'noclear'`. The default is `'clear'`.

`keyword2` determines if the P_LOCK bit is cleared after `rtx` is executed.

Truth table for `rtx`.

Status of P_LOCK bit in current exp	Status of P_LOCK bit in filename	keyword1	result
on	on	keep or rt	do not rt
on	off	keep or rt	do not rt
off	on	keep or rt	do rt
off	off	keep	do not rt
off	off	rt	do rt
<no parameter>	on	keep or rt	do rt
<no parameter>	off	keep	do not rt
<no parameter>	off	rt	do rt

See also *NMR Spectroscopy User Guide*

Related
execpars	Set up the exec parameters (M)
rtp	Retrieve parameters (M)

S

s	Save display parameters as a set (M)
s(n)	Save display parameters (C)
s2pul	Set up parameters for standard two-pulse sequence (M)
sa	Stop acquisition (C)
sample	Submit change sample, Autoshim experiment to acquisition (M)
samplechange	Automation utility
samplename	Sample name (P)
save	Save data (M)
savefid	Save fid
savefile	Base file name for saving files (P)
saveglobal	Save selected parameters from global tree (P)
savesampglobal	Saves Sample Global Parameters
sb	Sinebell constant in directly detected dimension (P)
sb1	Sinebell constant in 1st indirectly detected dimension (P)
sb2	Sinebell constant in 2nd indirectly detected dimension (P)
sbs	Sinebell shift in directly detected dimension (P)
sbs1	Sinebell shift in 1st indirectly detected dimension (P)
sbs2	Sinebell shift in 2nd indirectly detected dimension (P)
sc	Start of chart (P)
sc2	Start of chart in second direction (P)
scalelimits	Set limits for scales in regression (M)
scalesw	Set scaling factor for multipulse experiments (M)
scalesw	Scale spectral width in directly detected dimension (P)
scalesw1	Set f_1 scaling factor for 2D multipulse experiments (M)
scalesw1	Scale spectral width in 1st indirectly detected dimension (P)

scalesw2	Scale spectral width in 2nd indirectly detected dimension (P)
schedulerhelp	Proshim Maintenance Scheduler help(C)
sd	Set first decoupler frequency to cursor position (M)
sd2	Set second decoupler frequency to cursor position (M)
sd3	Set third decoupler frequency to cursor position (M)
sda	Set first decoupler frequency array (M)
sd2a	Set second decoupler frequency array (M)
sd3a	Set third decoupler frequency array (M)
sdp	Show diffusion projection (M)
sel1d	Apptype macro for Selective 1D experiments
select	Select spectrum, FID, trace, or 2D plane without display (C)
selex	Defines excitation band (M)
selexcit	Set up PFG selective excitation pulse sequence (M)
selexHT	Set up a selective Hadamard experiment (M)
send2vnmr	Send a command to VnmrJ (U)
seqfil	Pulse sequence name (P)
seqgen	Initiate compilation of user's pulse sequence (M,U)
seqgenupdate	Update compilation of user's pulse sequence
serverport	Returns the VnmrJ network listening port value (C)
set2D	General setup for 2D experiments (M)
set2d	General setup for 2D experiments (M)
set3dproc	Set 3D processing (C)
setallshims	Set all shims into hardware (M)
setcolor	Set colors for graphics window and for plotters (C)
setdecpars	Set decoupler parameter values from probe file (M)
setDECpars	Sets Decoupler Parameters
setdec2pars	Set decoupler 2 parameter values from probe file (M)
setdgroup	Set the Dgroup of a parameter in a tree (C)
setenumeral	Set values of a string parameter in a tree (C)
setether	Connect or reconnect host computer to Ethernet (U)
setexport	Set parameter bits for use with protocols (M)
setfrq	Set frequency of rf channels (C)

setgauss	Set a Gaussian fraction for lineshape (M)
setgcal	Set the gradient calibration constant (M)
setgcoil	Assign sysgcoil configuration parameter (M)
setgrid	Divide graphics window into rows and columns (C)
setgroup	Set group of a parameter in a tree (C)
sethtfrq1	Set a Hadamard frequency list from a line list ((M)
sethw	Set values for hardware in acquisition system (C)
sethwshim	Set values for hardware in acquisition system (C)
setint	Set value of an integral (M)
setlimit	Set limits of a parameter in a tree (C)
setlk	Set up lock parameters (M)
setlockfreq	Set lock frequency (M)
setLP	Set up linear prediction in the direct dimension (M)
setLP1	Set F1 linear prediction parameters (M)
setlp0	Set parameters for zero linear phase (M)
setnoether	Disconnect host computer from Ethernet (U)
setobspars	Sets Observe Parameters
setoffset	Calculate offset frequency for given nucleus and ppm (M)
setparams	Write parameter to current probe file (M)
setpen	Set maximum number of HP plotter pens (M)
setplotdev	Return characteristics of a named plotter (C)
setpower	Set power and pulsewidth for a given γB1 value (M)
setprotect	Set protection mode of a parameter (C)
setpw180ad	Creates and sets observe adiabatic pulse shapes (M)
setpwx180ad	Creates and sets decoupler adiabatic pulse shapes (M)
setrc	Set receiver constants (M)
setref	Set frequency referencing (M)
setref1	Set freq. referencing for 1st indirectly detected dimension (M)
setref2	Set freq. referencing for 2nd indirect detected dimension (M)
setscout	Set up a scout run (M)
setssfilter	Set sslsfrq to the frequencies of each suppressed solvents (M)
setsw	Set spectral width (M)

setsw1	Set spectral width in evolution dimension (M)
setsw2	Set spectral width in 2nd evolution dimension (M)
setselfrqc	Set selective frequency and width (M)
setselinv	Set up selective inversion (M)
settcldefault	Select default display templates for pulse sequence (M)
settune	Opens the Auto Tune Setup dialog (M)
settype	Change type of a parameter (C)
setup	Set up parameters for basic experiments (M)
setup_dosy	Set up gradient levels for DOSY experiments (M)
setuserpsg	Creates/initializes user PSG directory
setvalue	Set value of any parameter in a tree (C)
setwave	Write a wave definition string into Pbox.inp file (M)
setwell	Adjust the label of the "t1" axis for VAST contour maps (M)
setwin	Activate selected window (C)
sf	Start of FID (P)
sf1	Start of interferogram in 1st indirectly detected dimension (P)
sf2	Start of interferogram in 2nd indirectly detected dimension (P)
sfrq	Transmitter frequency of observe nucleus (P)
sh2pul	Set up for a shaped observe excitation sequence (M)
shdec	Set up for shaped observe excitation sequence (M)
shell	Start a UNIX shell (C)
shelli	Start an interactive UNIX shell (C)
shim	Submit an Autoshim experiment to acquisition (C)
shimmult	Multiple the shim dacs of the current shimset
shimnames	Returns shim names
shimset	Type of shim set (P)
showconfig	Show system configuration settings (M)
showconsole	Show console configuration parameters (U)
showdosy	Show DOSY Plot (M)
showdosyfit	Plots the experimental signal attenuation, fitted attenuation and residual for one peak from a 2D or 3D DOSY experiment (M)
showdosyresidual	Plots the residual for one peak from a 2D or 3D DOSY experiment
showfit	Display numerical results of deconvolution (M)

showgradfit	Plots the experimental gradient variation with position and the power series fit in non-uniform gradient calibration (M)
showloginbox	Shows operator login dialog (M)
shownugfit	Plots the logarithm of the calculated diffusional attenuation and of the power series fit in non-uniform gradient calibration (M)
shownumx	x position counting from bottom left of every spectrum (P)
shownumy	y position counting from bottom left of every spectrum (P)
showoriginal	Restore first 2D spectrum in 3D DOSY experiment (M)
showplotter	Show list of currently defined plotters and printers (M)
showplotq	Display plot jobs in plot queue (M)
showprintq	Display print jobs in print queue (M)
showprotunegui	Show the graphical interface while tuning (P)
showrfmon	Show RF Monitor Button in Hardware Bar (P)
showsampglobal	Shows sample global parameters
showstat	Display information about status of acquisition (M,U)
sim	Sample in magnet (For systems equipped with a robot)
sin	Find sine value of an angle (C)
sine	Find values for a sine window function (M)
sinebell	Select default parameters for sinebell weighting (M)
sinesq	Find values for a sine-squared window function (M)
size	Returns the number of elements in an arrayed parameter (O)
slfreq	Measured line frequencies (P)
slw	Spin simulation linewidth (P)
smaxf	Maximum frequency of any transition (P)
sminf	Minimum frequency of any transition (P)
smsport	Sample Management System serial port connection (P)
sn	Signal-to-noise ratio (P)
solppm	Return ppm and peak width of solvent resonances (M)
solvent	Lock solvent (P)
solvinfo	Retrieve information from solvent table (C)
sort	Sort real values of a parameter (M)
sp	Start of plot in directly detected dimension (P)
sp1	Start of plot in 1st indirectly detected dimension (P)

sp2	Start of plot in 2nd indirectly detected dimension (P)
spadd	Add current spectrum to add/subtract experiment (C)
spcfrq	Display frequencies of rf channels (M)
specdc3d	3D spectral drift correction (P)
spin	Submit a spin setup experiment to acquisition (C)
spin	Sample spin rate (P)
spincad	Run SpinCAD program (C)
spingen	Compile SpinCAD pulse sequence *C)
spinll	Set up a slfreq array (M)
spinner	Open the Spinner Control window (C)
spins	Perform spin simulation calculation (C)
split	Split difference between two cursors (M)
spintype	Spinner Type ((P)
splmodprepare	Used by the dosy macro to prepare data for the program SPLMOD (C)
splmodread	Used by the dosy macro to convert the output of the SPLMOD program into a form suitable for ddif (C)
spmax	Take the maximum of two spectra (C)
spmin	Take minimum of two spectra in add/subtract experiment (C)
spsm	Enter spin system (M)
spsub	Subtract current spectrum from add/subtract experiment (C)
sqcosine	Set up unshifted cosine-squared window function (M)
sqdir	Study queue directory (P)
sqend	End a study queue (M)
sqexp	Load experiment from protocol (M)
sqfilemenu	Study queue file menu commands (M)
sqLog	Records specific events from a study queue
sqmode	Study queue mode (P)
sqname	Study queue parameter template (P)
sqpars	Create study queue parameters for imaging (M)
sqprotocol	Macro to create protocols (M)
sqreset	Reset study queue parameters for imaging (M)
sqrt	Return square root of a real number (O)
sqsavestudy	Macro to save study parameters for imaging (M)

sqsinebell	Set up unshifted sinebell-squared window function (M)
srate	Spinning rate for magic angle spinning (P)
sread	Read converted data into VnmrJ (C)
srof2	Calculate exact rof2 value for Cold Probes (M)
ss	Steady-state transients (P)
ssecho	Set up solid-state echo pulse sequence (M)
ssecho1	Set up parameters for SSECHO1 pulse sequence (M)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
sslsfrq	Center of solvent-suppressed region of spectrum (P)
ssntaps	Number of coefficients in digital filter (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
stack	Stacking mode for processing and plotting arrayed spectra (M)
stackmode	Stacking control for processing arrayed 1D spectra (P)
startq	Start a chained study queue (M)
status	Display status of sample changer (C,U)
std1d	Apptype macro for Standard 1D experiments (M)
stdshm	Interactively create a method string for autoshimming (M)
sth	Minimum intensity threshold (P)
string	Create a string variable (C)
string2array	Formats a String Variable into an Array
strstr	Find position of one string in another
strsv2array	Formats a String Separated Variable into an Array
strtext	Starting point for LP data extension in np dimension (P)
strtext1	Starting point for LP data extension in ni dimension (P)
strtext2	Starting point for LP data extension in ni2 dimension (P)
strtlp	Starting point for LP calculation in np dimension (P)
strtlp1	Starting point for LP calculation in ni dimension (P)
strtlp2	Starting point for LP calculation in ni2 dimension (P)
studyid	Study identification (P)
studypar	Study parameters (P)
studystatus	Study status (P)
studytime	Study time (P)

su	Submit a setup experiment to acquisition (M)
sub	Subtract current FID from add/subtract experiment (C)
substr	Select a substring from a string (C)
suselfrq	Select peak, continue selective excitation experiment (M)
svdat	Save data (C)
svf	Save FIDs in current experiment (M)
svfdf	Save FID data in FDF format (M)
svfdir	Directory for non-study data (P)
svfj	Save FID in JCAMP-DX format (M)
svfname	Filename parameter template for non-study data ((P)
svfname	Create path for data storage (C)
svimg	Generate and Save images as FDF files (macro)
svllj	Save peak listing in JCAMP-DX X,Y or X,Y,M format (M)
svlsj	Save large dynamic range spectrum in JCAMP-DX format (M)
svp	Save parameters from current experiment (M)
svpdp	
svs	Save shim coil settings (C)
svs	Spin simulation vertical scale (P)
svsis	Generate and Save images as FDF files (macro)
svsj	Save spectrum in JCAMP-DX format (M)
svtmp	Move experiment data into experiment subfile (M)
svxyj	Save spectrum in JCAMP-DX X,Y format (M)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 1st indirectly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)
sysgcoil	System gradient coil (P)
system	System type (P)
systemdir	VnmrJ system directory (P)

s — Save display parameters as a set (M)

Syntax (1) sset_number
(2) s(set_number)

Description Saves a copy of the current values of all display parameters. The set is data-independent because the parameters that govern a display (sp, wp, vs, etc.) are saved but no data is saved.

Arguments set_number is number of the display parameter set to be saved.

Examples s2
s(3)

See also *NMR Spectroscopy User Guide*

Related
fr	Full recall of display parameter set (M)
r	Recall display parameter set (M)

s(n) — Save display parameters (C)

Applicability All

Syntax s(n<,noupdate>)

Description Saves a copy of the current values of all display parameters as display parameter set n in the current experiment

noupdate as second argument prevents the automatic update of interactive programs.

Arguments n=1 to 9

Related
fr(n)	Recall all the parameters of the specified display parameter set (C)
r(n)	Recalls limited number of display parameters)

s2pul — Set up parameters for standard two-pulse sequence (M)

Description Converts the current experiment to an experiment suitable for the standard two-pulse sequence (S2PUL).

See also *NMR Spectroscopy User Guide*

sa — Stop acquisition (C)

Syntax sa<(option|number)>

Description Stops an experiment that has been submitted to acquisition. If experiment is active, it is stopped. Data is retained. sa applies to the experiment that you are joined to at the time the sa command is entered. Thus, if experiment 1 is active, you must be joined to experiment 1 for sa to stop that acquisition. If you are in experiment 2, entering sa has no effect on experiment 1.

When experiments are queued, the behavior of sa is more complex. If an experiment is active in exp1 and queued in exp2, entering sa from exp1 stops that experiment and immediately begins acquisition on exp2. Entering sa from exp2, on the other hand, removes exp2 from the queue, without affecting the active experiment 1.

Entering sa from an experiment that is not active or queued has no effect.

Arguments option is one of the following:

- 'eos', 'ct', 'scan' are keywords to stop at the next ct.
- 'eob', 'bs' are keywords to stop at the next block size.
- 'eof', 'nt', 'fid' are keywords to stop at the next complete FID.
- 'eoc', 'il' are keywords to stop at next complete il cycle (i.e., the latest block size that has been completed for all FIDs in interleave cycle.

number is an integer number to stop at the next ct, where the value of ct is a multiple of number. This is useful when you want to complete a phasecycle before stopping.

Examples

```
sa
sa('ct')
sa(4)
```

See also *NMR Spectroscopy User Guide*

Related

bs	Block size (P)
ct	Completed transients (P)
il	Interleave arrayed and 2D experiments (P)
nt	Number of transients (P)
ra	Resume acquisition stopped with *sa* command (C)

sample Submit change sample, Autoshim experiment to acquisition (M)

Applicability Systems with a sample changer.

Description Performs the combined operations change, spin, lock, and shim, making it a convenient setup command for a new sample.

See also *NMR Spectroscopy User Guide*

Related

au	Submit experiment to acquisition and process data (C)
change	Submit a change sample experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
lock	Submit an Autolock experiment to acquisition (C)
shim	Submit an Autoshim experiment to acquisition (C)

spin	Submit a spin setup experiment to acquisition (C)
su	Submit a setup experiment to acquisition (M)

sample ChangeAutomation utility

Syntax

Applicability VnmrJ 3.1

Description This is a utility macro to remove the sample from the magnet after an automation queue finishes. It is only available with systems with the 7600-AS or 7510-AS robot systems. The choice to either put a reference sample into the magnet, leave the current sample in the magnet, or remove the current sample from the magnet, is made from the Preferences pop-up window.

samplename Sample name (P)

Description Specifies the name of the sample. It is saved with a liquids study.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

cqsavestudy	Macro to save study queue parameters (M)
notebook	Notebook name (P)
page	Name of page (P)
studypar	Study parameters (P)

save Save data (M)

Description Macro to save data. In a study, it uses sqdir and autoname to construct the data filename. If not in a study, it uses `svfdir` and `svfname` to construct the data filename.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

acquire	Acquire data (M)
autoname	Create path for data storage (C)
autoname	Prefix for automation data file (P)
sqdir	Study queue directory (P)
svfdir	Directory for non-study data (P)
Svfname	Create path for data storage (C)
svfname	Filename parameter template for non-study data ((P)

savefid Save fid

Description This utility saves the data in the current workspace according to the templates in the Preferences/Templates panel.

Syntax `savefid`

Related svf

savefile Base file name for saving files (P)

Applicability Systems with LC-NMR accessory.

Description Contains the base file name using the format `savefile.001`, `savefile.002`, etc., to which a series of FIDs or data sets are saved. If `savefile` does not exist, the parlc macro can create it.

See also *NMR Spectroscopy User Guide*

Related parlc Create LC-NMR parameters (M)

saveglobal Save selected parameters from global tree (P)

Description Saves an array of parameter names from the global or systemglobal tree. Whenever go is executed, the parameters listed are saved in the current tree with an underscore (_) appended. These parameters are copied back into the global tree (without the underscore) whenever processing by `wbs`, `wnt`, `wexp`, or `werr` occurs.

See also *NMR Spectroscopy User Guide*

Related go Submit experiment to acquisition (C)
loc Location of sample in tray (P)

savesampglobalSaves Sample Global Parameters

Description Updates sample global parameters in the study directory from the current workspace.

See also `savesampglobal`

Related getsampglobal, resetsampglobal, savesampglobal, mvsampglobal, showsampglobal

sb Sinebell constant in directly detected dimension (P)

Description Applies a sinebell constant along the directly detected dimension. This dimension is often referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc.

Values A positive value applies a sinebell of the form $\mathtt{sin}\left(\frac{\mathtt{t} \cdot \pi}{2 \cdot \mathtt{sb}}\right)$

A negative value applies a squared sinebell function of form $\mathtt{sin}^2\left(\frac{\mathtt{t} \cdot \pi}{2 \cdot \mathtt{sb}}\right)$

`sb` is given in seconds. Typical value is `sb='n'`.

See also *NMR Spectroscopy User Guide*

Related

sb1	Sinebell constant in 1st indirectly detected dimension (P
sb2	Sinebell constant in 2nd indirectly detected dimension (P)
sbs	Sinebell shift constant in directly detected dimension (P)
sine	Find values for a sine window function (M)
sinebell	Select default parameters for sinebell weighting (M)
sinesq	Find values for a sine squared window function (M)

sb1 Sinebell constant in 1st indirectly detected dimension (P)

Description Applies a sinebell constant along the first indirectly detected dimension. This dimension is often referred to as the f_1 dimension in multidimensional data sets. `sb1` works analogously to the parameter `sb`. The “conventional” parameters, such as `lb` and `gf`, operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

Values A positive value applies a sinebell of the form $\mathtt{sin}\left(\frac{\mathtt{t} \cdot \pi}{2 \cdot \mathtt{sb1}}\right)$

A negative value applies a squared sinebell function of form $\mathtt{sin}^2\left(\frac{\mathtt{t} \cdot \pi}{2 \cdot \mathtt{sb1}}\right)$

`sb1` is given in seconds. Typical value is `sb1='n'`.

See also *NMR Spectroscopy User Guide*

Related

sb	Sinebell constant in the directly detected dimension (P)
sb2	Sinebell constant in 2nd indirectly detected dimension (P)

sb2 Sinebell constant in 2nd indirectly detected dimension (P)

Description Applies a sinebell constant along the second indirectly detected dimension. This dimension is often referred to as the f_2 dimension in multidimensional data sets. `sb2` works analogously to the parameter sb. The value of sb2 can be set with wti on the 2D interferogram data.

Values A positive value applies a sinebell of the form $\mathrm{sin}\left(\frac{t \cdot \pi}{2 \cdot sb2}\right)$

A negative value applies a squared sinebell function of form $\mathrm{sin}^2\left(\frac{t \cdot \pi}{2 \cdot sb2}\right)$

sb2 is given in seconds. Typical value is sb2='n'.

See also *NMR Spectroscopy User Guide*

Related	sb	Sinebell constant in directly detected dimension (P)
	sb1	Sinebell constant in 1st indirectly detected dimension (P)
	wti	Interactive weighting (C)

sbs Sinebell shift in directly detected dimension (P)

Description Working in combination with the parameter sb, sbs allows shifting the origin of the sinebell function along the directly detected dimension. This dimension is often referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc.

Values The origin is shifted according to the formula $\mathrm{sin}\left(\frac{(t - sbs) \cdot \pi}{2 \cdot sb}\right)$

The square of this function is applied if sb is negative. sbs is given in seconds. The typical value is sbs='n'.

See also *NMR Spectroscopy User Guide*

Related	sb	Sinebell constant in directly detected dimension (P)
	sbs1	Sinebell shift in 1st indirectly detected dimension (P)
	sbs2	Sinebell shift in 2nd indirectly detected dimension (P)
	sine	Find values for a sine window function (M)
	sinesq	Find values for a sine squared window function (M)

sbs1 Sinebell shift in 1st indirectly detected dimension (P)

Description Working in combination with the parameter sb1, sbs1 allows shifting the origin of the sinebell function along the first indirectly detected dimension. This dimension is often referred to as the f_1 dimension in multidimensional data sets. sbs1 works analogously to parameter sbs. The “conventional” parameters, such as lb and gf, operate on the detected FIDs, while this “2D” parameter is used during processing of the interferograms.

Values The origin is shifted according to the form $\mathrm{sin}\left(\frac{(t - sbs1) \cdot \pi}{2 \cdot sb1}\right)$

The square of this function is applied if sb1 is negative. sbs1 is given in seconds. The typical value is sbs1='n'.

See also *NMR Spectroscopy User Guide*

Related	sb1	Sinebell constant in 1st indirectly detected dimension (P)
	sbs	Sinebell shift constant in directly detected dimension (P)
	sb2	Sinebell constant in 2nd indirectly detected dimension (P)

sbs2 Sinebell shift in 2nd indirectly detected dimension (P)

Description: Working in combination with the parameter sb2, sbs2 allows shifting the origin of the sinebell function along the second indirectly detected dimension. This dimension is often referred to as the f_2 dimension in multidimensional data sets. sbs2 works analogously to parameter sbs. sbs2 can be set with wti on the 2D interferogram data.

Values: The origin is shifted according to the formula $\mathtt{sin}\left(\frac{(\mathtt{t}-\mathtt{sbs2})\cdot\pi}{2\cdot\mathtt{sb2}}\right)$

The square of this function is applied if sb2 is negative. sbs2 is given in seconds. The typical value is sbs2='n'.

See also *NMR Spectroscopy User Guide*

Related	sbs	Sinebell shift constant in directly detected dimension (P)
	sb2	Sinebell constant in 2nd indirectly detected dimension (P)
	wti	Interactive weighting (C)

sc Start of chart (P)

Description: Positions of the start of the plotting position (the “chart”) with respect to the right edge of the plotter.

Values: 0 to wcmax, in mm

See also *NMR Spectroscopy User Guide*

Related	sc2	Start of chart in second direction (P)
	wc	Width of chart (P)
	wcmax	Maximum width of chart (P)

sc2 Start of chart in second direction (P)

Description: Controls the start of plotting position of the second axis (or *y* axis) of a 2D contour plot. The parameter wc2 controls the width of the chart.

Values: 0 to wc2max, in mm.

See also *NMR Spectroscopy User Guide*

Related	sc	Start of chart (P
	wc2	Width of chart in second direction (P)
	wc2max	Maximum width of chart in second direction (P)

scalelimits Set limits for scales in regression (M)

Syntax	`scalelimits(x_start,x_end,y_start,y_end)`
Description	Causes the command `expl`, which is used by regression to display data, to use typed-in scale limits. The limits are retained as long as an `expl` display is retained.
Arguments	`x_start,x_end,y_start,y_end` are *x*-axis and *y*-axis starting and ending limits. The default is that `scalelimits` prompts for the limits.
See also	*NMR Spectroscopy User Guide, User Programming*

Related

`autoscale`	Resume autoscaling after limits set by `scalelimits` (M)
`expl`	Display exponential or polynomial curves (C)

scalesw Set scaling factor for multipulse experiments (M)

Description	Sets the spectral width scaling factor for the multipulse sequences set up by macros `br24` and `mrev8`. The value of the scaling factor is stored in the parameter `scalesw`.
See also	*User Guide: solid-State NMR*

Related

`br24`	Set up BR24 multiple pulse experiment (M)
`mrev8`	Set up MREV8 multiple pulse experiment (M)
`scalesw`	Scale spectral width in directly detected dimension (P)
`scalesw1`	Set f_1 scaling factor for 2D multipulse experiments (M)

scalesw Scale spectral width in directly detected dimension (P)

Description	Adjusts the frequency scale dimension used with the parameter sets in the sequences set up by the `br24`, `mrev8`, `ssecho`, and `xpolar1` macros. If `scalesw` is active, the labels for the frequency scales includes the letters `sc` in parentheses. A scaled frequency can be referenced using the `rl` macro.
Values	`'n'`, number greater than 0.0
See also	*User Guide: Solid-State NMR*

Related

`br24`	Set up BR24 multiple pulse experiment (M)
`mrev8`	Set up MREV8 multiple pulse experiment (M)
`rl`	Set reference line (M)
`scalesw`	Set scaling factor for multipulse experiments (M)
`scalesw1`	Scale spectral width in 1st indirectly detected dimension (P)
`scalesw2`	Scale spectral width in 2nd indirectly detected dimension (P)

ssecho	Set up solid-state echo pulse sequence (M)
xpolar1	Set up parameters for XPOLAR1 pulse sequence (M)

scalesw1 Set f_1 scaling factor for 2D multipulse experiments (M)

Description Sets the f_1 spectral width scaling factor for the multipulse sequences set up by the br24 and mrev8 macros. The value of the scaling factor is stored in the parameter scalesw1.

See also *User Guide: Solid-State NMR*

Related

br24	Set up BR-24 multiple pulse experiment (M)
mrev8	Set up MREV8 multiple pulse experiment (M)
scalesw1	Scale spectral width in 1st indirectly detected dimension (P)

scalesw1 Scale spectral width in 1st indirectly detected dimension (P)

Description Analogous to the scalesw parameter except that scalesw1 applies to first indirectly detected dimension of a multidimensional data set. A scaled frequency along this dimension can be referenced using the rl1 macro.

Values 'n', number greater than 0.0

See also *User Guide: Solid-State NMR*

Related

rl1	Set reference line in 1st indirectly detected dimension (M)
scalesw	Scale spectral width in directly detected dimension (P)
scalesw1	Set f_1 scaling factor for 2D multipulse experiments (M)
scalesw2	Scale spectral width in 2nd indirectly detected dimension (P)

scalesw2 Scale spectral width in 2nd indirectly detected dimension (P)

Description Analogous to the scalesw parameter except scalesw2 applies to second indirectly detected dimension of a multidimensional data set. A scaled frequency along this dimension can be referenced using the rl2 macro.

Values 'n', number greater than 0.0

See also *User Guide: Solid-State NMR*

Related	rl2	Set reference line in 2nd indirectly detected dimension (M)
	scalesw	Set scaling factor for multipulse experiments (M)
	scalesw1	Set f_1 scaling factor for 2D multipulse experiments (M)

schedulerhelp Proshim Maintenance Scheduler help(C)

Applicability VnmrJ 3.2

Description Brings up help for the Proshim Maintenance Scheduler.

sd Set first decoupler frequency to cursor position (M)

Description Sets the first decoupler frequency offset parameter dof to place the first decoupler at the cursor position in the spectrum. This works only if the transmitter nucleus and first decoupler nucleus are the same (tn=dn).

See also *NMR Spectroscopy User Guide*

Related	dof	Frequency offset for first decoupler (P)
	dn	Nucleus of first decoupler (P)
	sd2	Set second decoupler frequency to cursor position (M)
	sd3	Set third decoupler frequency to cursor position (M)
	sda	Set first decoupler frequency array (M)
	tn	Nucleus for observe transmitter (P)

sd2 Set second decoupler frequency to cursor position (M)

Applicability Systems with a second decoupler.

Description Sets the second decouple frequency offset parameter dof2 to place the second decoupler at the cursor position in the spectrum. This works only if the transmitter nucleus and second decoupler nucleus are the same (tn=dn2).

See also *NMR Spectroscopy User Guide*

Related	dn2	Nucleus for second decoupler (P)
	dof2	Frequency offset for second decoupler (P)
	sd	Set first decoupler frequency to cursor position (M)
	sd2a	Set second decoupler frequency array (M)
	tn	Nucleus for observe transmitter (P)

sd2a Set second decoupler frequency array (M)

Applicability VnmrJ 3.1

Description With the cursor set to some position in the spectrum, "sd2" sets the decoupler offset parameter "dof2" to place the second decoupler at that position in the spectrum. To set up an array of offset values for the second decoupler, use "sd2" for the first position and "sd2a" for all subsequent positions. Either command will only work if the parameter "tn" is the same as the parameter "dn2".

sd3 Set third decoupler frequency to cursor position (M)

Applicability Systems with a third decoupler.

Description Sets the third decoupler frequency offset parameter dof3 to place the third decoupler at the cursor position in the spectrum. This works only if the transmitter nucleus and third decoupler nucleus are the same (tn=dn3).

See also *NMR Spectroscopy User Guide*

Related

dn3	Nucleus for third decoupler (P)
dof3	Frequency offset for third decoupler (P)
sd	Set first decoupler frequency to cursor position (M)
sd3a	Set third decoupler frequency array (M)
tn	Nucleus for observe transmitter (P)

sda Set first decoupler frequency array (M)

Description Sets up an array of offset values for the first decoupler, using sd for the first decoupler position and sda for subsequent positions. This works only if the transmitter nucleus and first decoupler nucleus are the same (tn=dn).

See also *NMR Spectroscopy User Guide*

Related

dn	Nucleus for first decoupler (P)
sd	Set first decoupler frequency to cursor position (M)
sd2a	Set frequency array for second decoupler (M)
sd3a	Set frequency array for third decoupler (M)
tn	Nucleus for observe transmitter (P)

sd3a Set third decoupler frequency array (M)

Applicability Systems with a third decoupler.

Description	Sets up an array of offset values for the third decoupler, using sd3 for the first position and sd3a for subsequent positions. This works only if the transmitter nucleus and third decoupler nucleus are the same (tn=dn3).
See also	*NMR Spectroscopy User Guide*

Related

dn2	Nucleus for third decoupler (P)
sd3	Set third decoupler frequency to cursor position (M)
sda	Set first decoupler frequency array (M)
tn	Nucleus for observe transmitter (P)

sdp — Show diffusion projection (M)

Description	Displays projection onto diffusion axis using the dsp facility. Use with 2D or 3D DOSY data after DOSY analysis. The unit of the resulting axis is D (10^{-10} m^2/sec). Because sdp overwrites the parameters in the current experiment, use it in only an experiment in which it is okay for existing data to be overwritten.
See also	*NMR Spectroscopy User Guide*

Related

dosy	Process DOSY experiments (M)

sel1d — Apptype macro for Selective 1D experiments (M)

Description: Perform the actions for Selective 1D protocols to set up, process, and plot experiments.

Examples:
sel1d('setup') – execute sel1d experimental setup
sel1d('process') – execute sel1d processing
sel1d('plot') – execute sel1d plotting

Related

apptype	Application type (p)
execpars	Set up the exec parameters (M)

select — Select spectrum, FID, trace, or 2D plane without display (C)

Syntax:
(1) select<('next'|'prev'|selection)><:index>
(2) select<(<'f1f3'|'f2f3'|'f1f2'><,'proj'>
<,'next'|'prev'|plane>)><:index>

Description: Directs future actions to apply to a particular spectrum or FID in a 1D array, to a trace in 2D (syntax 1), or to a particular 2D plane from a 3D data set (syntax 2). If select is called with no arguments, it returns the current index. When VnmrJ is first booted up, select is in 1D mode. select enters the 2D mode if any of the keywords 'f1f3', 'f2f3', 'f1f2', or 'proj' are present in the argument list. Entering the ds and jexp commands set select back in the 1D mode.

Arguments For 1D operations (syntax 1):

- 'next' is keyword to increment by 1 the 1D spectrum or trace index.
- 'prev' is keyword to decrement by 1 the 1D spectrum or trace index.
- selection is a number selecting a 1D spectrum, FID, or trace.
- index returns the number of the current 1D spectrum, FID, or trace.

For selecting various 2D planes of a 3D data set (syntax 2):

- 'f1f3', 'f2f3', and 'f1f2' are types of 2D planes. The parameters plane and index2 serve to indicate the exact 2D plane that is currently viewable by VnmrJ. Note that index2 cannot be entered from the keyboard (i.e., you cannot select a new 2D plane by changing the value of index2); you must use the select command instead.
- 'proj' is keyword to use the 2D projection whose plane type is determined by the parameter plane.
- 'next' is keyword to increment the parameter index2 to its next value and sets up VnmrJ to be ready to display the 2D plane whose number is the new index2 value.
- 'prev' performs analogously except that index2 is decremented.
- plane is a number selecting the plane.
- index returns the number of the current plane.

Examples

```
select('next')
select(2):r1
select('f1f3')
```

See also *NMR Spectroscopy User Guide, User Programming*

Related

arraydim	Dimension of experiment (P)
ds	Display a spectrum (C)
index2	Projection or 3D plane index selected (P)
jexp	Join existing experiment (C)
plane	Currently displayed 3D plane type (P)

selex Defines excitation band (M)

Syntax selex<(sh<,pw<,st<,ph<,fla<,trev>>>>>)>

Description Defines the excitation band from the position of cursors in the graphics window and reports them to user. It also sets r1 to excitation bandwidth and r2 to offset. selex is part of the Pbox software environment and uses the Pbox macros pbox_bw and putwave.

Arguments sh is the name of a shape file.

pw is the pulsewidth, in sec.

st is the spin status: 0 for excitation, 0.5 for refocusing, or 1 for de-excitation.

ph is the phase (or phase cycle, see wavelib/supercycles).

fla is the flip angle.

trev is the time reversal. This argument can be used to cancel time reversal introduced by setting the spin status (st) to 1 for de-excitation.

Examples selex
selex('esnob',0.0,1,90.0)

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

selexcit Set up PFG selective excitation pulse sequence (M)

Applicability Systems with a pulsed field gradient module.

Description Prepares an experiment for PFG (pulsed field gradient) selective excitation, with presaturation option.

See also *NMR Spectroscopy User Guide*

selexHT Set up a selective Hadamard experiment (M)

Description Sets up parameters for a selective shaped pulse Hadamard-encoded test experiment.

See also *NMR Spectroscopy User Guide*

Related

htofs1	Hadamard offset in ni (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
ft2d	Fourier transform 2D data (C)
sethtfrq1	Set Hadamard frequency list from a line list (M)

send2vnmr Send a command to VnmrJ (U)

Syntax send2Vnmr $vnmruser/.talk command

Description Sends a command from UNIX to VnmrJ using the port number stored in the $vnmruser/.talk file. This file is created when the macro listenon is entered on the VnmrJ command line.

Arguments command is any character string (commands, macros, or if statements) normally typed into the VnmrJ command line.

Examples send2Vnmr $vnmruser/.talk dg

See also *User Programming*

Related
`bootup` Macro executed automatically when VnmrJ activated (M)
`listenon` Enable receipt of messages from `send2Vnmr` (M)
`listenoff` Disable receipt of messages from `send2Vnmr` (M)

`seqfil` Pulse sequence name (P)

Description Identifies the name of the pulse sequence to be used. The value of `seqfil` is displayed on the top line of the screen after the "`Seq:`" label. Macros used to set up new pulse sequences, such as `Dept` and `Apt`, automatically change the `seqfil` parameter.

See also *NMR Spectroscopy User Guide*

Related `pslabel` Pulse sequence label (P)

`seqgen` Initiate compilation of user's pulse sequence (M,U)

Syntax
(From VnmrJ) `seqgen(<-static,>file<.c>)`
(From VnmrJ) `seqgen(file<.c>)`
(From VnmrJ) `seqgen`
(From VnmrJ) `seqgen('file<.c> file2 file3 ...')`
(From UNIX) `seqgen <–static> file<.c> <file1,...>`

Description Begins compilation of a user pulse sequence. When used from VnmrJ, the macro `seqgen` calls the UNIX shellscript `seqgen`, which can also be called directly from UNIX, as shown above. The `seqgen` shellscript then calls the compilation makefile `seqgenmake`, located in the directory `/vnmr/acqbin`.

The specified pulse sequence can be located in `~/vnmrsys/psglib` or in `/vnmr/psglib`. If two files with the same name exist in these two directories, the local directory (`~/vnmrsys/psglib`) takes precedence. For sequences in `/vnmr/psglib`, `seqgen` first copies the file into the local directory `~/vnmrsys/psglib` and then compiles it there; the resulting executable is then placed in `~/vnmrsys/seqlib`. A copy of the pulse sequence is also copied into the `seqlib` directory along with the executable. As it is running, `seqgen` reports where it found the specified sequence(s).

`seqgen` uses library files (object modules) found in `/vnmr/lib`. If `setuserpsg` and `psggen` has been run, the library files in the local directory `~/vnmrsys/psg` take precedence of those in `/vnmr/lib`.

Error messages are written into the file `file.errors`, where `file` is the name of the pulse sequence in `psglib` in which compilation is performed.

Note that `seqgen` not only accepts file names with and without extensions, but also accepts files specified with wildcards and complex paths (`seqgen` strips the directory part, and `seqgen /vnmr/psglib/apt` will compile `~/vnmrsys/psglib/atp.c` if it exists).

Arguments `-static` is a keyword for `seqgen` to use static rather than dynamic binding. Static binding results in larger executables in `seqlib` (several hundred Kbytes), but these sequences execute slightly faster (i.e., the `go` command). While insignificant generally, faster execution is helpful in some special applications such as the Scout Scan™ mode of LC-NMR, where the time spent on the `go` command becomes critical. Static binding results in a fixed-size time gain, regardless of the number of increments; for large multidimensional experiments, the speed difference is not noticeable.

`file` is the file name of a standard two-pulse sequence.

`.c` is the extension on the file name.

`file1,file2,...` are the names of files containing more sequences.

Examples (From VnmrJ) `seqgen('/vnmr/psglib/*.c')`
(From UNIX) `seqgen /vnmr/psglib/*.c`
(From UNIX) `seqgen apt dept noesy`
(From UNIX) `seqgen -static lc1d`

See also *User Programming*

`seqgenupdate` Update compilation of user's pulse sequence

Applicability VnmrJ 3.1

Description seqgenupdate has the same syntax as `seqgen`. Just like `seqgen`, one or more pulse sequence names can be supplied. `seqgenupdate` proceeds in two steps. In the first step, if any arguments are given, it passes them to `seqgen` for compilation. In the second step, it looks at the results of a preceeding `seqgen`. If permissions allow, it will move the compiled sequences back to the application directories or absolute paths they were copied from.

See also VNMR User Programming, Chapter 2, "Pulse Sequence Programming".

Related `psggen` compile a user PSG object library (M.U)

`serverport` Returns the VnmrJ network listening port value (C)

Applicability VnmrJ

Syntax `serverport`

Description The serverport command returns the port number when VnmrJ opens a network port (socket) for other programs to send it network messages. See the `write('net',...)` command for an example on how to use this port number.

Related `write` Write formatted text to a device (C)

set2D General setup for 2D experiments (M)

Syntax `set2D<(F2_dig_res<,F1_dig_res>)>`

Description Similar to set2d but does not execute par2d and does not make sw1, rfl1, and rfp1 decisions based on tn=dn condition.

Arguments F2_dig_res is the f_2 digital resolution desired, in Hz/pt. Default is 6.

F1_dig_res is the f_1 digital resolution desired, in Hz/pt. Default is 12.

Related

rfl1	Reference peak position in 1st indirectly detected dimension (P)
rfp1	Reference peak frequency in 1st indirectly detected dimension (P)
set2d	General setup for 2D experiments (M)
sw1	Spectral width in 1st indirectly detected dimension (P)

set2d General setup for 2D experiments (M)

Syntax `set2d(experiment<,F2_dig_res<,F1_dig_res>>)`

Description Runs the macro par2d to create new parameters needed for 2D experiments, then selects starting values for a number of parameters. The set2d macro is “internal” and not normally typed directly by the user.

Arguments experiment is the name of a 2D experiment (e.g., 'noesy').

F2_dig_res is the f_2 digital resolution desired, in Hz/pt.

F1_dig_res is the f_1 digital resolution desired, in Hz/pt.

Examples

```
set2d('cosyps')
set2d('hetcor',16)
set2d('het2dj',16,(2*sw1)/fn1)
```

See also *NMR Spectroscopy User Guide*

Related

par2d	Create 2D acquisition parameters (M)

set3dproc Set 3D processing (C)

Syntax `set3dproc<(<'nocoef'><,directory>)>`

Description Creates the file procdat that contains binary 3D information used by ft3d in processing the 3D FID data. It also creates the 3D parameter set procpar3d that is used by the select command to display the 2D planes from the 3D transformed data. set3dproc can only create the proper 3D coefficient file if the parameters phase and phase2 are used to generate States-Haberkorn (hypercomplex) or TPPI data along the t_1 and t_2 dimensions.

set3dproc creates the coefficient file for the following five values of array (where SH is States-Haberkorn):

- if array='' (null string), type of 3D data is TPPI(t_1) – TPPI(t_2)
- if array='phase', type of 3D data is SH(t_1) – TPPI(t_2)
- if array='phase2', type of 3D data is SH(t_2) – TPPI(t_1)
- if array='phase2,phase', type of 3D data is SH(t_1) – SH(t_2)

If array is set to some other value, set3dproc cannot create the 3D coefficient file and an error is reported within VnmrJ.

Arguments 'nocoef' is a keyword that the 3D coefficient file coef is not to be created.

directory is the name of the directory for procdat and procpar3d. The default is the subdirectory info in the directory curexp.

Examples
```
set3dproc
set3dproc('nocoef','curexp/info3d')
```

See also *NMR Spectroscopy User Guide*

Related

array	Parameter order and precedence (P)
ft3d	Perform a 3D Fourier transform (M,U)
phase	Phase selection (P)
phase2	Phase selection for 3D acquisition (P)
select	Select a spectrum or 2D plane without displaying it (C)
wftt3	Process f_3 dimension during 3D acquisition (M)

setallshims Set all shims into hardware (M)

Description Sets shims from the current parameter tree into hardware. setallshims is equivalent to entering load='y'su but without setting all the hardware parameters normally set by su (temperature, decoupling, transmitter initialization, etc.). The shims used depend on the shimset configuration. For the shim set on the Ultra•nmr shim system, setallshims is active only if hardware-to-software shim communication is enabled.

See also *NMR Spectroscopy User Guide*

Related

load	Load status of displayed shims (P)
readallshims	Read all shims from hardware (M)
readhw	Read current values of acquisition hardware (C)
sethw	Set values for hardware in acquisition system (C)
shimset	Type of shim set (P)
su	Submit a setup experiment to acquisition (M)

setcolor Set colors for graphics window and for plotters (C)

Syntax
(1) setcolor('pcl',item_index,'color')
(2) setcolor('hpgl',item_index,'color')
(3) setcolor('pen',pen_number,'color')

(4) `setcolor('graphics',item_index,red,green,blue)`
(5) `setcolor('ps',item_index,red,green,blue)`
(6) `setcolor('plotter',black_plane,color_planes)`

Description Sets colors used on the graphics window and on plotters. This command is a utility program used by the `color` macro and other macros. It is not expected that `setcolor` would be entered directly from the input window.

Arguments `'pcl'` is a keyword to set colors on a plotter device that uses the PCL language. PCL plotters are the laser type of plotter.

`'hpgl'` is a keyword to set colors on a plotter device that uses the HPGL language. HPGL plotters are the pen type of plotter.

`'pen'` is a keyword that next two arguments set the color for a physical pen on a plotter device that uses the HPGL language.

`'graphics'` is a keyword to set colors on the graphics window.

`'ps'` is a keyword to set colors on a plotter using the PostScript language.

`red`, `green`, `blue` are three integers between `0` and `255` that set the amount of red, green, and blue color on the graphics window or PostScript plotter.

`'plotter'` is a keyword that the next two arguments set the black mode and number of colors available for a plotter device.

`item_index` is an index number from the following list that represents a specific drawing item.

8	background of images
9	real channel of an FID
10	imaginary channel of an FID
11	spectrum
12	integral
13	parameters
14	scale
15	threshold line (graphics device only)
16	second spectrum or FID in `addi` (graphics device only)
17	result spectrum or FID in `addi` (graphics device only)
18	cursors (graphics device only)
19	foreground of images
20	background color of graphics window (graphics device only)
20-35	contour 0 to contour 15 of absolute value 2D display
36-42	contours –7 to –1 of phased 2D display
44-50	contours 1 to 7 of phased 2D display

`pen_number` is an integer from `1` to `8` that specifies the physical pen used.

`color` is a string for the color set for the device: `'red'`, `'green'`, `'blue'`, `'cyan'`, `'magenta'`, `'yellow'`, `'white'`, or `'black'`.

`black_plane` is 1 or 0, specifying whether the plotter has a separate black mode. Because all currently supported plotters have this feature, the value is usually 1.

`color_planes` specifies how many colors are available. Use 3 for color plotters and `0` for black and white plotters.

Examples `setcolor('pcl',11,'green')`
`setcolor('hpgl',11,'red')`
`setcolor('pen',2,'red')`
`setcolor('graphics',11,255,0,0)`
`setcolor('ps',11,255,255,0)`
`setcolor('plotter',1,0)`

See also *NMR Spectroscopy User Guide*

Related `addi` Start interactive add/subtract mode (C)
`color` Select plotting colors from a graphical interface (M)

setDECpars Sets Decoupler Parameters

Description Called to set decoupler parameters when dn is changed during customizations.

Syntax setDECpars

Related `setDECpars, setobspars`

setdec2pars Set decoupler 2 parameter values from probe file (M)

Syntax `setdec2pars`

Description Reads from the probe file `pwx2lvl`, `pwx2`, `dpwr2`, `dmf2`, `dmm2`, `dres2`, and `dseq2` values, if they exist, and updates the current experiment parameters.

Related `setdecpars` Set decoupler parameter values from probe file (M)

setdgroup Set the Dgroup of a parameter in a tree (C)

Syntax `setdgroup(parameter,dgroup<,tree>)`

Description Sets the Dgroup of a parameter in a tree. The application determines the usage of `setdgroup`. Only Tcl-dg currently uses this feature.

Arguments `parameter` is the name of the parameter.

`dgroup` is an integer.

`tree` is `'current'`, `'global'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the description of the `create` command for more information on types of trees.

Examples `setdgroup('a',1)`
`setdgroup('b',3,'global')`

See also *User Programming*

Related `create` Create new parameter in a parameter tree (C)

setenumeral Set values of a string parameter in a tree (C)

Syntax `setenumeral(parameter, N, enum1, enum2,..., enumN<, tree>)`

Description Sets the possible values of a string parameter in a parameter tree. To remove enumerated values from a parameter, set argument `N` to `0` (see example below).

Arguments `parameter` is the name of the parameter.

`N` is the number of enumeral values to be assigned to `parameter` (or removed from `parameter` if `N` is set to `0`).

`enum1` to `enumN` are the possible string values of the parameter.

`tree` is `'current'`, `'global'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the description of the `create` command for more information on types of trees.

Examples
```
setenumeral('size',0)
setenumeral('size',2,'large','small')
setenumeral('user',3,'user','superuser','master',
   'global')
```

See also *User Programming*

Related `create` Create new parameter in a parameter tree (C)

setether Connect or reconnect host computer to Ethernet (U)

Description Connects or reconnects the host computer to the Ethernet network. Only `root` can execute this shellscript properly. If the system is already connected to the Ethernet network, `setether` does nothing.

On systems running Solaris, `setether` undoes the work of `setnoether`. You cannot use `setether` unless you previously entered the `setnoether` command. `setether` restores the files `hostname.le0`, `defaultdomain`, and `defaultrouter` so that Ethernet is activated on the host computer when UNIX is rebooted.

See also *VnmrJ Installation and Administration*

Related `setnoether` Disconnect host computer from Ethernet (U)

setexport Set parameter bits for use with protocols (M)

Description Set the parameter protection bits for use with the rtx command. Usually called by other macros, and not used from the command line.

Related `rtx`
`cqprotocol` Create study queue parameters for liquids (M)

setfrq Set frequency of rf channels (C)

Syntax `setfrq<(channel)><('nucleus')>`

Description Calculates frequencies based on the nucleus (tn, dn, dn2, etc.), referencing (lockfreq), solvent, and the offset parameter (tof, dof, etc.). The result of the calculation is stored in parameters sfrq, dfrq, dfrq2, etc. The parameters are rounded to the resolution of the channel—either 0.1 or 100 Hz.

The `setfrq` command should never need to be entered from the keyboard. It is called automatically when the appropriate parameters are changed or a parameter set is returned. If a parameter is entered that affects a single frequency, `setfrq` is called from an internal underscore macro (e.g., `_tn`, `_tof`, `_dn`, `_dof`) to recalculate the frequency for that channel. Likewise, if a parameter is entered that affects all frequencies, `setfrq` is called from an internal underscore macro (e.g., `_solvent`, `_lockfreq`) to recalculate the frequencies.

Arguments `channel` is a single integer specifying the rf channel to be set. The default is to calculate the frequencies for all rf channels.

`nucleus` displays or returns the frequency of the supplied nucleus. Channel 1 is assumed for rounding information and an offset (e.g., tof or dof) is not added to the result.

Examples
```
setfrq
setfrq(2)
setfrq('P31'):freq
```

See also *NMR Spectroscopy User Guide*

Related spcfrq Display frequencies of rf channels (M)

setgauss Set a Gaussian fraction for lineshape (M)

Syntax (1) `setgauss(fraction)`
(2) `setgauss(fraction*)`

Description Modifies the output of a deconvolution using pure Lorentzian lineshape (`fitspec.outpar`) and makes it the input for a subsequent analysis (`fitspec.inpar`), after first modifying the Gaussian fraction. To allow this fraction to vary, use syntax 1; to fix the fraction, use syntax 2.

Arguments `fraction` is the Gaussian fraction of the lineshape, a number from 0 to 1. To fix the fraction (syntax 2), suffix the value with an asterisk (*) and enclose the value in single quotes (see the second example below).

Examples
```
setgauss(0.4)
setgauss('1.0*')
```

See also *NMR Spectroscopy User Guide*

Related fitspec Perform spectrum deconvolution (C)

setgcal Set the gradient calibration constant (M)

Applicability Systems with pulsed field gradients (PFG) or imaging capabilities.

Description Determines the gradient calibration constant gcal by using a proton phantom of known dimensions. setgcal requests the linear dimension of the phantom in the readout direction. It uses the value entered, together with cursor separation of this dimension from the image profile and the strength of the readout gradient gzlvl1 if pulsed field gradients, to calculate gcal in units of gauss/cm-DAC units. You are then prompted whether this value should be entered. If you answer yes, it is stored as a system constant in the your global file.

Note that a particular value of gcal is closely related to the current eddy current compensation settings. If these settings are changed (e.g., reading in a new curecc file), a different value of gcal should be expected.

Before running setgcal, use the pulse sequence set up by profile to acquire a signal from a known sized object while the gradient is on.

See also *Pulsed Field Gradient Modules Installation; VnmrJ Imaging NMR*

Related

gcal	Gradient calibration constant (P)
profile	Set up pulse sequence for gradient calibration (M)

setgcoil Assign sysgcoil configuration parameter (M)

Syntax `setgcoil<(file)>`

Description Allows users to change the configured gcoil for the system. setgcoil updates the systemglobal parameter sysgcoil to the named table and updates the assignment value of the parameter gcoil in the named table. The directory $vnmrsystem/imaging/gradtables must have write permission for all users for the macro to be effective. This table now exists in the system local /var/vnmr/gradtables directory, with a soft link from $vnmrsystem/imaging/gradtables to that directory.

Arguments file is the any legal file name defined for the parameter gcoil.

See also *VnmrJ Imaging NMR*

Related

config	Display current configuration and possible change it (M)
gcoil	Read data from gradient calibration tables (P)
sysgcoil	System value for gcoil parameter (P)

setgrid Divide graphics window into rows and columns (C)

Syntax `setgrid(row<,column>)`

Description	Divides graphics window into an array of rows and columns (or window panes). Only one pane is active at a time. An individual pane can be activated by double-clicking in it with the left mouse button or by entering `setwin` in the input window.
Arguments	`row` is the number of rows (maximum is 3) in the graphics window. If 0 is entered, the number of rows remains the same; e.g., in `setgrid(0,2)`, the number of rows is unchanged and two columns are created in each row. `column` is the number of columns (maximum is 3) in the graphics window.

Examples	`setgrid(3)`
	`setgrid(3,3)`
	`setgrid(0,2)`

See also	*NMR Spectroscopy User Guide*

Related

`curwin`	Current window (P)
`fontselect`	Open FontSelect window (C)
`jwin`	Activate current window (M)
`mapwin`	List of experiment numbers (P)
`setwin`	Activate selected window (C)

setgroup — Set group of a parameter in a tree (C)

Syntax	`setgroup(parameter,group<,tree>)`
Description	Sets the group of a parameter in a tree.
Arguments	`parameter` is the name of the parameter. `group` is one of the following keywords: `'all'`, `'sample'`, `'acquisition'`, `'processing'`, `'display'`, or `'spin'`. `tree` is one of the keywords 'current', `'global'`, or `'processed'`. The default is `'current'`. See the `create` command for information on the types of trees.
Examples	`setgroup('a','sample')` `setgroup('b','all','global')`
See also	*User Programming*

Related

`create`	Create new parameter in a parameter tree (C)
`destroy`	Destroy a parameter (C)
`destroygroup`	Destroy parameters of a group in a tree (C)
`display`	Display parameters and their attributes (C)
`groupcopy`	Copy parameters of group from one tree to another (C)
`paramvi`	Edit a parameter and its attributes using `vi` text editor (M)
`setlimit`	Set limits of a parameter in a tree (C)
`setprotect`	Set protection mode of a parameter (C)

sethtfrq1 Set a Hadamard frequency list from a line list ((M)

Description A macro to set the Hadamard frequency list htfrq1 from a line list curexp+'/dll.out'. It assumes that the line list has already been created. The macro also sets ni to the Hadamard matrix size, creates htofs1, and sets fn1 from the minimum frequency difference in htfrq1.

See also *NMR Spectroscopy User Guide*

Related

htfrq1	Hadamard frequency list in ni (P)
dll	Display listed line frequencies and intensities (C)
htofs1	Hadamard offset in ni (P)
fn1	Fourier number in the 1st indirectly detected dimension (P)
ni	Number of increments in the 1st indirectly detected dimension (P)

sethw Set values for hardware in acquisition system (C)

Applicability Syntax 1 through 5 apply to all systems. Syntax 6 applies only to systems with a sample changer. Syntax 7 and 8 apply only to systems with a variable temperature (VT) controller.

Syntax The following syntax is used with the sethw command:

```
1 sethw(<'wait'|'nowait',>par1,val1<,par2,val2,...)
2 sethw('lock','on'|'off')
3 sethw('spin',speed)
4 sethw('spinner','bump')
5 sethw('eject','on'|'off')
6 sethw('loc',location)
7 sethw('vt','reset'|'off')
8 sethw('temp',temperature)
9 sethw('lockfreq',lockfreq_value)
```

Description sethw allows the VNMR program to set values for selected parameters in the acquisition hardware. sethw cannot be used when an acquisition is in progress or when the acqi program is active.

Syntax 1 can be used to set the lock system parameters lockpower, lockgain, lockphase, and z0. This syntax can also be used to set the values of the shims. The particular shim that can be set depends upon the type of shim hardware present in the system. See the description of shimset for a list of the shim names for each type of shim hardware.

Syntax 2 turns the hardware lock on or off.

Syntax 3 controls spinning speed.

Syntax 4 carries the sample to bump by giving it a short burst of eject air. This is sometimes useful to reseat the sample if it is failing to spin.

Syntax 5 ejects and inserts samples into the probe. Entering the command `sethw('eject','on')` is equivalent in function to macros `eject` and `e`; and `sethw('eject','off')` is equivalent to macros `insert` and `i`.

Syntax 6 sets a location for the sample currently in the magnet on a system with a sample changer. The parameter `loc` is updated.

Syntax 7 resets the VT controller, useful when changing the probe in a system with VT regulation. By entering `sethw('vt','reset')` after installing a new probe in the magnet and attaching the VT controller interface to the probe, the VT controller is ready to regulate the temperature. No other parameters can be modified by the command. As an alternate, you can manually turn the VT controller unit off and then back on. Syntax 7 also turns the VT controller off by entering `sethw('vt','off')`.

Syntax 8 sets the temperature in degrees celsius. The host computer does not wait for the temperature to regulate.

Syntax 9 sets the lock frequency, in MHz.

Arguments `'wait'` or `'nowait'` keyword must be either the first or last argument.

- `'wait'` sends the new values to the acquisition console, verifies these values, and updates the corresponding parameters. This is the default.
- `'nowait'` sends the new values to the console without verifying them or changing parameters.

`parameter1,value1,parameter2,value2,...` are paris of parameter names and their values (see the first two examples below). At least one parameter name and its value must be specified. A maximum of ten parameters can be set.

`'lock'`,`'on'` is a keyword pair to turn the hardware lock on.

`'lock'`,`'off'` is a keyword pair to turn the hardware lock off.

`'liqbear'` sets the bearing air on level; see `liqbear` parameter.

`'pneufault'` second argument is `'clear'`, `'n'`, `'w'`, or `'y'` to clear or set the pneumatics fault code.

`'spin'` is a keyword that identifies the next argument, `speed`, as the sample spinning speed, in Hz.

`'spinner'`,`'bump'` is a keyword pair to bump the sample.

`'eject'`,`'on'` is a keyword pair to eject the sample from the probe.

`'eject'`,`'off'` is a keyword pair to insert the sample into the probe.

`'loc'` is a keyword to identify that the next argument, `location`, is a number for the sample currently in the magnet (`'loc'` is unrelated to the `loc` parameter).

`'vt'`,`'reset'` is a keyword pair to reset the VT controller after the controller has been disconnected from the probe. This is equivalent to turning the VT controller power off and on.

`'vt'`,`'off'` is a keyword pair to turn the VT controller off.

`'temp'` is a keyword that identifies the next argument, `temperature`, as the requested sample temperature, in degrees celsius.

`'lockfreq'` is a keyword that the next argument is the lock frequency.

`lockfreq_value` is the `lockfreq` value, in MHz, for the lock frequency.

`'lockrate'` is a number <5000 used internally; usually 20 or 2000.

Examples
```
sethw('z1c',30,'z2c',-50)
sethw('wait','z1',150,'z2',-400)
sethw('lock','on')
sethw('spin',20)
sethw('spinner','bump')
sethw('eject','on')
sethw('loc',5)
sethw('vt','reset')
sethw('lockfreq',46.042)
```

See also *NMR Spectroscopy User Guide*

Related

`loc`	Location of sample in tray (P)
`lockpower`	Lock power (P)
`lockfreq`	Lock frequency (P)
`lockgain`	Lock gain (P)
`lockphase`	Lock phase (P)
`readhw`	Read current values of acquisition hardware (C)
`sethwshim`	Set values for hardware in acquisition system (C)
`spin`	Sample spin rate (P)
`z0`	Z0 field position (P)

sethwshim — Special case of sethw for setting shims (C)

Applicability VnmrJ 3.2

Description `sethwshim` sethwshim command is a special case of `sethw`. It takes two arguments, the shim name and shim value, as in sethwshim('z1',1000)

Arguments `sethwshim('z1',1000)`

See also *NMR Spectroscopy User Guide*

Related

`sethw`	Set values for hardware in acquisition system (C)

setint — Set value of an integral (M)

Syntax `setint(int_number<,value>)`

Description Sets the value of an integral.

Arguments int_number is the integral number. It corresponds to the index number displayed by dli if all integrals are shown (i.e., intmod='full') or the region if alternating integrals are shown (i.e., intmod='partial').

value sets the actual value of the selected integral. The default is ins.

Examples
```
setint(2)
setint(1,3)
```

See also *NMR Spectroscopy User Guide*

Related		
	dli	Display list of integrals (C)
	ins	Integral normalization scale (P)
	intmod	Integral display mode (P)

setlimit Set limits of a parameter in a tree (C)

Applicability All

Syntax setlimit(name, max,min,step [,tree])

setlimit(name, index[,tree])

Description setlimit sets the limits of a variable in a tree.

The limits are max value, min. value and step size. A variable, such as an index into the table, can look up maximum, minimum, and step sizes in a table. Supplying all three (max, min., and step) arguments sets the parameter's protection bits (see setprotect) so that the table lookup is turned off. The parameter's protection bits are set so that table lookup is turned on if only a single index argument is supplied.

The step value is only used if the parameter is a real number.

Step Value	Parameter setting
< -1	The parameter is set to the nearest larger value that is a power of 2. The fn parameter uses a step of -2 to select this case.
>-1 and < 0	The inverse of the parameter is set to the nearest multiple of the absolute value of the step. The sw parameter uses a step of negative of the minimum dwell time to select this mode.
>0 and <1	The parameter is set to the nearest multiple of the step value. As an equation, value = n * step where n is a positive or negative integer.
≥ 1	The parameter is set to nearest value that is a multiple of step relative to the minimum value. For example, setlimit('var',3,-3,2) allows only the following values -3, -1, 1, and 3. As an equation, value = min + n*step where n is an integer >= 0. In this example, the equation is: value = (-3) + (n * 2).

Up to four optional return arguments can be used. The first will return the maximum, the second will return the minimum, and the third will return the step size. The fourth argument will return a 0 if the parameter is not using an indexed table lookup for the maximum, minimum, and step size. If the parameter is using the table lookup mechanism, the fourth argument will be set to the index for that table.

The variable trees are `'current'`, `'global'`, `'processed'` and `'systemglobal'`. The default tree is `'current'`.

Arguments `name` — the name of the variable.

`tree` — the variable tree: `current (the default)`, global, `processed`, or `systemglobal`.

Examples `setlimit('a',10000,0,.3)`

`setlimit('b',1e5,-3e2,1,'global')`

`setlimit('dpwr',9)`

See also *User Programming*

Related

`create`	Create new parameter in a parameter tree (C
`destroy`	Destroy a parameter (C)
`display`	Display parameters and their attributes (C)
`fread`	Read parameters from file and load them into a tree (C)
`fsave`	Save parameters from a tree to a file (C)
`getlimit`	Get the limits of a variable in a tree (C)
`paramvi`	Edit a parameter and its attributes using `vi` text editor (M)
`parmax`	Parameter maximum values (P)
`parmin`	Parameter minimum values (P)
`parstep`	Parameter step size values (P)
`prune`	Prune extra parameters from current tree (C)
`setgroup`	Set group of a parameter in a tree (C)
`setprotect`	Set protection mode of a parameter (C)
`settype`	Change type of a parameter (C)
`setvalue`	Set value of any parameter in a tree (C)

`setlk` Set up lock parameters (M)

Syntax `setlk(solvent)`

Description Called from other macros to provide adjustment of locking and shimming as a function of solvent. Removing quotation marks from around different parts of the text file of the macro places that particular section into effect. If the macro is left unchanged, setting `alock='s'` is required in the parameter sets where used.

Arguments `solvent` is the solvent to be used.

See also *NMR Spectroscopy User Guide*

Related

`alock`	Automatic lock status (P)

setlockfreq Set lock frequency (M)

Description Calculates and sets the lock frequency parameter lockfreq. Before using setlockfreq, you must acquire a signal using ^{1}H as the transmitter nucleus (tn='H1'). To avoid errors in calculating frequencies, set lockfreq='n' before starting the acquisition.

See also *VnmrJ Installation and Administration*

Related
lockfreq Lock frequency (P)
tn Nucleus for observe transmitter (P)

setLP Set up linear prediction in the direct dimension (M)

Applicability ALL

Syntax setLP(n)

Description Sets up linear prediction in the direct dimension using the number of coefficients specified.

Examples setLP(3)

See also *NMR Spectroscopy User Guide*

Related
lpext LP data extension in np dimension (P)
lpfilt LP coefficients to calculate in np dimension (P)
lpnupts LP number of data points in np dimension (P)
lpopt LP algorithm data extension in np dimension (P)
proc Type of processing on np FID (P)
setrc Set frequency referencing based upon lock signal shift (M)
strtext Starting point for LP data extension in np dimension (P)
strtlp Starting point for LP calculation in np dimension (P)

setLP1 Set F1 linear prediction parameters (M)

Syntax setLP1<(extended_length<,current_length>)>

Description Sets F1 linear prediction parameters. If no arguments are specified, the interferograms are quadrupled in length.

Arguments extended_length is the number of complex points now existing (ni).

current_length is the number of points desired after the (forward) linear prediction.

See also *NMR Spectroscopy User Guide*

Related
ni Number of increments in 1st indirectly detected dimension (P)

setlp0 Set parameters for zero linear phase (M)

Syntax `setlp0`

Description A new value of `ddrtc` is calculated by `setlp0` using the current values of `alfa`, `rof2`, and `lp` to achieve a zero linear phase condition (`lp=0`). A trial experiment must first be acquired and phased for pure absorption before running `setlp0`. A value of `lp` near zero is required for flat base line.

See also NMR Spectroscopy User Guide

Related

`alfa`	Set alfa delay before acquisition (P)
`ddrtc`	Set ddr time constant (P)
`lp`	First-order phase in directly detected dimension (P)
`rp`	Zero-order phase in directly detected dimension (P)
`sw`	Spectral width in directly detected dimension (P)
`rof2`	Receiver gating time following pulse (P)

setnoether Disconnect host computer from Ethernet (U)

Description Disconnects the host computer from the Ethernet network. Only `root` can execute this shellscript properly. `setnoether` does nothing if the system is already disconnected from the Ethernet network.

On systems running Solaris, `setnoether` renames the `hostname.le0`, `defaultdomain`, and `defaultrouter` files so that Ethernet is not activated when the system is rebooted.

See also *VnmrJ Installation and Administration*

Related `setether` Connect or reconnect host computer to Ethernet (U)

setobspars Sets Observe Parameters

Description Called to set observe parameters when tn is changed during customizations.

Syntax setobspars

Related `setDECpars, setobspars`

setoffset Calculate offset frequency for given nucleus and ppm (M)

Syntax `setoffset(nucleus,ppm):offsetfreq`

Description Using the `setref` macro, `setoffset` calculates the offset frequency for a given chemical shift and returns the value.

Arguments	`nucleus` is the given nucleus.	
	`ppm` is the chemical shift.	
	`offsetfreq` returns the offset frequency for the given chemical shift.	
Examples	`setoffset(tn,5):tof` `setoffset('C13',85):dof`	
See also	*NMR Spectroscopy User Guide*	
Related	`setref`	Set frequency referencing for proton spectra (M)

setparams — Write parameter to current probe file (M)

Syntax `setparams(param,value<,nucleus>)`

Description Writes the value of a parameter to the current probe file. The name of the probe file is referenced from the parameter `probe`.

Arguments `param` is the name of the parameter to write.

`value` is a string with the value to be written for the parameter.

`nucleus` is the nucleus to write in the probe file. The default is the current value of the parameter `tn`.

Examples

```
setparams('pw90','10')
setparams('pplvl','60')
setparams('dpwr',$strdpwr,'H1')
```

See also *NMR Spectroscopy User Guide*

Related

`addnucleus`	Add new nucleus to existing probe file (M)
`addparams`	Add parameter to current probe file (M)
`addprobe`	Create new probe directory and probe file (M)
`getparam`	Retrieve parameter from probe file (M)
`probe`	Probe type (P)
`tn`	Nucleus for the observe transmitter (P)
`updateprobe`	Update probe file (M)

setpen — Set maximum number of HP plotter pens (M)

Syntax `setpen<(maxpen,max_number_pens)>`

Description Allows the user to interactively define the maximum number of pens when changing to a Hewlett-Packard plotter.

Arguments `maxpen` is the current value of the parameter `maxpen`.

`maximum_number_pens` is the maximum number of pens to be used. If the value of `max_number_pens` is less than or equal to the current value of the parameter `maxpen`, this value becomes the new value of `maxpen`.

See also *NMR Spectroscopy User Guide*

Related		
	color	Select plotting colors from a graphical interface (M)
	maxpen	Maximum number of pens to use (P)

setplotdev Return characteristics of a named plotter (C)

Syntax `setplotdev<:plotter_type,plotter_host,ppmm,raster>`

Description Returns information from the devicenames and devicetable files to identify the characteristics of a plotter. This command need never be entered directly by a user because it is automatically called whenever the plotter parameter is set. Note that different “types” of plotters (and printers) are characterized in devicetable. The devicenames file associates different “names” to a given “type.”

Arguments plotter_type returns the type of the named plotter.

plotter_host returns the host associated with the plotter.

ppmm returns the plotter resolution in points per millimeter.

raster returns the value from the devicetable file.

See also *VnmrJ Installation and Administration*

Related		
	plotter	Plotter device (P)

setpower Set power and pulsewidth for a given γB1 value (M)

Syntax `setpower(γB1,nucleus)`

Description Sets power level and pw90 values. For tn, setpower uses ref_pwr and ref_pw90 from the parameter set or from the probe table. For dn, it uses ref_pwxlvl and ref_pwx90 from the parameter set or from the probe table. For dn2, it uses ref_pwx2lvl and ref_pwx290 from the parameter set or from the probe table. If the reference power levels and pulse width do not exist, setpower uses tpwr (pw90), dpwr (1/dmf) or dpwr2 (1/dmf2) (if the nucleus is tn, setpower uses tpwr; if the nucleus is dn, it uses dpwr; if the nucleus is dn2, it uses dpwr2).

Arguments γB1 is a given γB1 value.

nucleus is a given nucleus.

Examples
```
setpower(sw,tn)
setpower(5000,H1)
```

Related		
	dn	Nucleus for first decoupler (P)
	dn2	Nucleus for second decoupler (P)
	dpwr	Power level for first decoupler with linear amplifiers (P)
	dpwr2	Power level for second decoupler (P)
	pw90	90° pulse width (P)

sw	Spectral width in directly detected dimension (P)
tpwr	Observe transmitter power level with linear amplifiers (P)

setprotect Set protection mode of a parameter (C)

Syntax `setprotect(parameter,'set'|'on'|'off',bit_vals<,tree>)`

Description Enables changing the protection bits associated with a parameter.

Arguments `parameter` is the name of the parameter.

`'set'` causes the current protection bits for the parameter to be completely replaced with the bits specified by `bit_vals`.

`'on'` causes the bits specified in `bit_vals` to be turned on without affecting any other protection bits.

`'off'` causes the bits specified in `bit_vals` to be turned off without affecting any other protection bits.

`'list'` causes all parameter with the specified bit_vals to be listed. This list may be returned to the calling macro.

`'clear'` option clears the specified bit_vals from all parameters. For both the list and clear options, the names argument can be `''`. The return value when `setprotect` is called with the `list` option can be used as the `'names'` argument for other forms of `setprotect`. It can also be names for other commands which use lists of parameter names, such as `writeparam` and `readparam`.

`bit_vals` is the *sum* of the *values* of bits selected from the following list:

Bit	Value	Description
0	1	Cannot array the parameter
1	2	Cannot change active/not active status
2	4	Cannot change the parameter value
3	8	Causes `_parameter` macro to be executed (e.g., if parameter is named `sw`, macro `_sw` is executed when `sw` is changed)
4	16	Avoids automatic redisplay
5	32	Cannot delete parameter
6	64	System ID for spectrometer or data station
7	128	Cannot copy parameter from tree to tree
8	256	Will not set `array` parameter
9	512	Cannot set parameter enumeral values
10	1024	Cannot change the parameter's group
11	2048	Cannot change protection bits
12	4096	Cannot change the display group
13	8192	Look up minimum, maximum, step values in table
14	16384	Parameter marked for locking (P_LOCK; see rtx)
15	32768	Global parameter not shared in multiple VJ viewports
16	65536	Force automatic redisplay in VJ templates

For example, to change the first two protection bits, with values 1 and 2, either enter `setprotect` twice (once for each value) with the keyword 'on', or enter setprotect once with bit_vals set to 3 (sum of 1 and 2) with the keyword 'set'.

tree is one of the keywords 'global', 'current', 'processed', or 'systemglobal'. The default is 'current'. Refer to the create command for more information on the types of parameter trees.

Examples
```
setprotect('syn,'on',2)
setprotect('pslabel','on',8)
```

See also *User Programming*

Related

array	Parameter order and precedence (P)
create	Create new parameter in a parameter tree (C)
destroy	Destroy a parameter (C)
display	Display parameters and their attributes (C)
fread	Read parameters from file and load them into a tree (C)
fsave	Save parameters from a tree to a file (C)
getlimit	Get the limits of a variable in a tree (C)
paramvi	Edit a parameter and its attributes using vi text editor (M)
prune	Prune extra parameters from current tree (C)
setlimit	Set limits of a parameter in a tree (C)

setpw180ad Creates and sets observe adiabatic pulse shapes (M)

Syntax
```
setpw180ad(tn,<'make' or 'create'>,<'base shape'>,<'bandwidth in ppm''>)
```

Applicability VnmrJ 3.1

Description Based upon probe calibrations, this will create adiabatic pulse shapes for a given nucleus as defined in the tn parameter.

Based upon the second argument, which defaults to "make", it will set the adiabatic pulse parameter values.

Examples
```
setpw180ad(tn)
setpw180ad(tn,'make')
setpw180ad(tn,'make',wurst2i)
setpw180ad(tn,'create',wurst2i)
setpw180ad(tn,'make','wurst2i',115)
```

Related setpwx180ad Och_adiabtic_module

setpwx180ad Creates and sets decoupler adiabatic pulse shapes (M)

Syntax
```
setpwx180ad(dn,<'make' or 'create'>,<'base shape'>,<'bandwidth in ppm''>)
```

Applicability VnmrJ 3.1

Description
: Based upon probe calibrations, this will create adiabatic pulse shapes for a given nucleus as defined in the `dn` parameter.

Based upon the second argument, which defaults to "`make`", it will set the adiabatic pulse parameter values.

Examples
: `setpwx180ad(dn)`

`setpwx180ad(dn,'make')`

`setpwx180ad(dn,'make',wurst2i)`

`setpwx180ad(dn,'create',wurst2i)`

`setpwx180ad(dn,'make','wurst2i',115)`

Related
: `setpw180ad` `Dch_adiabtic_module`

setrc — Set receiver constants (M)

Applicability
: VNMRS and 400 - MR systems

Syntax
: `setrc`

Description
: Sets receiver time constants to optimal values. `alfa` is set to a minimum value from the probe file (default is 10 µs). `rof2` is set to a minimum value from the probe file (default is 25 µs). lp is set to zero. `ddrtc` is set to a value based upon the ddrpm parameter, which is set based upon pulse sequence type (default value `ddrpm = 'p'`). Linear prediction is turned on in the direct dimension if the `ddrtc` value is more than a dwell time. `setrc` is used in the apptype macros for setting up pulse sequences or from the command line to optimize receiver constants.

Description
: sets receiver time constants to optimal values.

See also
: *NMR Spectroscopy User Guide*

Related

`alfa`	Set alfa delay before acquisition (P)
`rof2`	Receiver gating time following pulse (P)
`pw`	Pulse width (P)
`probe`	Probe type (P)
`ddrtc`	Set ddr precession mode (P)
`ddrpm`	Set ddr precession mode (P)
`sw`	Spectral width in directly detected dimension (P)
`setLP`	Set F1 linear prediction parameters (M)

setref — Set frequency referencing (M)

Syntax
: `setref<(nucleus)>:$rfl,$rfp,$reffrq,$refpos`

Description
: Calculates the referencing for a given parameter or FID data set, for samples locked on deuterium, and based on the chemical shift of the lock solvent line. `setref` uses information in `/vnmr/solvents` (^{2}H chemical shift for current solvent) and `/vnmr/nuctables/nuctabref`

(absolute reference frequencies for NMR nuclei) to predict the position of the reference frequency with the current solvent, spectral window, and spectrometer frequency. `setref` assumes a locked sample.

The macro calculates the (auxiliary) 2H reference frequency (TMS-d1) from the lock frequency (`lockf = lockfreq + lkof/1e6`) as follows:

`H2_TMSfreq = lockf / (1 + solppm/1e6)`

then takes the Ξ values for ^{2}H and `tn` and calculates the auxiliary reference frequency (`reffrq`) for the observe nucleus at the given field strength:

`reffrq = (H2_TMSfreq / Ξ(H2)) * Ξ(tn)`

from this, `rfl` and `rfp` are set:

`rfp=0  rfl = sw/2 - (sfrq - reffrq)*1e6.`

Setting the global (or local) flag `bioref ='y'` uses Bio-NMR referencing (based on `nuctables/nuctabrefBio`) rather than standard IUPAC / organic chemistry referencing (based on `nuctables/nuctabref`)

Ξ is the normalized frequency such that the ^{1}H signal from TMS is 100.00 MHz.

This estimate of the frequency based upon the chemical shift value of the lock signal and does not account for temperature, pH, or other factors affecting the chemical shift of the lock solvent.

The default tree is `'current'`.

Arguments An argument and return values are beneficial for the use of `setref` within other macros such as `setref1` and `setref2`. By default (i.e., without an argument), `setref` calculates the referencing for 1D spectra or for the directly detected dimension in nD spectra (f2 in 2D, f3 in 3D).

When only `nucleus` is used as an argument, `setref` returns values without setting parameters.

`$rfl,$rfp,$reffrq,$refpos` are return values for reference peak position, reference peak frequency, reference line frequency, and reference line position, respectively.

Examples

```
setref
setref('C13'):$rfl,$rfp
```

See also *NMR Spectroscopy User Guide*

Related

`reffrq`	Reference frequency of reference line (P)
`refpos`	Position of reference frequency (P)
`rfl`	Reference peak position (P)
`rfp`	Reference peak frequency (P)
`rl`	Set reference line in directly detected dimension (M)
`setref1`	Set frequency referencing for 1st indirectly detected dimension (M)
`setref2`	Set frequency referencing for 2nd indirectly detected dimension (M)
`setup`	Set up parameters for basic experiments (M)

tmsref	Reference 1D proton or carbon spectrum to TMS (M)
bioref	Use nuctables/nuctabrefBio) rather than standard IUPAC / organic chemistry

setref1 Set freq. referencing for 1st indirectly detected dimension (M)

Syntax setref1(nucleus)

Description Calculates the referencing for the first indirect dimension (f1) in nD parameters and FID data sets, for samples locked on deuterium, and for the solvent specified by the solvent parameter. setref1 uses the setref macro to calculate the reference frequency and based on the chemical shift of the lock solvent line and /vnmr/nuctables/nuctabref (absolute reference frequencies for NMR nuclei) to predict the referencing in f1 (reffrq1, rfl1, rfp1) with the current solvent, sw1, and for the frequency of the specified nucleus.

This estimate of the frequency based upon the chemical shift value of the lock signal, as in setref, and does not account for temperature, pH, or other factors affecting the chemical shift of the lock solvent. Using setref, setref1, and setref2, maintains a consistent reference for all dimensions.

Ξ is the normalized frequency such that the ^{1}H signal from TMS is 100.00 MHz.

Setting the global (or local) flag bioref ='y' uses bio-NMR referencing (based on nuctables/nuctabrefBio) rather than standard IUPAC / organic chemistry referencing (based on nuctables/nuctabref)

See /vnmr/nuctables/nuctabref.

Arguments nucleus is the frequency-relevant nucleus in f1.

Examples
```
setref1(tn)
setref1('C13')
```

See also *NMR Spectroscopy User Guide*

Related

reffrq1	Reference frequency of reference line in 1st indirect dimension (P)
refpos1	Position of reference frequency in 1st indirect dimension (P)
rfl	Reference peak position (P)
rfl1	Reference peak position in 1st indirectly detected dimension (P)
rfp1	Reference peak frequency in 1st indirectly detected dimension (P)
setref	Set frequency referencing (M)
bioref	Use nuctables/nuctabrefBio

setref2 Set freq. referencing for 2nd indirect detected dimension (M)

Syntax `setref2(nucleus)`

Description Calculates the referencing for the second indirect dimension (f2) in nD parameters and FID data sets, for samples locked on deuterium, and for the solvent specified by the `solvent` parameter. `setref2` uses `setref` to calculate the reference frequency and based on the chemical shift of the lock solvent line and `/vnmr/nuctables/nuctabref` (absolute reference frequencies for NMR nuclei) to predict the referencing in f2 (`reffrq2`, `rfl2`, `rfp2`) with the current solvent, `sw2`, and for the frequency of the specified nucleus.

This estimate of the frequency based upon the chemical shift value of the lock signal, as in `setref`, and does not account for temperature, pH, or other factors affecting the chemical shift of the lock solvent. Using `setref`, `setref1`, and `setref2`, maintains a consistent reference for all dimensions.

Setting the global (or local) flag `bioref` ='y' uses bio-NMR referencing (based on `nuctables/nuctabrefBio`) rather than standard IUPAC / organic chemistry referencing (based on `nuctables/nuctabref`)

See `/vnmr/nuctables/nuctabref`.

Arguments `nucleus` is the frequency-relevant nucleus in f2.

Examples
```
setref2(tn)
setref2('C13')
```

See also *NMR Spectroscopy User Guide*

Related

`reffrq2`	Reference frequency of reference line in 2nd indirect dimension (P)
`refpos2`	Position of reference frequency in 2nd indirect dimension (P)
`rfl2`	Reference peak position in 2nd indirectly detected dimension (P)
`rfp2`	Reference peak frequency in 2nd indirectly detected dimension (P)
`rl2`	Set reference line in 2nd indirectly detected dimension (M)
`setref`	Set frequency referencing (M)
`bioref`	Use `nuctables/nuctabrefBio`

setscout Set up a scout run (M)

Applicability Systems with LC-NMR accessory.

Description Designed to help run simple experiments during the setup phase of LC-NMR or to be the first of two experiments run on peaks in a stopped-flow or loop-flushing mode. In the latter application, you can set `wexp='setwet au'` so that the scout run is analyzed, parameters adjusted, and an appropriate solvent-suppressed experiment run.

If parameters already exist in the current experiment for performing the `lc1d` pulse sequence, `setscout` turns off the solvent suppression portion of the sequence; if they do not exist, they are created and set to default values using `lc1d`.

See also *NMR Spectroscopy User Guide*

Related

`lc1d`	Pulse sequence for LC-NMR (M)
`setwet`	Set up a solvent-suppressed experiment (M)

setssfilter Set sslsfrq to the frequencies of each suppressed solvents (M)

Applicability Systems with LC-NMR accessory.

Description Sets `sslsfrq` to the frequencies of each of the suppressed solvents.

See also *NMR Spectroscopy User Guide*

setsw Set spectral width (M)

Syntax `setsw(downfieldppm,upfieldppm)`

Description Sets `sw` and `tof` for the given spectral window and also does referencing.

Arguments `downfieldppm` is the downfield frequency, in ppm.

`upfieldppm` is the upfield frequency, in ppm.

Examples

```
setsw(12,0)
setsw(235,-15)
```

See also *NMR Spectroscopy User Guide*

Related

`setsw1`	Set spectral width in evolution dimension (M)
`setsw2`	Set spectral width in 2nd evolution dimension (M)
`sw`	Spectral width in directly detected dimension (P)
`tof`	Frequency offset for observe transmitter (P)

setsw1 Set spectral width in evolution dimension (M)

Syntax `setsw1(nucleus,downfieldppm,upfieldppm):offset`

Description Sets `sw1` for the given spectral window and also does referencing.

Arguments `nucleus` returns the nucleus.

`downfieldppm` is the downfield frequency, in ppm.

`upfieldppm` is the upfield frequency, in ppm.

`offset` returns the appropriate offset.

Examples `setsw1(tn,12,0)`
`setsw1(dn,235,-15):dof`

See also *NMR Spectroscopy User Guide*

Related		
	setsw	Set spectral width (M)
	sw1	Spectral width in 1st indirectly detected dimension (P)

setsw2 Set spectral width in 2nd evolution dimension (M)

Syntax `setsw2(nucleus,downfieldppm,upfieldppm):offset`

Description Sets sw2 for the given spectral window and also does referencing.

Arguments `nucleus` returns the nucleus.

`downfieldppm` is the downfield frequency, in ppm.

`upfieldppm` is the upfield frequency, in ppm.

`offset` returns the appropriate offset.

Examples `setsw2(tn,12,0)`
`setsw2(dn,235,-15):dof`

See also *NMR Spectroscopy User Guide*

Related		
	setsw	Set spectral width (M)
	sw2	Spectral width in 2nd indirectly detected dimension (P)

setselfrqc Set selective frequency and width (M)

Description Sets selective frequency and width of the excitation bandwidth for selective excitation. Used after TOCSY1D and Noesy1d selection. Selected frequencies and widths of the excitation bandwidth are used by suselfrq.

Related		
	Noesy1d	Change parameters for NOESY1D experiment (M)
	suselfrq	Select peak, continue selective excitation experiment (M)
	TOCSY1D	Change parameters for TOCSY1D experiment (M)

setselinv Set up selective inversion (M)

Description Sets power, pulsewidth, and shape for selective inversion; used by suselfrq. By default, `setselinv` selects a q3 gaussian cascade pulse if a waveform generator or linear modulator is present. Otherwise, `setselinv` selects a “rectangular” pulse.

Related	`setselfrqc`	Select selective frequency and width (M)
	`suselfrq`	Select peak, continue selective excitation experiment (M)

`settcldefault` Select default display templates for pulse sequence (M)

Syntax	`settcldefault<(<default><,sequence>)>`
Description	Selects the display templates to use as the default for a pulse sequence.
Arguments	`default` is the name of the set of display templates to use for the default display of the current pulse sequence (defined by the parameter `seqfil`). If no arguments are given, the user is prompted for the name of the display templates.
	`sequence` defines which pulse sequence will use the default displays of the pulse sequence given as the first argument. The default is the pulse sequence defined by the parameter `seqfil`.

Examples	`settcldefault`
	`settcldefault('cosy')`
	`settcldefault('default2d','HMQC8')`

See also	*User Programming*
Related	`seqfil` Pulse sequence name (P)

`settune` Opens the Auto Tune Setup dialog (M)

Applicability	Automation, VnmrJ Walkup
Syntax	`settune`
Description	Opens a dialog for setting when to tune in automation using ProTune.
See also	*NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related	`protune`	Macro to start ProTune (M)
	`wtune`	Specify when to tune (P)

`settype` Change type of a parameter (C)

Syntax	`settype(parameter,type<,tree>)`
Description	Changes the type of an existing parameter. A string parameter can be changed into a string or flag type, or a real parameter can be changed into a real, delay, frequency, pulse, or integer type. Note that `settype` cannot change a string parameter into a real, or change a real into a string.

Arguments parameter is the name of an existing parameter.

type is one of the keywords 'string', 'flag', 'real', 'delay', 'frequency', 'pulse', or 'integer'.

tree is one of the keywords 'global', 'current', 'processed', or 'systemglobal'. The default is 'current'. Refer to the create command for more information on the types of parameter trees.

Examples settype('in','flag','global')
settype('p12','pulse')

See also *User Programming*

Related

create	Create new parameter in a parameter tree (C)
display	Display parameters and their attributes (C)
setgroup	Set group of a parameter in a tree (C)
setlimit	Set limits of a parameter in a tree (C)
setprotect	Set protection mode of a parameter (C)
setvalue	Set value of any parameter in a tree (C)

setup Set up parameters for basic experiments (M)

Syntax setup<(nucleus<,solvent>)>

Description Returns a parameter set to do the experiment requested, complete with positioning of the transmitter and decoupler. Parameters set by setup are recalled from the /vnmr/stdpar directory or from the user's stdpar directory if the appropriate file exists there. Any changes made to the files in these directories are reflected in setup. The default parameters for carbon and proton survey spectra are in files /vnmr/stdpar/C13.par and /vnmr/stdpar/H1.par, respectively. These files should be modified as desired to produce spectra under desirable conditions.

Arguments nucleus is a nucleus chosen from the files in /vnmr/stdpar or in the user's stdpar directory (e.g., 'H1', 'C13', 'P31').

solvent is a solvent chosen from the file /vnmr/solvents (e.g., 'CDCl3', 'C6D6', 'D2O'). The default is 'CDCl3'.

Examples setup
setup('H1')
setup('C13','DMSO')

See also *NMR Spectroscopy User Guide*

setup_dosy Set up gradient levels for DOSY experiments (M)

Description Initiates a dialogue to set up an array of gzlvl1 values for DOSY experiments. setup_dosy requests the number of array increments and an initial and a final gzlvl1 value and sets up an array that gives increments in gzlvl1 squared between these limits. setup_dosy retrieves the gradient strength from the probe calibration file if `probe<>''` and stores it in the local experimental parameter `DAC_to_G`. If `probe=''` (i.e., the probe is not defined), then `DAC_to_G` is set to the current value of the global parameter `gcal`.

See also *NMR Spectroscopy User Guide*

Related		
	`dosy`	Process DOSY experiments (M)
	`DAC_to_G`	Parameter to store gradient calibration value in DOSY sequences (P)
	`setgcal`	Set the gradient calibration constant (M)

setuserpsg — Creates/initializes user PSG directory

Syntax `setuserpsg`

Applicability VnmrJ 3.1

Description SETUSERPSG is a UNIX shellscript which performs the following functions:

- creates the user PSG directory if one does not already exist;
- and initializes the user PSG directory with the appropriate PSG object libraries from the system PSG directory, if necessary.

For reference, the user PSG object library in the system PSG directory is LIBPSGLIB.A; and the Agilent PSG object library in the same directory is LIBPARAM.A. SETUSERPSG is automatically invoked by the shellscript PSGGEN.

`setvalue` — Set value of any parameter in a tree (C)

Syntax `setvalue(parameter,value<,index><,tree>)`

Description Sets the value of any parameter in a tree. This command bypasses the normal range checking for parameter entry, as well as bypassing any action that would be invoked by the parameter's protection mode (see the `setprotect` command). If the parameter entry normally causes a `_parameter` macro to be executed, this action also is bypassed.

Arguments `parameter` — name of the parameter.

`value` —set value for the parameter.

`index` — number of a single element in an arrayed parameter. The default is 1. A value of 0 for the index resets an arrayed (or non-arrayed) parameter to the one element supplied as the second argument to `setvalue`.

`tree` — keyword `'global'`, `'current'`, `'processed'`, or `'systemglobal'`. The default is `'current'`. Refer to the `create` command for more information on the types of parameter trees.

Examples `setvalue('arraydim',128,'processed')`

See also	*User Programming*	
Related	create	Create new parameter in a parameter tree (C)
	setprotect	Set protection mode of a parameter (C)

setwave Write a wave definition string into Pbox.inp file (M)

Syntax `setwave('sh bw/pw ofs st ph fla trev d1 d2 d0')`

Description Sets up a single excitation band in the `Pbox.inp` file. An unlimited number of waves can be combined by reapplying `setwave`.

Arguments A single string of 1 to 10 wave parameters in predefined order. Note that a single quote is required at the start and the end of the entire string, but no single quotes are required surrounding characters and strings inside the entire string.

`sh`	name of a shape file.
`bw/pw`	either the bandwidth, in Hz, or the pulsewidth, in sec.
`ofs`	offset, in Hz.
`st`	number specifying the spin status: 0 for excitation 1 for de-excitation 0.5 for refocusing.
`ph`	phase (or phase cycle, see `wavelib/supercycles`).
`fla`	flip angle. `fla` can override the default flip angle.
`trev`	time reversal. This can be used to cancel time reversal if spin status (`st`) is set to 1 for Mxy.
`d1`	delay, in sec, prior the pulse.
`d2`	delay, in sec, after the pulse.
`d0`	delay or command prior to `d1`. If `d0=a`, the wave is appended to the previous wave.

Examples
```
setwave('eburp1')
setwave('GARP 12000.0')
setwave('esnob 600 -1248.2 1 90.0 n n 0.001')
```

See also *NMR Spectroscopy User Guide*

Related Pbox Pulse shaping software (U)

setwell Adjust the label of the "t1" axis for VAST contour maps

Applicability VnmrJ 3.1

Description The setwell macro sets the label of the vertical axis in contour plots to "well" (instead of seconds).

See also `plateglue`
`vastglue`

setwin — Activate selected window (C)

Syntax `setwin(row<,column>)`

Description Activates a specific pane in the graphics window. Panes are numbered sequentially from left to right and top to bottom.

Arguments `row` is the number of the row containing the pane to be activated.

`column` is the number of the column containing the pane to be activated.

Examples
```
setwin(3)
setwin(1,2)
```

See also *NMR Spectroscopy User Guide*

Related

curwin	Current window (P)
fontselect	Open FontSelect window (C)
jwin	Activate current window (M)
mapwin	List of experiment numbers (P)
setgrid	Activate selected window (M)

sf — Start of FID (P)

Description Sets the start of the FID display. This parameter can be entered in the usual way or interactively controlled by the sf wf button during a FID display.

Values `0` to the value of `at`, in seconds.

See also *NMR Spectroscopy User Guide*

Related

at	Acquisition time (P)
dcon	Display noninteractive color intensities map (C)
dconi	Interactive 2D data display (C)
df	Display a single FID (C)
sf1	Start of interferogram in 1st indirectly detected dimension (P)
sf2	Start of interferogram in 2nd indirectly detected dimension (P)
vf	Vertical scale of FID (P)
wf	Width of FID (P)

sf1 — Start of interferogram in 1st indirectly detected dimension (P)

Description Sets the start of the interferogram display in the first indirectly detected dimension.

Values `0` to (2 × `ni`)/`sw1`, in seconds.

See also *NMR Spectroscopy User Guide*

Related

ni	Number of increments in 1st indirectly detected dimension (P)
sf	Start of FID (P)
sw1	Spectral width in 1st indirectly detected dimension (P)
wf1	Width of interferogram in 1st indirectly detected dimension (P)

sf2 Start of interferogram in 2nd indirectly detected dimension (P)

Description Sets the start of the interferogram display in the second indirectly detected dimension.

Values `0` to (2 × `ni2`)/`sw2`, in seconds.

See also *NMR Spectroscopy User Guide*

Related

ni2	Number of increments in 2nd indirectly detected dimension (P
sf	Start of FID (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)
wf2	Width of interferogram in 2nd indirectly detected dimension (P)

sfrq Transmitter frequency of observe nucleus (P)

Description Contains the frequency for the observe transmitter. `sfrq` is automatically set when `tn` is changed, and it should not be necessary for the user to manually set this parameter.

Values Number, in MHz.

See also *NMR Spectroscopy User Guide*

Related

dfrq	Transmitter frequency of first decoupler (P)
dfrq2	Transmitter frequency of second decoupler (P)
dfrq3	Transmitter frequency of third decoupler (P)
tn	Nucleus for observe transmitter (P)
tof	Frequency offset for observe transmitter (P)
spcfrq	Display frequencies of rf channels (M)

sh2pul Set up for a shaped observe excitation sequence (M)

Applicability Systems with waveform generators.

Syntax `sh2pul`

Description Behaves like standard two-pulse sequence S2PUL but with the normal hard pulses changed into shaped pulses from the waveform generator.

The name of the shaped pulse associated with `pw` is `pwpat` and `p1` is `p1pat`. Information about the specifics of power settings and bandwidths is available from the macros `bandinfo` and `pulseinfo`.

See also *User Programming*

Related		
	`bandinfo`	Shaped pulse information for calibration (M)
	`p1pat`	Shape of an excitation pulse (P)
	`pwpat`	Shape of refocusing pulse (P)
	`pulseinfo`	Shaped pulse information for calibration (M)

shdec — Set up for shaped observe excitation sequence (M)

Applicability: Systems with waveform generators.

Description: Sets up the SHDEC pulse sequence that generates a shaped pulse on the observe channel using the waveform generator. It also allows for programmed (e.g.: multiselective) homodecoupling or solvent presaturation using the observe transmitter, and an optional gradient pulse following the excitation pulse.

See also: *NMR Spectroscopy User Guide*

Related		
	`Pbox`	Pulse shaping software (U)

shell — Start a UNIX shell (C)

Syntax: `shell<(command)>:$var1,$var2,...`

Description: Brings up a normal UNIX shell for the user. On the Sun, a pop-up window is created. On the GraphOn terminal, the entire terminal is used.

Arguments: `command` is a UNIX command line to be executed by `shell`. The default is to bring up a UNIX shell. If the last character in the command line is the symbol &, the command is executed in background, which allows commands to be entered and executed while the `shell` command is still running. Note that if this background feature is used, any printed output should be redirected to a file. Otherwise, the output may pop up in the text window at random times.

`shell` calls involving pipes or input redirection (<) require either an extra pair of parentheses or the addition of`; cat` to the `shell` command string.

`$var1, $var2,...` are names of variables to hold text lines that are generated as a result of the UNIX command. The default is to display the text lines. Each variable receives a single display line. `shell` always returns a text line; in many cases, it is a simple carriage return. To prevent this carriage return from being shown, capture it in a dummy variable, such as

`shell('command'):$dum`

Examples
```
shell
shell('ps')
shell('ls -lt'):$filelist
shell(systemdir+'/acqbin/Acqstat '+hostname+' &')
shell('ls -t|grep May; cat')
```
or
```
shell('(ls -t|grep May)')
```

See also *NMR Spectroscopy User Guide, User Programming*

Related `shelli` Start an interactive UNIX shell (C)

`shelli` Start an interactive UNIX shell (C)

Syntax `shelli(command)`

Description On a terminal, runs interactively the UNIX command line given as the argument. No return or output variables are allowed.

Arguments `command` is a UNIX command line to be executed.

Examples `shelli('vi myfile')`

See also *NMR Spectroscopy User Guide, User Programming*

Related `shell` Start a UNIX shell (C)

`shim` Submit an Autoshim experiment to acquisition (C)

Description Performs validity checks on the acquisition parameters and then submits an Autoshim experiment to acquisition.

See also *NMR Spectroscopy User Guide*

Related
`au`	Submit experiment to acquisition and process data (C)
`change`	Submit a change sample experiment to acquisition (M)
`ga`	Submit experiment to acquisition and FT the result (C)
`go`	Submit experiment to acquisition (C)
`lock`	Submit an Autolock experiment to acquisition (C)
`sample`	Submit change sample, autoshim experiment to acquisition (M)
`spin`	Submit a spin setup experiment to acquisition (C)
`su`	Submit a setup experiment to acquisition (M)

`shimmult` Multiple the shim dacs of the current shimset

Syntax `shimmult<(multiplier)>`

Applicability VnmrJ 3.1

Description The `shimmult` macro will multiply the value of each dac in the current shimset by a multiplier. The multiplier may be supplied as an argument. The default value is 1.0/1.5. One might use this macro if the current output by the shim power supply has changed. This macro does not load the new values into the hardware. Follow the shimmult macro with a call to "`su`" to set the hardware. Note also that shim dac values are integer values. Therefore, shimmult(1/3) followed by shimmult(3) may not give the original values, do to truncation effects.

shimnames Returns shim names

Syntax `shimnames<:$names,$num>`

Applicability VnmrJ 3.1

Description This command returns a list of the names of the active shims. These are returned in a single string parameter. A second argument will return the number of active shims. The `substr` command can be used to extract individual shim names from the returned list.

Arguments

Examples `shimnames:$names,$num`

shimset Type of shim set (P)

Description Configuration parameter for the type of shims on the system. The value of `shimset` is set using the Shimset label in the Spectrometer Configuration window.

Values `1` to `14`, where the value identifies one of the following shim sets:

`1` is a shim set in a Agilent 13-shim supply with computer-controlled axial shims z1, z1c, z2, z2c, z3, z4, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3. Shims can be adjusted from –2047 to +2047. This value is used with the Ultra•nmr shim system when operated from the HIM box (Agilent 13 Shims choice in Spectrometer Configuration window).

`2` is a shim set in a Oxford 18-shim supply with computer-controlled axial shims z1, z1c, z2, z2c, z3, z4, z5, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2. Shims can be adjusted from –2047 to +2047 (Oxford 18 Shims choice in Spectrometer Configuration window).

`3` is a shim set in a Agilent 23-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, z6, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy. Shims can be adjusted from
–32767 to +32767 (Agilent 23 Shims choice in Spectrometer Configuration window).

`4` is a shim set in a Agilent 28-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, z6, z7, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x, z4y. Shims can be adjusted from –32767 to +32767 (Agilent 28 Shims choice in Spectrometer Configuration window).

5 is a shim set in an Ultra•nmr shim system (39 shim channels) with computer-controlled axial shims z1, z1c, z2, z2c, z3, z3c, z4, z4c, z5, z6, z7, z8, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x, z4y, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Ultra Shims choice in Spectrometer Configuration window).

6 is a shim set in a Agilent 18-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2. Shims can be adjusted from –32767 to +32767 (Agilent 18 Shims choice in Spectrometer Configuration window).

7 is a shim set in a Agilent 20-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y. Shims can be adjusted from –32767 to +32767 (Agilent 20 Shims choice in Spectrometer Configuration window).

8 is a shim set in a Oxford 15-shim supply with computer-controlled axial shims z1, z2, z3, z4, and radial shims x1, y1, xz, yz, xy, x2y2, zx2y2, xz2, yz2, zxy. Shims can be adjusted from –2047 to +2047 (Oxford 15 Shims choice in Spectrometer Configuration window).

9 is a shim set in a Agilent Ultra•nmr shim system II (40 shim channels) with computer-controlled axial shims z1, z1c, z2, z2c, z3, z3c, z4, z4c, z5, z6, z7, z8, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, x4, y4, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x, z4y, z3x2y2, z3xy, z2x3, z2y3, z3x3, z3y3, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from
–32767 to +32767 (Agilent 40 Shims choice in Spectrometer Configuration window).

10 is a shim set in a Agilent 14-shim supply with computer-controlled axial shims z1, z1c, z2, z2c, z3, z4, z5, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3. Shims can be adjusted from –2047 to +2047 (Agilent 14 Shims choice in Spectrometer Configuration window).

11 is a shim set in a Agilent 8-shim supply with computer-controlled axial shims z1, z2, and radial shims x1, y1, xz, yz, xy, x2y2. Shims can be adjusted from –32767 to +32767 (Whole Body Shims choice in Spectrometer Configuration window).

12 is a shim set in a Agilent 26-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, x4, y4. Shims can be adjusted from –32767 to +32767 (Agilent 26 Shims choice in Spectrometer Configuration window).

13 is a shim set in an Agilent 29-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, z6, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x,

z4y, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Agilent 29 Shims choice in Spectrometer Configuration window).

14 is a shim set in a Agilent 35-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, z6, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, x4, y4, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x, z4y, z3x2y2, z3xy, z4x2y2, z4xy, z5x, z5y. Shims can be adjusted from –32767 to +32767 (Agilent 35 Shims choice in Spectrometer Configuration window).

15 is the Agilent 15 Shim.

16 is the Ultra 18 Shims.

17 is a shim set in an Agilent 15-shim supply with computer-controlled axial shims z1, z2, z3, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zx2y2, zxy. Shims can be adjusted from -32767 to +32767 (Agilent Combo Shims choice in Spectrometer Configuration window).

18 is a shim set in an Agilent 28-shim supply with computer-controlled axial shims z1, z2, z3, z4, z5, z6, z7, and radial shims x1, y1, xz, yz, xy, x2y2, x3, y3, xz2, yz2, zxy, zx2y2, z3x, z3y, z2x2y2, z2xy, zx3, zy3, z4x, z4y. Shims can be adjusted from -32767 to +32767 (Agilent 28 Thin Shims choice in Spectrometer Configuration window).

See also *VnmrJ Installation and Administration*

Related	`config`	Display current configuration and possibly change it (M)
	`readhw`	Read current values of acquisition hardware (C)

`showconfig` Show system configuration settings (M)

See also Displays the system configuration settings in the text window. To print the settings, enter the following in the VnmrJ command line: **`printon showconfig printoff`**.

See also *VnmrJ Installation and Administration*

Related	`config`	Display current configuration and possibly change it (M

`showconsole` Show system configuration settings (U)

Description Displays console hardware configuration parameters and system versions. This information is recorded during console bootup and represents the system hardware options recognized by the acquisition computer. The command is used mainly when troubleshooting or performing diagnostics.

See also *NMR Spectroscopy User Guide*

Related	`ihwinfo`	Hardware status of console (C)

showdosy Show DOSY Plot (M)

Syntax showdosy(<expno>)

Applicability VnmrJ 3.1

Description The macro 'showdosy' is a convenient way of displaying the pseudo 2D DOSY spectrum. Typing showdosy(N) after the completion of the "dosy" macro joins experiment N and displays the DOSY spectrum automatically. It sets fn1=256 and fn=8k, which can be adjusted to achieve better resolution.

Arguments 'expno' experiment number to display the DOSY plot.

Related dosy
ddif

showdosyfit Plots the experimental signal attenuation, fitted attenuation and residual for one peak from a 2D or 3D DOSY experiment

Syntax showdosyfit(peaknr)
showdosyfit(peaknr,expFac)

Applicability VnmrJ 3.1

Description Displays using expl the result of fitting peak peaknr using dosy. Experimental data points are in red, fitted points in blue, and residuals in magenta.

Arguments The macro takes one or two arguments (peaknr, expFac), which are the peak number and the expansion factor of the residual respectively. When expansionfactor is not given it defaults to 1.

See also dosy

showdosyresidual Plots the residual for one peak from a 2D or 3D DOSY experiment

Syntax showdosyresidual(peaknr)
showdosyresidual(peaknr,expFac)

Applicability VnmrJ 3.1

Description Displays using expl the residuals of fitting peak peaknr using dosy.

Arguments The macro takes one or two arguments (peaknr, expFac), which are the peak number and the expansion factor of the residual respectively. When expansionfactor is not given it defaults to 1.

See also dosy

showgradfit Plots the experimental gradient variation with position and the power series fit in non-uniform gradient calibration.

Syntax `showgradfit`

Applicability VnmrJ 3.1

Description Displays (using expl) the result of fitting the experimental variation of gradient strength with position, measured during non-uniform gradient calibration, and the result of fitting with a power series.Experimental data points are in red and fitted points in blue.

Arguments

Examples

See also `gradfit`
`nugcalib`
`powerfit`
`shownugfit`

showfit Display numerical results of deconvolution (M)

Description After a deconvolution, the results are written into file `fitspec.outpar` in an abbreviated format. `showfit` converts these data to an output format more suitable for examination and printing.

See also *NMR Spectroscopy User Guide*

Related	`fitspec`	Perform spectrum deconvolution (C)
	`plfit`	Plot deconvolution analysis (M)
	`usemark`	Use “mark” output as deconvolution starting point (M)

showloginbox Shows operator login dialog (M)

Description Shows the login dialog for operators.

shownugfit Plots the logarithm of the calculated diffusional attenuation and of the power series fit in non-uniform gradient calibration

Syntax `shownugfit`

Applicability VnmrJ 3.1

Description Displays (using expl) the result of fitting the calculated signal attenuation as a function of gradient squared to the exponential of a power series. Calculated data points are in red and fitted points in blue.

See also gradfit

nugcalib

powerfit

shownugfit

shownumx Show x position of number (P)

Description Show the **X** position of the number. The bottom left of every spectrum is defined as 0.

See also *User Programming*

Related shownumy y position counting from bottom left of every spectrum (P)

shownumy Show y position of number (P)

Description Show the **Y** position of the number. The bottom left of every spectrum is defined as 0.

See also *User Programming*

Related shownumx x position counting from bottom left of every spectrum (P)

showoriginal Restore first 2D spectrum in 3D DOSY experiment (M)

Description Restores the first 2D spectrum in a 3D DOSY experiment (if it has been saved by the dosy macro).

See also *NMR Spectroscopy User Guide*

Related dosy Process DOSY experiments (M)

showplotter Show list of currently defined plotters and printers (M)

Description Shows a list of currently defined plotters and printers.

See also *NMR Spectroscopy User Guide*

Related `plotter` Plotter device (P)
`printer` Printer device (P)

showplotq Display plot jobs in plot queue (M)

Description Displays current plot jobs in the plot queue for the active plotter.

See also *NMR Spectroscopy User Guide*

Related `killplot` Stop plot jobs and remove from plot queue (C
`showprintq` Display print jobs in print queue (C)

showprintq Display print jobs in print queue (M)

Description Displays current print jobs in the print queue for the active printer.

See also *NMR Spectroscopy User Guide*

Related `killprint` Stop print jobs and remove from print queue (C)
`showplotq` Display plot jobs in plot queue (M)

showprotuneguishow the graphical interface while tuning (P)

Syntax `showprotunegui='argument'`

Description This is a global string parameter that does not exist by default. The user can create it to force the ProTune GUI to be shown during normal tuning operation.

Arguments `'n'` — Do not force the GUI to be shown.

`'y'` — Show the GUI, except in automation.

`'a'` — Always show the GUI, even in automation.

Set `showprotunegui='a' will` cause ProTune to fail in automation unless the proper display permission has been set. Set the display permissions on Linux systems by executing "`xhost local:`" on the Linux command line.

See also *NMR Spectroscopy User Guide*

Related `protune` Macro to start ProTune (M)

showrfmon Show RF Monitor Button in Hardware Bar (P)

Applicability Imaging

Syntax showrfmon=<value>

Description Show RF Monitor Button in Hardware Bar.

Values 1 show RF Monitor button.

-1 hide RF Monitor button.

See also *VnmrJ Imaging User Guide*

showsampglobal Shows sample global parameters

Description Shows sample global parameter values in current workspace.

Syntax showsampglobal

Related getsampglobal, resetsampglobal, savesampglobal, mvsampglobal

showstat Display information about status of acquisition (M,U)

Syntax (From VnmrJ) showstat<(remote_system)>
(From UNIX) showstat <remote_system>

Description Displays information in the text screen about the status of acquisition on a spectrometer. The command is similar to Acqstat, but displays the information in a non-graphical manner and only once.

Arguments remote_system is the host name of a remote spectrometer. The default is to display information about acquisition on the local system.

See also *NMR Spectroscopy User Guide*

Related Acqstat Bring up the acquisition status display (U)

sim Sample in magnet (For systems equipped with a robot)

Syntax sim

Applicability VnmrJ 3.1

Description The sim macro generates a pop-up window to set the number of the sample currently in the magnet. The sim macro is only available for systems with a robot. This macro would typically be used only after a manual insert of a sample. In this case, the NMR console is unaware of the proper location of the inserted sample. The sim macro allows a location to be assigned to the inserted sample. In addition to assigning the number of the sample in the magnet, the "Sample in Magnet" popup can also be used to remove the current sample or to replace the current sample.

sin — Find sine value of an angle (C)

Syntax `sin(angle)<:n>`

Description Finds the sine value of an angle.

Arguments `angle` is the angle given in radians.

`n` is a return value giving the sine of `angle`. The default is to display the sine value in the status window.

Examples
```
sin(.5)
sin(val):sin_val
```

See also *User Programming*

Related

asin	Find arc sine of number (C)
atan	Find arc tangent of a number (C)
cos	Find cosine value of an angle (C)
exp	Find exponential value (C)
ln	Find natural logarithm of a number (C)
tan	Find tangent value of an angle (C)

sine — Find values for a sine window function (M)

Syntax `sine<(shift<,number_points<,domain>)>`

Description Calculates appropriate values for parameters `sb` and `sbs` (if the `domain` argument is `'f2'`) or for parameters `sb1` and `sbs1` (if the `domain` argument is `'f1'`) in order to achieve a sine window function. The value of the parameter `trace` is used if the `domain` argument is not entered.

Arguments If `shift` is greater than 1, the `sbs` parameter is calculated as `2*sb/shift` (`sbs1` is calculated as `2*sb1/shift`). `sine(2)` gives a “PI/2-shifted” sine window, i.e., cosine weighting. `sine(3)` gives a “PI/3” shifted sine window, etc. If `shift` is less than or equal to 1, an unshifted sine window is used (`sbs='n'` or `sbs1='n'`).

`number_points` specifies the number of real points that the window function spans. The value of the window function for subsequent points is 0. `number_points` must be greater than 0 and a multiple of 2. The default is `ni*2` if `trace='f1'`, or `np` if `trace='f2'`.

`domain` is `'f1'` or `'f2'`. The default is the current setting of `trace`.

See also *NMR Spectroscopy User Guide*

Related

np	Number of data points (P)
sb	Sinebell const. in directly detected dimension (P)
sb1	Sinebell const. in 1st indirectly detected dimension (P)
sbs	Sinebell shift const. in directly detected dimension (P)
sbs1	Sinebell shift const. in 1st indirectly detected dimension (P)
sinesq	Find values for a sine squared window function (M)
trace	Mode for *n*-dimensional data display (P)

sinebell Select default parameters for sinebell weighting (M)

Description Generates initial guess at good sinebell weighting parameters by setting the sb and sb1 parameters to one-half the acquisition time and turning off all other weighting. Use sinebell in absolute-value 2D experiments only.

See also *NMR Spectroscopy User Guide*

Related

pseudo	Set default parameters for pseudo-echo weighting (M)
sb	Sinebell const. in directly detected dimension (P)
sb1	Sinebell const. in 1st indirectly detected dimension (P)

sinesq Find values for a sine-squared window function (M)

Syntax `sinesq<(shift<,number_points<,domain>)>`

Description Calculates appropriate values for parameters sb and sbs (if the domain argument is 'f2') or for parameters sb1 and sbs1 (if the domain argument is 'f1') in order to achieve a sine-squared window function. The value of parameter trace is used if the domain argument is not entered.

Arguments shift sets the starting value for the window function. If shift is greater than 0, the starting value is given by sin p/shift; otherwise, if shift is less than or equal to 0, the starting value is 0. The default value is 0.

number_points specifies the number of real points that the window function spans. The value of the window function for subsequent points is 0. The number_points argument must be greater than 0 and a multiple of 2. The default is ni*2 if trace='f1', or np if trace='f2'.

domain is 'f1' or 'f2'. The default is the current setting of trace.

See also *NMR Spectroscopy User Guide*

Related

ni	Number of increments in 1st indirectly detected dimension (P)
np	Number of data points (P)
sb	Sinebell const. in directly detected dimension (P)
sb1	Sinebell const. in 1st indirectly detected dimension (P)
sbs	Sinebell shift const. in directly detected dimension (P)
sine	Find values for a sine window function (M)
trace	Mode for *n*-dimensional data display (P)

size — Returns the number of elements in an arrayed parameter (O)

Description
: In MAGICAL programming, an operator that returns the number of elements in an arrayed parameter.

Examples
: `r1 = size('d2')`

See also
: *User Programming*

Related

`arraydim`	Dimension of experiment (P)
`typeof`	Return identifier for argument type (O)
`length`	Determine length of a string (C)

slfreq — Measured line frequencies (P)

Description
: Contains a list of measured line frequencies. In iterative spin simulation, a calculated spectrum is matched to the lines in the list. The `spinll` macro fills in `slfreq` from the last line listing or a `mark` operation. Use `assign` to make assignments between the measured lines and the calculated transitions. `slfreq` is a global parameter and is displayed by `dla`.

See also
: *NMR Spectroscopy User Guide* .

Related

`assign`	Assign transitions to experimental lines (M)
`cla`	Clear all line assignments (M)
`dla`	Display spin simulation parameter arrays (M)
`fitspec`	Perform spectrum deconvolution (C)
`mark`	Determine intensity of a spectrum at a point (C)
`spinll`	Set up an `slfreq` array (M)

slw — Spin simulation linewidth (P)

Description
: Sets linewidth for individual transitions in the displayed spectrum. Only one linewidth is provided, so all transitions must be given the same linewidth. If the Set Params button is used in setting up spin simulation parameters, `slw` is automatically set to the measured linewidth of the tallest line displayed.

: `slw` is also the starting default linewidth for deconvolution calculations. This linewidth will be set automatically when deconvolution is operated using the menu mode and is bypassed if the `usemark` command has been used in conjunction with two cursor input.

Values
: 0.01 to 1e6. The typical value is 1.

See also *NMR Spectroscopy User Guide*

Related `usemark` Use “mark” output as deconvolution starting point (M)

smaxf Maximum frequency of any transition (P)

Description Sets the maximum frequency limit for the calculation of the final simulated spectrum. It should be set before the calculation is performed. If the Set Params button is used in setting up spin simulation parameters, `smaxf` is initialized to `sp`+`wp`; which assumes that you have already expanded the region of the spectrum that you wish to simulate before beginning the spin simulation process.

Values –1e10 to 1e10, in Hz. The typical value is the maximum chemical shift + 50.

See also *NMR Spectroscopy User Guide*

Related
`sminf` Minimum frequency of any transition (P)
`sp` Start of plot (P)
`wp` Width of plot (P)

sminf Minimum frequency of any transition (P)

Description Sets the minimum frequency limit for the calculation of the final simulated spectrum. It should be set before the calculation is performed. If the Set Params button is used in setting up spin simulation parameters, `sminf` is initialized to `sp`, which assumes that you have already expanded the region of the spectrum that you wish to simulate before beginning the spin simulation process.

Values –1e10 to 1e10, in Hz. The typical value is 0.

See also *NMR Spectroscopy User Guide*

Related
`smaxf` Maximum frequency of any transition (P)
`sp` Start of plot (P)
`wp` Width of plot (P)

smsport Sample Management System serial port connection (P)

Description Sets which serial port on the host computer is connected to a Sample Management System (i.e., a sample changer). The value of `smsport` is set using the Sample Changer Serial Port label in the Spectrometer Configuration window.

Values `'a'` sets the connection for serial port A. This value is the default.
`'b'` sets the connection for serial port B.

See also *VnmrJ Installation and Administration; NMR Spectroscopy User Guide*

Related `config` Display current configuration and possibly change it (M)

sn Signal-to-noise ratio (P)

Description Sets a ratio for testing signal-to-noise. The `testsn` macro checks whether a signal-to-noise ratio equal to `sn` has been achieved.

Values Typical value is 35.

See also *NMR Spectroscopy User Guide*

Related
`dsn` Measure signal-to-noise (C)
`getsn` Get signal-to-noise estimate of a spectrum (M)
`testsn` Test signal-to-noise of a spectrum (M)
`testct` Check *ct* for resuming signal-to-noise testing (M)

solppm Return ppm and peak width of solvent resonances (M)

Syntax `solppm:chemical_shift,peak_width`

Description Returns to the calling macro information about the chemical shift and peak spread of solvent resonances in various solvents for either ^{1}H or ^{13}C, depending on the observe nucleus `tn` and the parameter `solvent`. This macro is used “internally” by other macros only.

Arguments `chemical_shift` returns the chemical shift of the solvent in ppm.

`peak_width` returns the approximate peak spread of solvent resonances.

See also *User Programming*

Related
`solvent` Lock solvent (P)
`tn` Nucleus for observe transmitter (P)

solvent Lock solvent (P)

Description Contains one of a series of lock solvents from the `/vnmr/solvents` file, which contains the ^{2}H chemical shift of each lock solvent. By editing the file, additional solvents can be added. Values for `solvent` are not case- sensitive (e.g., `solvent='C6D6'` and `solvent='c6d6'` are identical)

The `auto_dir` macro now controls most of the automation features, including setting the value of `solvent`.

Values Standard values in `/vnmr/solvents` include:

Deuterium Oxide	CDCl3	MethyleneChloride
D2O	Cyclohexane	MethylAlcohol-d4
Acetone	C6Dl2	CD2Cl2
CD3COCD3	Toluene	CD3OD
Benzene	C6D5CH3	Chloroform
C6D6	Acetic_Acid	
DMSO	CD3COOD	

See also *NMR Spectroscopy User Guide*

Related
`lastlk` Last lock solvent used (P)
`solvinfo` Retrieve information from solvent table (C)
`tof` Frequency offset for observe transmitter (P)

solvinfo Retrieve information from solvent table (C)

Syntax `solvinfo(solvent):$chemical_shift,$name`

Description Retrieves solvent shift and solvent name from the solvent table.

Arguments `solvent` is the name of a solvent from the `/vnmr/solvents` file. This argument is not case-sensitive (e.g., `'c6d6'` is the same as `'C6D6'`).

`chemical_shift` returns the chemical shift of the solvent, in ppm.

`name` returns the name of the solvent. The name returned will match the case of the letters (upper or lower) in `/vnmr/solvents`.

Examples
```
solvinfo('acetone'):$shift
solvinfo('d2o'):$shift,solvent
```

See also *NMR Spectroscopy User Guide*

Related
`lookup` Look up words and lines from a text file (C)
`solvent` Lock solvent (P)

sort Sort real values of a parameter (M)

Syntax `sort(parametername<,sortType>:order,val`

Description Sorts the real values of a parameter. The `sort` macro is not used for parameters holding string values. The default behavior is to the array into values of increasing value. A `sortType` can be given to sort into descending order ('r').

If only unique values are wanted, the 'u' `sortType` can be used. The 'ru' `sortType` given unique values in descending order.

The name of a parameter is the first argument to sort. Two return values hold the results of the sort. The first return value is an array containing the original indexes of the sorted array. The second return value gives the sorted array.

Examples With `par=10,8,6,4,2` the `display('par')` command will show:

```
[1] = 10
[2] = 8
[3] = 6
[4] = 4
[5] = 2
```

The command `sort('par'):$order,$val` will set:

```
$order=5,4,3,2,1
$val =2,4,6,8,10
```

sp Start of plot in directly detected dimension (P)

Description Low-frequency limit of the display or plotted region of the spectrum. `sp` is always stored in Hz, but can be entered in ppm by using the `p` suffix (e.g., `sp=2p` sets the start of plot to 2 ppm).

See also *NMR Spectroscopy User Guide*

Related `sp1` Start of plot in 1st indirectly detected dimension (P)
`sp2` Start of plot in 2nd indirectly detected dimension (P)

sp1 Start of plot in 1st indirectly detected dimension (P)

Description Analogous to the `sp` parameter except that `sp1` applies to the first indirectly detected dimension of a multidimensional data set.

See also *NMR Spectroscopy User Guide*

Related `sp` Start of plot in directly detected dimension (P)
`sp2` Start of plot in 2nd indirectly detected dimension (P)

sp2 Start of plot in 2nd indirectly detected dimension (P)

Description Analogous to the `sp` parameter except that `sp2` applies to the second indirectly detected dimension of a multidimensional data set.

See also *NMR Spectroscopy User Guide*

Related `sp` Start of plot in directly detected dimension (P)

spadd Add current spectrum to add/subtract experiment (C)

Syntax (1) `spadd<(multiplier<,shift>)>`
(2) `spadd('new')`
(3) `spadd('trace',index)`

Description Performs noninteractive spectral addition. The last displayed or selected spectrum is added to the current contents of the add/subtract experiment (`exp5`). A multi-element add/subtract experiment can be created using the `'new'` keyword. Individual spectra in a multi-element add/subtract experiment can be subsequently added to using the `'trace'` keyword followed by an index number of the spectrum.

Arguments `multiplier` is a value to multiply each spectrum being added to the add/subtract experiment (`exp5`). The normal range of `multiplier` would be `+1` to `–1` but the range is actually unlimited. The default is 1.0.

`shift` is the number of data points to shift each spectrum. A positive value shifts the spectrum being added to a higher frequency, or to the left. A negative value shifts the spectrum to a lower frequency, or to the right. The default is 0.

`'new'` is a keyword to create a new spectrum in the add/subtract experiment.

`'trace'` is a keyword to select the spectrum given by the index number argument (`index`) and add it to the add/subtract experiment. The default is to add to the first spectrum in the add/subtract experiment.

`index` is the index number of the spectrum to be used as a target in a multi-element add/subtract experiment.

Examples

```
spadd
spadd(.5,25)
spadd('new')
spadd('trace',2)
```

See also *NMR Spectroscopy User Guide*

Related

`add`	Add current FID to add/subtract experiment (C)
`addi`	Start interactive add/subtract mode (C)
`clradd`	Clear add/subtract experiment (C)
`ds`	Display a spectrum (C)
`jexp`	Join existing experiment (C)
`select`	Select a spectrum without displaying it (C)
`spmin`	Take minimum of two spectra in add/subtract experiment (C)
`spsub`	Subtract current spectrum from add/subtract experiment (C)

spcfrq Display frequencies of rf channels (M)

Description Displays the parameters sfrq, dfrq, dfrq2, and dfrq3 with seven decimal points (to nearest 0.1) to provide the exact frequencies of each rf channel. The number of values displayed depends on numrfch.

Prior to VNMR version 4.3, spcfrq set the frequency of the observe channel. The parameter sfrq now sets the frequency instead of spcfrq.

See also *NMR Spectroscopy User Guide*

Related	
dfrq	Transmitter frequency of first decoupler (P)
dfrq2	Transmitter frequency of second decoupler (P)
dfrq3	Transmitter frequency of third decoupler (P)
numrfch	Number of rf channels (P)
setfrq	Set frequency of rf channels
sfrq	Transmitter frequency of observe nucleus (P)

specdc3d 3D spectral drift correction (P)

Description Sets whether a 3D spectral dc correction occurs. The spectral dc correction is the last operation to be performed upon the data prior to forming linear combinations of the data, using the coefficients in the 3D coefficient file (coef), and then writing the data to disk. If specdc3d does not exist, it is created by the macro par3d.

Values A three-character string selected from 'nnn', 'nny', 'nyn', etc. Each character may take one of two values: n for no spectral dc correction along the relevant dimension, and y for spectral dc correction along the relevant dimension. The first character refers to the f_3 dimension (sw, np, fn), the second character refers to the f_1 dimension (sw1, ni, fn1), and the third character refers to the f_2 dimension (sw2, ni2, fn2). The default is 'nnn'.

See also *NMR Spectroscopy User Guide*

Related	
dc	Calculate spectral drift correction (C)
fiddc3d	3D time-domain drift correction (P)
fn	Fourier number in directly detected dimension (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
fn2	Fourier number in 2nd indirectly detected dimension (P)
ft3d	Perform a 3D Fourier transform (M)
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
np	Number of data points (P)
par3d	Create 3D acquisition, processing, display parameters (C)
ptspec3d	Region-selective 3D processing (P)
sw	Spectral width in directly detected dimension (P)

sw1	Spectral width in 1st indirectly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

spin — Submit a spin setup experiment to acquisition (C)

Description: Regulates sample spinning according to the *parameter* spin, using the acquisition computer. It also sets rf frequency, decoupler status, and temperature.

See also: *NMR Spectroscopy User Guide*

Related:

au	Submit experiment to acquisition and process data (C)
change	Submit a change sample experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
lock	Submit an Autolock experiment to acquisition (C)
sample	Submit change sample, autoshim experiment to acquisition (M)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Sample spin rate (P)
su	Submit a setup experiment to acquisition (M)

spin — Sample spin rate (P)

Description: Selects a regulated spin rate. The rate is changed when a sample is inserted or spin, go, ga, au, or sample are entered.

Values: 0 indicates non-spinning operation.

5 to 39 are spinning rates.

'n' leaves the spin rate at the currently used value and does not wait for regulated spinning before performing acquisition.

See also: *NMR Spectroscopy User Guide*

Related:

au	Submit experiment to acquisition and process data (C)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
sample	Submit change sample, Autoshim experiment to acquisition (M)
sethw	Set values for hardware in acquisition system (C)
spin	Submit a spin setup experiment to acquisition (C)

spincad — Run SpinCAD program (C)

Applicability: SpinCAD Software.

Description Opens the graphical pulse sequence generation utility.

See also *SpinCAD*

Related vnmr2sc VNMR to SpinCAD pulse sequence translator (M)

spingen Compile SpinCAD pulse sequence (M,U)

Applicability SpinCAD Software.

Syntax (From VnmrJ)

```
spingen
spingen(pulsesequence)
spingen<(<option,>pulsesequence<,pulsesequence2>)>
spingen('-psg',pulsesequence)
spingen('-all',pulsesequence)
spingen('-dps',pulsesequence)
```

(From UNIX)

```
spingen pulsesequence < pulsesequence2,,>
spingen <option> pulsesequence < pulsesequence2,, >
spingen -psg pulsesequence
spingen -dps pulsesequence
spingen -all pulsesequence
```

Description Compiles the SpinCAD pulse sequence. The most common usage is the first one (spingen, with no arguments), which compiles the current pulse sequence. Two or more options to SpinCAD compilation are: (1) '-psg' option: compilation for the acquisition go command (2) '-dps' option: compilation for dps usage and (3) '-all' option: include both of the above options and compilation of any Java programs that the pulse sequence may use.

The spingen macro with no arguments does both the go and dps compilations. Individual compilations for go ('-psg' option) and dps ('-dps' option) can also be done (these are rarely used)

In case of SpinCAD sequences and C sequences having the same name, the last compiled sequence will be used for the go command. The isspincad macro can be used to check if the current sequence is SpinCAD or of C type.

Compilation of a SpinCAD sequence generates two files in the user's seqlib directory, pulsesequence.psg and pulsesequence_dps.psg, for every source file pulsesequence. Compiled SpinCAD files are distinct from the C files, in that they have .psg extension in the filenames. Java program files (if used) must reside in ~/vnmrsys/spincad/classes directory. Java programs are compiled and the class files placed in the same ~/vnmrsys/spincad/classes directory. The spingen macro checks for any Java files in /vnmr/spincad/classes directory, if it does not exit in the users's classes directory.

Compilation of a SpinCAD sequence differs from the conventional compilation of C sequences; it involves the expansion of any composites used; transformation of parallel events to a format that Jpsg program can resolve.

Arguments <no option> – compilations for `go` and `dps`
`-psg` – compilation for `go` only
`-dps` – compilation for `dps` only
`-all` – compilations for `go`, `dps`, and also compile any Java programs called from the SpinCAD sequence.

See also *SpinCAD*

Related `spincad` Display SpinCAD interface (M)

`spinll` Set up a slfreq array (M)

Syntax `spinll<('mark')>`

Description Copies a list of frequencies to the `slfreq` parameter in iterative spin simulation and runs `dla`. This macro also clears previous line assignments.

Arguments `'mark'` is a keyword to copy the list of frequencies from the `mark1d.out` file to `slfreq`. The default is to copy the frequencies from the last line listing by `nll` or `dll` to the `slfreq`. Use the cursor and the mark button to place the lines to be assigned in `mark1d.out`. Enter `mark('reset')` to clear the file, and use `nl` to move the cursor to the center of a selected line.

See also *NMR Spectroscopy User Guide*

Related

`dla`	Display line assignments (M)
`dll`	Display listed line frequencies and intensities (C)
`mark`	Determine intensity of the spectrum at a point (C)
`nl`	Position the cursor at the nearest line (C)
`nll`	Find line frequencies and intensities (C)
`slfreq`	Measured line frequencies (P)

`spinner` Open the Spinner Control window (C)

Description Opens the Spinner Control window. This window has the following capabilities:

- Turn the sample spinner off.
- Turn the sample spinner on at a specified speed, in Hz.
- Enable spinner control from within an experiment using the `spin` parameter and the `spin`, `go`, `ga`, or `au` commands. This mode is the default.

- Alternatively, turn off experiment control of the sample spinner and allow only the Spinner Control window (and acqi and sethw) to set the spinning speed. This mode has the advantage that, often times, the spin parameter is different between experiments. Joining a different experiment and entering go can unexpectedly change the spinning speed. This alternate mode prevents this problem. In this mode, when a go, su, ga, or au is entered, the spin parameter is first set to the speed selected in the Spinner Control window and then the spin parameter is set to “Not Used.”
- Select the style of spinner: low-speed style or a high-speed style. If the high-speed style of spinner (used for solids) is selected, the choice of setting the spinning speed or the air flow rate is provided. Setting the air flow rate is useful when setting up the solids spinning apparatus.

  If the spinning speed is controlled only through the Spinner Control window, the action to be taken after a spinner error can be selected:
- Display a warning but continue acquisition.
- Stop acquisition and display a warning.

  If experiment control of spinning speed is selected, these selections are faded because they are inoperative, and the selection of the action to be taken after a spinning speed error is provided by the parameter in.

See also *NMR Spectroscopy User Guide*

Related		
	acqi	Interactive acquisition display process (C)
	au	Submit experiment to acquisition and process data (C)
	change	Submit a change sample experiment to acquisition (M)
	ga	Submit experiment to acquisition and FT the result (C)
	go	Submit experiment to acquisition (C)
	in	Lock and spin interlock (P)
	lock	Submit an Autolock experiment to acquisition (C)
	sample	Submit change sample, autoshim experiment to acquisition (M)
	sethw	Set values for hardware in acquisition system (C)
	shim	Submit an Autoshim experiment to acquisition (C)
	spin	Sample spin rate (P)
	su	Submit a setup experiment to acquisition (M)

spins — Perform spin simulation calculation (C)

Syntax `spins<(options)>`

Description Performs a spin simulation, using the current spin system parameters. Refer to the description of spsm for setting up the parameters. Use dsp to display the spectrum resulting from the simulation. The output file is `spins.list` in the current experiment. This file includes the calculated transitions ordered by frequency.

Line assignments are required for the iteration. These consist of a list of observed frequencies, which is stored in the arrayed parameter slfreq, and the line assignments stored in the array clindex. spinll copies the frequencies from the last line listing by nll or dll into the parameter slfreq. The line listing can be from an observed spectrum or from the results of deconvolution. After spinll, line assignments are most easily made by entering assign. dla displays the assignments. Single assignments can also be made by assign(transition_number,line_number), where transition_number is the index of a transition and line_number is the index of the measured line. Setting the line_number argument to 0 deletes assignments. dla('long') produces an expanded display of assignments.

Be aware that spin simulation line numbers and line list line numbers are *not* the same. Conventional line lists produced by dll number the lines from left to right (low- to high-field). The spin simulation software numbers lines according to a more complicated scheme, and these numbers are rarely if ever in frequency order.

The parameters to be iterated are chosen by setting the string parameter iterate (e.g, iterate='A,B,JAB'). If several parameters have the same value due to symmetry, use iterate='A,B,C,JAB,JAC=JAB'. This string sets the iterated parameter JAC to JAB during the iteration. JAB must be defined as an iterated parameter in the string before it can be used at the right side of the equal sign. Sets of parameters with up to six members may be set up in this way. The member in the set that is used on the right side of the equal sign must always come first in the parameter display (e.g., JAB=JAC would be wrong). A parameter is held constant during iteration if it is not included in the iterate string.

The command initialize_iterate sets iterate to iterate all spins not named X, Y, or Z and the associated coupling constants.

Following an iterative spin simulation, dga displays the new values of the coupling constants and chemical shifts. undospins restores a spin system as it was before the last iterative run. It returns the chemical shifts, coupling constants, and line assignments, making it possible to continue from this state with modified line assignments.

Note that major changes in the starting values of parameters may change the numbering of the energy levels and hence the line numbers. The line assignments would then be incorrect and would have to be reentered.

For a successful iteration, it is often necessary to keep some parameters fixed. For example, it is sometimes useful to alternately iterate couplings and shifts, keeping one group fixed while the other is iterated independently.

Arguments The following variations of spins are available:

- spins('calculate','energy') puts an energy-level table in the output file.
- spins('calculate','transitions') puts a second table of transitions ordered by transition number in the output file.

- spins('display') and dsp are equivalent.
- spins('system','spinsystemname') and spsm('spinsystemname') are equivalent.
- spins('iterate') runs interactively to match experimental and calculated lines.
- spins('iterate','iteration') lists parameters after each iteration in the output file.
- spins('iterate'<,options>) provides for determining the chemical shifts and coupling constants to produce a spectrum that matches a table of observed lines. spins iterates until the rms (root-mean-square) error of the line matching meets a built-in test, unless it first reaches the value given by number_iterations. Iteration also stops if the rms error increases.
- Put multiple list options into the second argument, separated by a blank (e.g., spins('calculate','transitions energy')).

Examples
```
spins
spins('calculate','energy')
spins('iterate')
```

See also *NMR Spectroscopy User Guide*

Related

assign	Assign transitions to experimental lines (M)
clindex	Index of experimental frequency of a transition (P)
dga	Display parameter groups (spin simulation) (C)
dla	Display line assignments (M)
dll	Display listed line frequencies and intensities (C)
dsp	Display calculated spectrum (C)
initialize_iterate	Set iterate to contain relevant parameters (M)
iterate	Parameters to be iterated (P)
niter	Number of iterations (P)
nll	Find line frequencies and intensities (C)
slfreq	Measured line frequencies (P)
spinll	Set up slfreq array (M)
spsm	Enter spin system (M)
undospins	Restore spin system as before last iterative run (M)

split — Split difference between two cursors (M)

Description Repositions the left-hand cursor halfway between its original position and the position of the other cursor. This macro is very useful for finding the center of a powder pattern: place the two cursors on the horns of the pattern and then enter split to give the center.

See also *NMR Spectroscopy User Guide*

Related `delta` Difference of two frequency cursors (P)

spintype Spinner Type ((P)

Description This global parameter determines which spinner hardware is used.

Values `'liquids'` for low speed spinning of 5 and 10 mm liquids samples
`'tach'` for high speed spinning of 5 and 7 mm Jacobsen probes
`'mas'` for high speed spinning using standalone spinner
`'nano'` for spinning of nano probes
`'none'` for no spinner controller is present, e.g. imaging

splmodprepare Used by the dosy macro to prepare data for the program SPLMOD

Syntax `splmodprepare`

Applicability VnmrJ 3.1

Description splmodprepare takes a dosy_in file as created by `dosy` and creates the file dosy_splmod.in in a format suitable for the SPLMOD program (http://s-provencher.com/index.shtml).

See also `splmodread`
`continread`
`continprepare`
`dosy`

splmodread Used by the dosy macro to convert the output of the SPLMOD program into a form suitable for ddif

Syntax `splmodread`

Applicability VnmrJ 3.1

Description `splmodread` takes the file `dosy_splmod.out`, created by SPLMOD (run by the splmodrun shell script from the `dosy` macro) and creates the files diffusion_display.inp and diffusion_spectrum in a suitable format for the `ddif` and `sdp` commands respectively.

See also `splmodread`
`continread`
`continprepare`
`dosy`

spmax Take the maximum of two spectra (C)

Description Takes the maximum of two spectra, considered point-by-point in an absolute-value sense. For example, if the two corresponding values are –2 and +3, the `spmax` spectrum will have +3; if the two values are +2 and –3, the `spmax` spectrum will have –3 at that point.

spmin Take minimum of two spectra in add/subtract experiment (C)

Description Takes the minimum of two spectra, considered point-by-point in an absolute-value sense. For example, if the two corresponding values are `-2` and `+3`, the `spmin` spectrum will have `-2`; if the two values are `+2` and `-3`, the `spmin` spectrum will have `+2` at that point.

The function of `spmin` is to essentially select for common features within two spectra while eliminating features that are not common between them. In particular, if two CP/MAS spectra are obtained at different spin rates, the peaks stay in the same place (and hence the `spmin` spectrum also contains the same peaks), but the sidebands move. If spectrum 1 has baseline where spectrum 2 has sideband, and spectrum 2 has baseline where spectrum 1 has sideband, then the `spmin` spectrum will contain only baseline in these regions, eliminating the spinning sidebands.

See also *NMR Spectroscopy User Guide*

Related
`addi` Start interactive add/subtract mode (C)
`spadd` Add current spectrum to add/subtract experiment (C)
`spsub` Subtract current spectrum from add/subtract experiment (C)

spsm Enter spin system (M)

Syntax `spsm(spin_system)`

Description Enables entry of the spin system for spin simulation and creates and initializes the appropriate parameters to describe the various chemical shifts and coupling constants. Chemical shifts can be entered for the X-nucleus, and the spectrum is calculated if that shift is in the window. Generally, however, it is not necessary to enter the X-nucleus chemical shift, and its value has no effect on the spectrum of the remainder of the spin system.

Arguments `spin_system` is an alphanumeric string of upper-case letters for chemical shift and coupling constant parameters. Chemical shifts are stored in parameters `A` through `Z`, and the coupling constants are stored in the parameters starting with `JAB` and ending with `JYZ`. Different nucleus types are handled by using letters starting with `A` for the first type, X for the second, and M for the third. Once created, these parameters are entered and modified in the usual way (e.g., A=78.5 JAC=5.6). Entry of chemical shifts in ppm is entered by using sfrq (e.g., B=7.5*sfrq).

Examples
```
spsm('AB')
spsm('A3B2')
spsm('AB2CMXY')
```

See also *NMR Spectroscopy User Guide*

Related

sfrq	Transmitter frequency of observe nucleus (P)
spins	Perform spin simulation calculation (C)

spsub Subtract current spectrum from add/subtract experiment (C)

Syntax
(1) spsub<(multiplier<,shift>)>
(2) spsub('new')
(3) spsub('trace',index)

Description
Performs non-interactive spectral subtraction. The last displayed or selected spectrum is subtracted from the current contents of the add/subtract experiment (exp5). A multi-element add/subtract experiment can be created using the 'new' keyword. Individual spectra in a multi-element add/subtract experiment can be subsequently subtracted from using the 'trace' keyword followed by an index number of the spectrum.

Arguments
multiplier is a value to multiply each spectrum being subtracted from the add/subtract experiment (exp5). The normal range of multiplier would be +1 to –1 but is actually unlimited. The default is 1.0.

shift is the number of data points to shift each spectrum. A positive value shifts the spectrum being added to a higher frequency, or to the left. A negative value shifts the spectrum to a lower frequency, or to the right. The default is 0.

'new' is a keyword to create a new spectrum in the add/subtract experiment.

'trace' is a keyword to select the spectrum given by the index number argument (index) and subtract it from the add/subtract experiment. The default is to subtract from the first spectrum in the add/subtract experiment.

index is the index number of the spectrum to be used as a target in a multi-element add/subtract experiment.

Examples
```
spsub
spsub(.5,25)
spsub('new')
spsub('trace',2)
```

See also *NMR Spectroscopy User Guide*

Related

clradd	Clear add/subtract experiment (C)
ds	Display a spectrum (C)
jexp	Join existing experiment (C)
spadd	Add current spectrum to add/subtract experiment (C)
select	Select a spectrum without displaying it (C)
spmin	Take minimum of two spectra in add/subtract experiment (C)
sub	Subtract current FID from add/subtract experiment (C)

sqcosine Set up unshifted cosine-squared window function (M)

Syntax sqcosine<(<t1_inc><,t2_inc>)>

Description Sets up an unshifted cosine-squared window function in 1, 2, or 3 dimensions. The macro checks whether the data is 1D, 2D, and 3D.

Arguments t1_inc is the number of t1 increments. The default is ni.

t2_inc is the number of t2 increments. The default is ni2.

See also *NMR Spectroscopy User Guide*

Related

gaussian	Set up unshifted Gaussian window function (M)
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
pi3ssbsq	Set up pi/3 shifted sinebell-squared window function (M)
pi4ssbsq	Set up pi/4 shifted sinebell-squared window function (M)
sqsinebell	Set up unshifted sinebell-squared window function (M)

sqdir Study queue directory (P)

Description Specifies the full path directory where a study is stored. It is set when a new study is created.

See also *NMR Spectroscopy User Guide, VnmrJ Automation, VnmrJ Walkup, VnmrJ Imaging User Guide*

Related

autodir	Automation directory absolute path (P)
globalauto	Automation directory name (P)
save	Save data (M)
sqname	Study queue parameter template (P)
startq	Start a chained study queue (M)
studyid	Study identification (P)

sqname	Study queue parameter template (P)
xminit	Initialize an imaging study queue (M)

sqend — End a study queue (M)

Description End a study queue. Usually called by other macros, and not used from the command line.

Related

sqfilemenu	Study queue file menu commands (M)

sqexp — Load experiment from protocol (M)

Applicability Imaging

Description Macro to load an experiment from a protocol.

Syntax `sqexp(experiment <, 'save'>)`

The first argument is the name of the experiment, and is required. The second argument is an optional keyword 'save'. If specified, it first saves parameter changes to the current experiment in the study queue before loading the parameters for the new experiment.

Examples `sqexp('epidw')`

`sqexp('spuls','save')`

See also *VnmrJ Imaging User Guide*

Related

apptype	Application type (P)
execpars	Set up the exec parameters (M)

sqfilemenu — Study queue file menu commands (M)

Description A macro to perform commands for the study queue operation. Usually the macro is called from the *study queue file menu* located below the study queue area, and not from the command line.

See also *VnmrJ Imaging User Guide*

Related

cqinit	Initialize liquids study queue (M)
cqreset	Reset study queue parameters (M)
sqend	End a study queue (M)
sqreset	Reset study queue parameters for imaging (M)
xminit	Initialize an imaging study queue (M)

sqLog — Records specific events from a study queue

Syntax `sqLog(event<,arg>) - log automation events`

`sqLog:$path` - return log file path

Applicability VnmrJ 3.1, VnmrJ 3.2

Description The `sqLog` macro records specific events from a study queue. The messages and details of the logging are customizable with the editLog utility.

The sqLog facility will record the following events.

- `SampleStart`
- `SampleEnd`
- `ExpStart`
- `ExpEnd`
- `ExpError`

Each event recorded in the logfile may may be preceded by header information. This may include things like the date, time, user, etc. This header information is also customizable. The `sqLog` macro is very generic. It gets all of its details from a file written be the `editLog` utility. This file has the same name as the macro and is in the `<appdir>/templates/vnmrj/loginfo` directory. For example, the current `sqLog` file is:

```
# Formatting statements for automation log files.
#
1Header Date: %MOC% %DAY% %YR% at %HR%:%MIN%:%SEC%,
User: $operator$, Sample: $samplename$,
1SampleStart Start new sample at location $loc$.
1SampleEnd Finish sample at location $loc$\\####
1ExpStart Experiment $pslabel$ started.
1ExpEnd Experiment $pslabel$ complete.
1ExpError Experiment error:
1ExpPrescan Prescan:
1File $autodir$/logfile
1Ifcondition (auto='y')
```

Lines starting with a hash mark (#) are comments. The first character of each non-comment line is a 1 or 0, indicating enabled or disabled. The rest of the first word, following the 1 or 0, is a keyword that is passed to the `sqLog` macro. The remainder of a line is the template for writing the log file. The template is passed to the `chkname` command for translation.

The File keyword defines where the log file will be saved. If this keyword is disabled, all of the `sqLog` event logging will be disabled. Disabling other keywords only disables that specific event or feature. The Ifcondition keyword allows the logging mechanism to make decisions as to whether to log the event. For example, in the case of sqLog, we only log events during an automation run. Logging will occur only if the Ifcondition is true. A special keyword of "None" for the Ifcondition specifies no special conditions. That is, events are always logged. The sqLog macro is called from appropriate places in the software. It is called with the keyword as the first argument. A second, optional argument can also be passed. It will be appended to the log message generated by the keyword. For example, when called with ExpError, we pass the actual error message as the second argument.

```
geterror:$err
sqLog('ExpError',$err)
```

During an automation run, messages written to 'line3', which puts them into the "acqlog". If sqLog is called with no arguments but one return value, the pathname of the log file, defined by the File keyword, is returned.

As defined above, sqLog saves logging information only for automation runs. By changing the File attribute to your userdir directory, and setting the Ifcondition to None, all study queue activities will be logged, both automation and foreground. The log editor can handle menus of choices. Files in templates/vnmrj/loginfo with the same name as the keyword will be used to make menus of choices to select from within the editLog editor. Files prefixed with the name of the logging macro, for example sqLog will make a File menu specific for editLog('sqLog').

The logging macro, along with the editLog editor are very general and can be used to log other events. As an example, suppose one wants to monitor access to the VnmrJ program. A "loginLog" could be made as follows. Make a copy of the sqLog macro called loginLog. Add a loginLog file describing the events to logged to the <appdir>/templates/vnmrj/loginfo. An example of such a file may be:

```
# Formatting statements for login log files.
#
1Header Date: %MOC% %DAY% %YR% at %HR%:%MIN%:%SEC%,
User: $operator$
1Login Login
1Logout Logout
1File $systemdir$/acqqueue/loginLog
1Ifcondition ((auto='n') and (jviewport=1))
```

The only remaining task is to place calls to the loginLog macro in various other macros. In this case, one might call loginLog('Login'):$res from the bootup macro and loginLog('Logout'):$res from the exit macro. If one wanted to monitor "operator" logins, one could and additional keywords such as operatorlogin and Operatorlogout to the above file and then call loginLog('Operatorlogin'):$res from the operatorlogin macro and call loginLog('Operatorlogout'):$res from the operatorlogout macro.

sqmode Study queue mode (P)

Description A global parameter that specifies the study queue mode. It is used to determine if the study queue acquisition is chained or not.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

startq	Start a chained study queue (M)
xmnext	Find next prescan or next experiment in study queue (M)
xmwexp	Processing macro for end of acquisition in study queue (M)

sqname Study queue parameter template (P)

Description Stores a string in the global tree that determines where a study is stored. It is set from the *Save data setup* dialog in the *Utilities* menu. Dollar signs ($) are used to delimit a string to search for a parameter to be used in the study file name. Percent signs (%) are used to delimit a numeric extension, e.g. %Rn%, or time specifications. Strings from the `sampleinfo` file are not used, since studies are created in foreground, not automation. Text not delimited by dollar signs or percent signs is copied from sqname without any changes.

If `sqname` does not start with a slash mark (/), the study is stored in the path given by `autodir` or `globalauto`; otherwise the name is used as is. A revision number is automatically appended. Values: If `sqname` is a null string, it defaults to %R2%, and the resulting study id is a two-digit revision number. The resulting path and file name must be accessible (with read-write permission) by that user.

Examples

```
sqname='s_%DATE%_%R3%' studyid='s_20040501_001'
sqname='s_$loc$_'       studyid='s_7_01'
sqname='r$vrack$z$vzone$/well$loc$%R0%'
studyid='r1z3/well16'
```

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

autodir	Automation directory absolute path (P)
autoname	Prefix for automation data file (P)
globalauto	Automation directory name (P)
sqdir	Study queue directory (P)
sqname	Study queue parameter template (P)
studyid	Study identification (P)
Svfname	Create path for data storage (C)

sqpars Create study queue parameters for imaging (M)

Applicability Imaging

Description A macro to create study queue parameters for imaging. Usually called by other macros, and not used from the command line.

See also *VnmrJ Imaging User Guide*

Related `fixpar` Correct parameter characteristics in experiment (M)

`sqprotocol` Macro to create protocols (M)

Applicability Imaging

Description A macro to create protocols for imaging applications. Called by the Make protocols dialogs in the Utilities menu.

`sqreset` Reset study queue parameters for imaging (M)

Applicability Imaging

Description Reset study queue parameters for imaging. Usually called by other macros, and not used from the command line.

`sqrt` Return square root of a real number (O)

Description A operator in MAGICAL programming that returns the square root of a real number. A negative argument to `sqrt` is evaluated to 0.0. Operator is not used from the command line.

Examples `a = sqrt(b)`

See also *User Programming*

Related

`asin`	Find arc sine of number (C)
`atan`	Find arc tangent of a number (C)
`cos`	Find cosine value of an angle (C)
`exp`	Find exponential value (C)
`ln`	Find natural logarithm of a number (C)
`tan`	Find tangent value of an angle (C)
`trunc`	Truncates real numbers (O)
`typeof`	Return identifier for argument type (O)

`sqsavestudy` Macro to save study parameters for imaging (M)

Applicability Imaging

Description A macro to save study parameters in the imaging study queue. Usually called by other macros, and not used from the command line.

See also *VnmrJ Imaging User Guide*

Related

acquire	Acquire data (M)
sqend	End a study queue (M)
studypar	Study parameters (P)

sqsinebell Set up unshifted sinebell-squared window function (M)

Syntax `sqsinebell<(<t1_inc><,t2_inc>)>`

Description Sets up an unshifted sinebell-squared window function in 1, 2, or 3 dimensions. The macro checks whether the data is 1D, 2D, and 3D.

Arguments `t1_inc` is the number of t1 increments. The default is ni.

`t2_inc` is the number of t2 increments. The default is ni2.

See also *NMR Spectroscopy User Guide*

Related

gaussian	Set up unshifted Gaussian window function (M
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
pi3ssbsq	Set up pi/3 shifted sinebell-squared window function (M)
pi4ssbsq	Set up pi/4 shifted sinebell-squared window function (M)
sqcosine	Set up unshifted cosine-squared window function (M)

srate Spinning rate for magic angle spinning (P)

Applicability Systems with solids module.

Description Set to the spinning speed for magic angle spinning (MAS). `srate` must be correct for the pulse sequence set up by xpolar1 to run TOSS or dipolar dephasing correctly. If `hsrotor='y'`, the measured spinning speed is reported in `srate` for systems that have rotor synchronization.

Values 0 to 10^7, in Hz.

See also *NMR Spectroscopy User Guide*

Related

hsrotor	Display rotor speed for solids operation (P)
xpolar1	Set up parameters for XPOLAR1 pulse sequence (M)

sread Read converted data into VnmrJ (C)

Syntax `sread(file<,template>)`

Description Reads 32-bit data files into VnmrJ. For Bruker data files in the AMX and AM formats, each file must first be converted using the `convertbru` command before `sread` can read the data in the file into VnmrJ.

Arguments `file` is the name of a file containing data converted using `convertbru`.

`template` is the full path of a parameter template file, but without appending the `.par` extension on the file name. The default is `bruker.par`. If no parameter template is specified and `bruker.par` cannot be found in the user or system `parlib` directory, `sread` aborts with an error message.

Examples `sread('brudata.cv','/vnmr/parlib/bruker')`

See also *NMR Spectroscopy User Guide*

Related `convertbru` Convert Bruker data (M,U)

srof2 Calculate exact rof2 value for Cold Probes (M)

Applicability Systems with Agilent, Inc. Cold Probes

Description Calculates the exact value needed for `rof2` to result in a `lp=0` condition for the given `sw`. Works with either `dsp='r'` and `fsq='y'` or with `dsp='i'`. Not compatible with `qcomp`.

Related
`dsp` Type of DSP for data acquisition (P)
`rof2` Receiver gating time following pulse (P)

ss Steady-state transients (P)

Description Sets the number of complete executions of the pulse sequence not accompanied by data collection prior to the acquisition of the real data (sometimes known as *dummy scans*). If `ss` is positive, `ss` steady-state transients are applied on the first increment only, and if `ss` is negative, `–ss` steady-state transients are applied at the start of each increment.

Values `'n'`, `–32768` to `32767`

See also *NMR Spectroscopy User Guide; User Programming*

ssecho Set up solid-state echo pulse sequence (M)

Applicability Systems with a solids module.

Syntax `ssecho`

Description Converts a standard two-pulse experiment to a ready-to-run solid-state NMR echo (SSECHO) pulse sequence.

See also *NMR Spectroscopy User Guide*

ssecho1 Set up parameters for SSECHO1 pulse sequence (M)

Applicability System with a wideline solids module.

Description Sets up a parameter set for the quadrupole echo pulse sequence SSECHO1.

See also *NMR Spectroscopy User Guide*

ssfilter Full bandwidth of digital filter to yield a filtered FID (P)

Description Specifies the full bandwidth of the digital filter applied to the original FID to yield a filtered FID for solvent subtraction. If `ssfilter` does not exist in the current experiment, enter `addpar('ss')` to add it. The command `addpar('ss')` creates additional time-domain solvent subtraction parameters `ssfilter`, `sslsfrq`, `ssntaps`, and `ssorder`.

Values `'n'`, `10` to `sw/2`, in steps of 0.1 Hz. The default is 100 Hz.

If `ssfilter` is set to a value and `ssorder` is set to some value, the zfs (zero-frequency) option of solvent subtraction is selected.

If `ssfilter` is set to `'n'`, ("Not Used"), both the lfs (low-frequency suppression) and zfs options are turned off.

See also *NMR Spectroscopy User Guide*

Related

`addpar`	Add selected parameters to the current experiment (M)
`ft`	Fourier transform 1D data (C)
`parfidss`	Create parameters for time-domain solvent subtraction (M)
`ssntaps`	Number of coefficients in the digital filter (P)
`sslsfrq`	Center of solvent-subtracted region of spectrum (P)
`ssorder`	Order of polynomial to fit digitally filtered FID (P)
`sw`	Spectral width in directly detected dimension (P)
`wft`	Weight and Fourier transform 1D data (C)

sslsfrq Center of solvent-suppressed region of spectrum (P)

Description Specifies the location of the center of the solvent-suppressed region of the spectrum. If `sslsfrq` does not exist in the current experiment, enter `addpar('ss')` to add it. `addpar('ss')` also creates time-domain solvent subtraction parameters `ssfilter`, `ssntaps`, and `ssorder`.

Values `'n'` (or `0`) specifies solvent suppresses a region centered about the transmitter frequency. This is the default

Non-zero value shifts the solvent-suppressed region by sslsfrq Hz. Multiple regions may be suppressed by arraying the value of sslsfrq. Up to 4 values are allowed.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
parfidss	Create parameters for time-domain solvent subtraction (M)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
ssntaps	Number of coefficients in the digital filter (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)

ssntaps Number of coefficients in digital filter (P)

Description Specifies the number of taps (coefficients) to be used in the digital filter for solvent subtraction. If ssntaps does not exist in the current experiment, enter addpar('ss') to add it. addpar('ss') also creates time-domain solvent subtraction parameters ssfilter, sslsfrq, and ssorder.

Values Integer from 1 to np/4. The default is 121. An odd number is usually best.

The more taps in a filter, the flatter the passband response and the steeper the transition from passband to stopband, giving a more rectangular filter.

For the lfs (low-frequency suppression) option, the default is suitable.

For the zfs (zero-frequency suppression) option, a value between 3 and 21 usually works better.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
ft	Fourier transform 1D data (C)
ni	Number of increments in 1st indirectly detected dimension (P)
np	Number of points (P)
parfidss	Create parameters for time-domain solvent subtraction (M)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
sslsfrq	Center of solvent-suppressed region of spectrum (P)
ssorder	Order of polynomial to fit digitally filtered FID (P)
wft	Weight and Fourier transform 1D data (C)

ssorder Order of polynomial to fit digitally filtered FID (P)

Description Specifies the order of the polynomial to fit the digitally filtered FID if the zfs (zero-frequency suppression) option is selected for solvent subtraction. ssorder is not used if the lfs (low-frequency suppression) option is selected. If ssorder does not exist in the current experiment, enter addpar('ss') to add it. addpar('ss') also creates time-domain solvent subtraction parameters ssfilter, sslsfrq, and ssntaps.

The solvent subtraction option (zfs or lfs) is selected as follows:

- If ssorder and ssfilter are both set to values, zfs is selected.
- If ssorder='n' and ssfilter is set to a value, lfs is selected.
- If ssorder='n' and ssfilter='n', zfs and lfs are both turned off.

Values 'n', integer from 1 to 20. The default is 'n'.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
parfidss	Create parameters for time-domain solvent subtraction (M)
ssfilter	Full bandwidth of digital filter to yield a filtered FID (P)
sslsfrq	Center of solvent-suppressed region of spectrum (P)
ssntaps	Number of coefficients in the digital filter (P)
wft	Weight and Fourier transform 1D data (C)

stack — Stacking mode for processing and plotting arrayed spectra (M)

Syntax `stack(mode)`

Description When processing and plotting arrayed 1D spectra, VnmrJ automatically determines if the *stacking mode* is horizontal, vertical or diagonal from the number of traces and the number of lines in the spectrum. If you do not want this automatic function (or it makes an undesirable decision), you can override it by placing the stack macro in the experiment startup macro or by calling stack before processing (or reprocessing) a spectrum. The macro autostack switches back to automatic determination of the stack mode by destroying the parameter stackmode.

Arguments mode is one of the stacking modes 'horizontal', 'vertical', or 'diagonal'.

See also *NMR Spectroscopy User Guide*

Related

autostack	Automatic stacking for processing and plotting arrays (M)
procarray	Process arrayed 1D spectra (M)
plarray	Plot arrayed 1D spectra (M)
stackmode	Stacking control for processing (P)

`stackmode` Stacking control for processing arrayed 1D spectra (P)

Description: Controls whether stacking for processing arrayed 1D spectra is automatic or nonautomatic. The *automatic stacking mode* can be overridden by creating and setting `stackmode` in the startup macro or before calling `procplot` or `procarray`. The `autostack` macro switches back to automatic determination of the stack mode by destroying this parameter.

Values: `'horizontal'`, `'vertical'`, or `'diagonal'`.

See also: *NMR Spectroscopy User Guide*

Related:

`autostack`	Automatic stacking for processing and plotting arrays (M)
`procarray`	Process arrayed 1D spectra (M)
`procplot`	Automatically process FIDs (M)
`stack`	Fix stacking mode for processing and plotting arrayed spectra (M)

`startq` Start a chained study queue (M)

Description: Start a chained acquisition for a study queue.

Related:

`sqmode`	Study queue mode (P)
`xmnext`	Find next prescan or next experiment in study queue (M)

`status` Display status of sample changer (C,U)

Applicability: Systems with an automatic sample changer.

Syntax: `status<(directory<,config_file>)>`
(From UNIX) `status directory <config_file>`

Description: Displays a status window with a summary of all experiments and a scrollable list of individual experiments. Individual experiments are selected by clicking anywhere on the experiment of interest. `status` updates as the state of an automation run changes. If an experiment finishes or a new experiment is added, the `status` display is updated.

Arguments: `directory` is the path to the directory where the done queue (`doneQ`) is stored. In the UNIX shell, a directory path is required. In VnmrJ, a directory path is optional. The default is the automation mode directory.

`config_file` is the name of a user-supplied file that customizes status for local use. Refer to the manual *User Programming* for details.

Examples	(From VnmrJ) `status` (From VnmrJ) `status('/home/vnmr1/AutoRun_621')` (From UNIX) `status /home/vnmr1/AutoRun_621 mystatus`	
See also	*VnmrJ Walkup; User Programming*	
Related	autodir	Automation directory absolute path (P)
	autoname	Prefix for automation data file (P)
	enter	Enter sample information for automation run (C,U)

std1d Apptype macro for Standard 1D experiments (M)

Applicability	Liquids	
Description	Perform the actions for Standard 1D protocols to set up, process, and plot experiments.	
See also	*NMR Spectroscopy User Guide, VnmrJ Walkup*	
Related	apptype	Application type (P)
	execpars	Set up the exec parameters (M)

stdshm Interactively create a method string for autoshimming (M)

Syntax `stdshm`

Description

Creates a `method` string to be used in adjusting the spinning controls z1, z2, z3, and z4 when a sample is changed. If non-spin controls also need adjusting, further shimming operations are required.

The `method` string is constructed in answer to questions about the sample length, the time available for shimming, and the solvent T_1 or, in FID shimming, the T_1 of the sample. In asking about sample height, `stdshm` assumes that z3 and z4 need adjusting only with short samples; therefore, select "sample height will vary" if z3 and z4 shimming is definitely wanted.

Try lock shimming first to see if it produces a satisfactory result. Lock shimming requires a much shorter shimming time than FID shimming and usually adjusts z1 and z2 just as well. If lock shimming is unsatisfactory, try FID shimming. Again, when z3 and z4 adjustment is required, lock shimming is faster, but FID shimming is more effective. `stdshm` displays the estimated shimming time, permitting revision when the time is too long.

To shim after running `stdshm`, enter `method='std'` (for lock shimming) or `method='fidstd'` (for FID shimming). Then enter `shim` or set the `wshim` parameter to shim before the start of acquisition.

Note that the command `newshm` is much like `stdshm` but that `newshm` provides more flexibility in making `method` strings

See also *NMR Spectroscopy User Guide*

Related
dshim	Display a shim method string (M)
method	Autoshim method (P)
newshm	Interactively create a shim method with options (M)
shim	Submit an Autoshim experiment to acquisition (C)
wshim	Conditions when shimming is performed (P)

sth Minimum intensity threshold (P)

Description Intensity threshold above which transitions are printed and included in the simulated spectrum. Transitions whose intensity falls below this threshold are omitted from the simulation.

Values 0 to 1.00. A typical value is 0.05.

See also *NMR Spectroscopy User Guide*

Related
spins	Perform spin simulation calculation (C)
spsm	Enter spin system (M)
th	Threshold (P)

string Create a string variable (C)

Syntax `string(variable)`

Description Creates a string variable without a value.

Arguments `variable` is the string variable to be created.

Examples `string('strvar1')`

See also *User Programming*

string2array Formats a String Variable into an Array

Description Converts a string variable into an array.

Syntax `string2array('parameter'):$array`

Examples `string2array():$S1`

Related array2string, array2csv, array2strsv, srtsv2array

strstr — Sets ret to the starting position of the first occurrence of string2 in string1

Syntax
`strstr(string1,string2):ret,s1,s2` - find position of one string in another

`strstr(string1,string2,'last'):ret,s1,s2` - find last position of one string in another

Applicability VnmrJ 3.1

Description This command sets ret to the starting position of the first occurrence of string2 in string1. The first character position is 1. This command returns 1 if string2 is empty. It returns 0 if string2 does not occur in string 1. Two additional values can be returned. These correspond to the segments of string1 which precede and follow string2, respectively. If string2 does not exist is string1, the two returned segments are both set to a null string. This command can be used in a variety of ways. The examples below demonstrate determination of the file extension. Also, it can reproduce the UNIX basename and dirname commands.

Arguments

Examples
`n1='/export/home/vnmr1/vnmrsys/data/studies/s_2002-04-10_001/data/sems_01.fid'`

`n2='/s_2002-04-10_001/data/'`

strstr(n1,n2):$ret,$s1,$s2

`$ret` will be set to 40

`$s1` will be set to parent of the studies directory

`'/export/home/vnmr1/vnmrsys/data/studies'`

`$s2` will be set to the file name of a saved data set.

`'sems_01.fid'`

The combined `$s1+n2+$s2` will be equal to `n1`.

If a third optional 'last' argument is given, then strstr will find the last occurrence of string2 in string1. The return arguments are the same. This might be used to find the extension of a file name. For example,

n1='/export/home/vnmr1/vnmrsys/data/old.studies/s_2002-04-10_001/data/sems_01.fid'

strstr(n1,'.'):$ret,$s1,$s2

`$ret` will be set to 36

`$s1` will be set to

`'/export/home/vnmr1/vnmrsys/data/old'`

`$s2` will be set to

`'studies/s_2002-04-10_001/data/sems_01.fid'`

However,

strstr(n1,'.','last'):$ret,$s1,$s2

`$ret` will be set to 74

`$s1` will be set to

'/export/home/vnmr1/vnmrsys/data/old.studies/s_2002-04-10_001/data/sems_01'

`$s2` will be set to the extension.

`'fid'`

To find the directory and basename of a file path, the following can be used.

strstr(n1,'/','last'):$ret,$s1,$s2

This will set `$s1` to the directory

`'/export/home/vnmr1/vnmrsys/data/old.studies/s_2002-04-10_001/data'`

and `$s2` will be the basename

`'sems_01.fid'`

strsv2array Formats a String Separated Variable into an Array

Description Converts a string separated variable into an array.

Syntax `strsv2array('parameter'):$array`

Examples `strsv2array():$R1`

Related `array2string`, `array2csv`, `array2stringview`, `string2array`

strtext Starting point for LP data extension in np dimension (P)

Description Specifies inclusively the complex time-domain data point at which LP (linear prediction) data extension (alteration) is to begin in the `np` dimension. Enter `addpar('lp')` to create `strtext` and other `np` dimension LP parameters in the current experiment.

Values `1` to `np/2`

See also *NMR Spectroscopy User Guide*

Related
`addpar`	Add selected parameters to the current experiment (M)
`lpalg`	LP algorithm in `np` dimension (P)
`np`	Number of data points (P)
`strtlp`	Starting point for LP calculation in `np` dimension (P)

strtext1 Starting point for LP data extension in ni dimension (P)

Description Specifies inclusively the complex time-domain data point at which LP (linear prediction) data extension (alteration) is to begin in the `ni` dimension. Enter addpar('lp',1) to create strtext1 and other ni dimension LP parameters in the current experiment.

Values 1 to ni/2

See also *NMR Spectroscopy User Guide*

Related		
	addpar	Add selected parameters to the current experiment (M)
	lpalg1	LP algorithm in ni dimension (P)
	ni	Number of increments in 1st indirectly detected dimension (P)
	strtlp1	Starting point for LP calculation in ni dimension (P)

strtext2 Starting point for LP data extension in ni2 dimension (P)

Description Specifies inclusively the complex time-domain data point at which LP (linear prediction) data extension (alteration) is to begin in the ni2 dimension. Enter addpar('lp',2) to create strtext2 and other ni2 dimension LP parameters in the current experiment.

Values 1 to ni2/2

See also *NMR Spectroscopy User Guide*

Related		
	addpar	Add selected parameters to the current experiment (M)
	lpalg2	LP algorithm in ni2 dimension (P)
	ni2	Number of increments in 2nd indirectly detected dimension (P)
	strtlp2	Starting point for LP calculation in ni2 dimension (P)

strtlp Starting point for LP calculation in np dimension (P)

Description Specifies the first complex, time-domain data point to be used in calculating the complex linear prediction (LP) coefficients in the np dimension. If lpopt='b', the strtlp-th complex time-domain data point and the ensuing (2*lpfilt–1) data points are used in this calculation. If lpopt='f', the strtlp-th complex time-domain data point and the preceding (2*lpfilt-1) data points are used in this calculation. Enter addpar('lp') to create strtlp and other np dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related		
	addpar	Add selected parameters to the current experiment (M)
	lpalg	LP algorithm in np dimension (P)
	lpfilt	LP coefficients to calculate in np dimension (P)
	lpnupts	LP number of data points in np dimension (P)
	lpopt	LP algorithm data extension in np dimension (P)
	strtext	Starting point for LP data extension in np dimension (P)

strtlp1 Starting point for LP calculation in ni dimension (P)

Description Specifies the first complex, time-domain data point to be used in calculating the complex linear prediction (LP) coefficients in the ni dimension. It functions analogously to strlp. Enter addpar('lp',1) to create strtlp1 and other ni dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpalg1	LP algorithm in ni dimension (P)
lpfilt1	LP coefficients to calculate in ni dimension (P)
lpnupts1	LP number of data points in ni dimension (P)
lpopt1	LP algorithm data extension in ni dimension (P)
strtext1	Starting point for LP data extension in ni dimension (P)

strtlp2 Starting point for LP calculation in ni2 dimension (P)

Description Specifies the first complex, time-domain data point to be used in calculating complex linear prediction (LP) coefficients in the ni2 dimension. strtlp2 functions analogously to strlp. Enter addpar('lp',2) to create strtlp2 and other ni2 dimension LP parameters in the current experiment.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
lpalg2	LP algorithm in ni2 dimension (P)
lpfilt2	LP coefficients to calculate in ni2 dimension (P)
lpnupts2	LP number of data points in ni2 dimension (P)
lpopt2	LP algorithm data extension in ni2 dimension (P)
strtext2	Starting point for LP data extension in ni2 dimension (P)

studyid Study identification (P)

Applicability Liquids

Description Specifies the relative directory where a study is stored. In Walkup, it is relative to autodir. In imaging, it is relative to globalauto; It is set when a new study is created.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related

autodir	Automation directory absolute path (P)
globalauto	Automation directory name (P)
sqdir	Study queue directory (P)
sqname	Study queue parameter template (P)

studypar Study parameters (P)

Applicability Liquids, Imaging

Description A global parameter that contains the list of parameters saved with a study. If the parameter does not exist, it is created by `cqsavestudy` for liquids or `sqsavestudy` for imaging when a study is saved.

See also *NMR Spectroscopy User Guide, VnmrJ Automation, VnmrJ Walkup, VnmrJ Imaging User Guide*

Related

`cqsavestudy`	Macro to save study queue parameters (M)
`sqsavestudy`	Macro to save study parameters for imaging (M)

studystatus Study status (P)

Applicability *VnmrJ Walkup*

Description The status of a study for a sample. The status is set from the status of the experiments within the study by the macro `cqsavestudy`.

See also *VnmrJ Walkup*

Related

`cqsavestudy`	Macro to save study queue parameters (M)
`studytime`	Study time (P)

studytime Study time (P)

Applicability *Walkup*

Description The total time it takes to run a study. It is set by the `xmtime` macro when a study is created.

See also *VnmrJ Walkup*

Related

`xmsubmit`	Submit sample(s) to the study queue (M)
`xmtime`	Update the study queue time (M)

su Submit a setup experiment to acquisition (M)

Description Sets up the system hardware to match the current parameters but does not initiate data acquisition. Typical uses of `su` are to change the system frequency in preparation for probe tuning, to change the sample temperature in advance of beginning an experiment (or after a variable temperature experiment is run), and to turn the decoupler on or off. If `load='y'`, `su` can be used to set shim values. `su` also sets lock parameters (`lockpower`, `lockgain`, `lockphase`) and the field offset parameter (`z0`).

su does *not* delete any existing data in the current experiment (only go, ga, and au do that). Everything that su does is also done by go, ga, and au.

Shim DAC values are automatically loaded when the acquisition system boots up; if the acquisition system has been recently rebooted, su must be entered before acqi or qtune can be run.

See also *NMR Spectroscopy User Guide*

Related

acqi	Interactive acquisition display process (C)
au	Submit experiment to acquisition and process data (C)
change	Submit a change sample experiment to acquisition (M)
ga	Submit experiment to acquisition and FT the result (C)
go	Submit experiment to acquisition (C)
load	Load status of displayed shims (P)
lock	Submit an Autolock experiment to acquisition (C)
lockgain	Lock gain (P)
lockphase	Lock phase (P)
lockpower	Lock power (P)
qtune	Tune probe using swept-tune graphical tool (C)
sample	Submit change sample, autoshim experiment to acquisition (M)
shim	Submit an Autoshim experiment to acquisition (C)
spin	Submit a spin setup experiment to acquisition (C)
z0	Z0 field position (P)

sub — Subtract current FID from add/subtract experiment (C)

Syntax
(1) sub<(multiplier<,'new'>)>
(2) sub('new')
(3) sub('trace',index)

Description
Subtracts the last displayed or selected FID from the current contents of the add/subtract experiment (exp5). lsfid and phfid can be used to shift or phase rotate the selected FID before it is subtracted from the data in add/subtract experiment. A multi-FID add/subtract experiment can be created by using the 'new' keyword. Individual FIDs in a multi-FID add/subtract experiment can subsequently be subtracted by using the 'trace' keyword followed by the index number of the FID.

Arguments
multiplier is a value that the FID is to be multiplied by before being subtracted from the add/subtract experiment (exp5). The default is 1.0.

'new' is a keyword to create a new FID element in an add/subtract experiment.

'trace' is a keyword to use the next argument (index) as the number of the FID to subtract from in an add/subtract experiment. The default is to subtract from the first FID in a multi-FID add/subtract experiment.

index is the index number of the FID to be used as a target in a multi-FID add/subtract experiment.

Examples
```
sub
sub(0.75)
sub('new')
sub('trace',2)
```

See also *NMR Spectroscopy User Guide*

Related

add	Add current FID to add/subtract experiment (C)
clradd	Clear add/subtract experiment (C)
lsfid	Number of complex points to left-shift ni interferogram (P)
phfid	Zero-order phasing constant for np FID (P)
select	Select a spectrum without displaying it (C)
spsub	Subtract current spectra from add/subtract experiment (P)

substr Select a substring from a string (C)

Applicability VnmrJ

Syntax
```
substr('string',word_number):$n1<,$n2<,$n3>>
substr('string',index,length<,'new_string'>):
n1<,$n2<,$n3>>
substr('string',word_number,'delimiter',
'delimiter_char'):
n1<,$n2<,$n3>>
```

Description Picks a substring or word out of a string, replace, or delete a set of characters from a string and returns the result to the string variable $n1. The position of the first character of the word and the number of characters of the word are returned to $n2 and $n3 if these string variables are supplied.

Arguments 'string' string or a string variable.

word_number is the number of the word to select. Words are counted sequential beginning with the first word of the string as 1.

index is the number of characters counted from the first character of the string or a string variable containing this number.

length is the number of characters in the substring.

new_string is string or a string variable to replace the contents of string at the position specified by index and length and pass the resulting string to the return string variable.

'delimiter' is a keyword that requires the 'delimiter_char' argument to specify that the argument that follows specifies the delimiter(s).

'delimiter_char' is a string of characters to use as delimiters to separate words.

Default delimiters are space and tab " \t".

`$n1` is the return string variable containing the searched for text.

`$n2` is the return variable containing the position of the first character of the word in the string.

`$num` is the return string variable containing the number of characters in the word specified by `word_number` and contained within the delimiters.

Examples Search examples:

`substr('There are 10 samples to be run',4):n1`
string `n1='samples'`

`substr('There are 10 samples to be run',4):n1,$f,$num`
sets strings `n1='samples' $f=14` and `$num=7`

`substr('abcdefg',2,3):n1`
string `n1='bcd'`

`substr('This is;a phrase',2):n1`
string `n1='is;a'`

`substr('This is;a phrase',2,`
`'delimiter',' ;\t'):n1,$f,$num`
sets strings `n1='is' $f=6` and `$num=2`

Text substitution examples:

Explicit text substitution and passing the result to the return string variable.
`substr('abcdefg',2,3,'1234'):n1`
string `n1='a1234efg'`

Text substitution in a string variable using results held in return string variable from a previous search. Start with the following text held in a string variable:
`n1='There are 10 samples to be run'`
`substr(n1,4):n2,$f,$num`
sets strings `n2=samples`, `$f=14`, and `$num=7`

`substr(n1,$f,$num,'experiments'):n3`
Counts 14 characters (`$f=14`) from the beginning of `n1`, substitutes the word `experiments` for the 7 character (`$num=7`) word in `n1`, and passes the new string to the return string variable setting
`n3='There are 10 experiments to be run'`

See also *User Programming*

Related		
	`length`	Determine length of a string (C)
	`string`	Create a string variable (C)

suselfrq Select peak, continue selective excitation experiment (M)

Syntax `suselfrq`

Description Sets up selective frequency pulse, power, and shape and continue with the selective excitation experiment. Used by `Noesy1d`, and `TOCSY1D`.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related	Noesy1d	Change parameters for NOESY1D experiment (M)
	setselinv	Set up selective inversion (M)
	setselfrqc	Select selective frequency and width (M)
	TOCSY1D	Change parameters for TOCSY1D experiment (M)

svdat Save data (C)

Syntax `svdat(file<,'f'|'m'|'i'|'b'>)`

Description Outputs current data from the current experiment to a file. Integer data is scaled when it is written.

Arguments `file` is the name of the data file. The file is created in the current directory VnmrJ is in unless a full directory path is given. If a file of the same name already exists, the user will queried to overwrite the file. If a fully qualified filename is not given, the file will be created in VnmrJ's current directory.

`'f'|'m'|'i'|'b'` defines how the data is to be written out: `'f'` is 32-bit floating point, `'m'` or `'i'` is 16-bit integer scaled to 12 bits, and `'b'` is 8-bit byte integer. The default is `'f'`.

Floating point data is not scaled when written.

Integer data is scaled when written. A data value `x` is scaled as `ax+b` where:

```
a = (vs*grays1*numgray)/64.0
b = numgray*(0.5-(grays1*grayctr/64.0))
```

where `numgray` (see below) has a default of 4096 for `'m'` and`'i'`formats and a default of 256 for the `'b'` format, `graysl` has a default of 1, and `grayctr` has a default of 32.0.

To scale 16-bit integer data other than 12-bits, the global parameter `numgray` can be created using `create(numgray,real,global)` and set to the value 2^n, where n is the number of bits desired. For example, to scale to 15-bits, set `numgray=32768`.

The display parameters `graysl` and `grayctr` are used to save data files for ImageBrowser.

Examples `svdat(rathead,'b')`

See also *VnmrJ Imaging NMR*

Related	create	Create new parameter in parameter tree (C)
	grayctr	Gray level window adjustment (P)
	graysl	Gray level slope (contrast) adjustment (P)

svf Save FIDs in current experiment (M)

Syntax `svf<(file<,'nolog'><,'arch'><,'force'><,'nodb'>)>`

Description Saves parameters, text, and FID data in the current experiment to a file. No data is removed from the current experiment; `svf` merely saves a copy of the data in a different file. You can enter `rt` to retrieve the complete data set, or enter `rtp` to retrieve parameters only.

Arguments `file` is the name of the file, with the suffix `.fid` added, to be created to save the data. The default is the system prompts for a file name. You are warned if you attempt to overwrite a file that already exists. In fact, if data has been acquired with the `file` parameter set, the data does not need to be saved. It is already stored in a named file.

`'nolog'` is a keyword to not save the log file with the data. The default is to save the log file.

`'arch'` is a keyword to assume that the data goes to a database and appends to the (or creates a) `doneQ` file with information that can be used by the command `status`.

If `force` is given, you are not warned and the older parameter set is removed.

`nodb` is a keyword to prevent `svp` from adding information to a database. This prevention is useful if temporary parameter files are saved that will soon be removed.

Examples
```
svf
svf('/home/vnmr1/mydatafile')
```

See also *NMR Spectroscopy User Guide*

Related

`file`	File name (P)
`rt`	Retrieve FID (M)
`rtp`	Retrieve parameters (M)
`status`	Display status of all experiments (C)

svfdf Save FID data in FDF format (M)

Syntax `svfdf(directory)`

Description Saves raw data from the FID file of the current experiment as an FDF (Flexible Data Format) file. Data is saved in multiple files, with one trace per file. The files are named `fid0001.fdf`, `fid0002.fdf`, etc. The `procpar` file from the current experiment is also saved in the same directory.

The FDF file format is described in the manual *User Programming*. Note that the data is complex (FDF type="`complex`"), and the FDF ordinate = {"`intensity`", "`intensity`"}, indicating that each point consists of a pair of intensities. The FDF headers also contain the following special fields:

- `nfile` gives the sequential number of this file in the series.
- `ct` is the value of the `ct` parameter. The data should be divided by `ct` to give the average signal intensity for one scan.
- `scale` gives the power of two scaling factor for the data. The data should be multiplied by $2^{\mathtt{scale}}$ to give the true values.

Arguments directory_name is the directory in which to store the files. The extension .dat is appended to the given name.

Examples svfdf(curexp+'/raw')

See also *User Programming*

Related	ct	Completed transients (P)

svfdir Directory for non-study data (P)

Description Specifies the directory where data is saved when not using a study in VnmrJ.

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related	fidsave	Save data (M)
	save	Save data (M)
	svfname	Filename parameter template for non-study data ((P)

svfj Save FID in JCAMP-DX format (M)

Syntax svfj<(filename<,opt>)>

Applicability VnmrJ 3.1

Description "svfj" saves the current 1D FID in JCAMP-DX format. "svfj" creates temporary files "/vnmr/tmp/jdxfid.real" and "/vnmr/tmp/jdxfid.imag"; it calls two external C programs "listparam" and "jdxfid". Only a single FID (the current trace in the case of an arrayed experiment) is saved. "svfj" does not work with nf>1.

Arguments "filename" is the name of the target file. If no filename is supplied, the software checks if "file" differs from "exp" (i.e., you have used "rt" to load the dataset from disk). If "file<>'exp'" and if the FID file is writable, then the JCAMP-DX data are saved as "{file}/dx_name.dx", where "{file}" has ".fid" added, if necessary, and "dx_name.dx" is the "basename" part of "file" (minus the ".fid" extension). If "file='exp'" or if the FID directory is not writable, the user is prompted for the filename.

The resulting ASCII file by default has a ".dx" extension, unless the specified filename has an alternative extension. (the default can be altered in the header of the macro). NOTE: ALL files created - whether they are FIDs, spectra or line lists - have the same (default) extension (the distinction between FIDs and spectra is made within the JCAMP-DX format); it is up to the user to avoid overwriting files! Where multiple arguments are specified / allowed, "filename"MUST be the first argument.

"opt" is an optional argument that permits controlling the compression in the data part of the JCAMP-DX file. Possible options and their effects are:

Table 3. Option

	tbl	fix	pac	sqz	dup	dif	difdup	Effect	Description
X,Y	X	-	-	-	-	-	-	X,Y list	
FIX	-	X	-	-	-	-	-	X(Y..Y)	readable
PAC	-	-	X	-	-	-	-	X(Y..Y)	packed
SQZ	-	-	-	X	X	X	X	X(Y..Y)	squeezed
DIF	-	-	-	-	-	X	X	SQZ	differences
DUP	-	-	-	-	X	-	X	SQZ	DUP mode

The default mode / option is "'difdup'" which usually gives the best compression. For human-readable data use the "'fix'" option, for programs expecting output in X Y format (one ordinate value per line) use the "'tbl'" option. All format options comply with the JCAMP-DX format and should be usable.

Examples `svfj`

svfj('myfid')

svfj('myfid.jdx','dup')

select(3) svfj('myfid_3')

Related `listparam` list parameters in simple format (UNIX)
`writetrace` write ascii file from phasefile (f1 or f2) trace (M)

Svfname — Create path for data storage (C)

Applicability Automation

Syntax `Svfname:$path`

`Svfname(name_template):$path`

`Svfname(name_template,suffix):$path`

`Svfname(name_template,suffix, excluded_suffix'):$path`

`Svfname(name_template,suffix<,'excluded_suffix',`
`<'keepspaces'|'replacespaces'>):$path`

Description Determines the name used to store data. This command provides the functionality of the autoname parameter without being in automation mode.

`Svfname` default naming command with alternate suffixes is `svfname` and the default directory is `svfdir`. `Svfname` does not read a sample info file. A suffix is specified as the second argument. Use a suffix of '' to access ordinary files and directories. Arguments used with `Svfname` are constructed the same way arguments are constructed for `autoname`.

The name is prefixed with using the value of the parameter autodir or userdir+'/data/' if name_template is a relative path.

The default suffix is .fid.

Arguments svfname is default naming parameter.

svfdir is default directory parameter.

name_template (no quotes) is string that contains keywords separated by substitution specifiers to represent the data storage path. Substitution specifiers in this template are either a percent sign (%) or a dollar sign ($). The keywords are obtained using % substitution specifiers or VNMR parameters using $ substitution specifiers.

Percent sign (%) substitution specifier is used to scan for the text specified by keyword between the first percent sign in the template string and the next percent sign. The text specified by the keyword between the % substitution specifiers is passed to $path.

The following percent substitutions (% keywords) for time and date are obtained from the system clock, not from the sample info file:

Keyword	Format	Description
%DATE%	YYYYMMDD	4-digit year, 2-digit month, 2-digit day
%TIME%	HHMMSS	2-digit hour, 2-digit minute, 2-digit second
%YR%	YYYY	4-digit year
%YR2%	YY	2-digit year
%MO%	MM	2-digit month
%DAY%	DD	2-digit day
%HR%	HH	2-digit hour
%MIN%	MM	2-digit month
%SEC%	SS	2-digit second

Dollar sign ($) substitution specifier is used with the Svfname command to interpreted a VNMR parameter and substitute the value of this parameter a suffix.

Numeric parameters are truncated and represented as a string with the form: <optional string>parameter value<optional string>. The name_template, pw=pwusec, with vnmr parameter pw having a value of 12.3 produces pw=12usec01 which is appended to .fid (or .img) and passed to $path.

String parameters cannot not contain any of the following characters: ' ', '!', '"', '$', '&', '\', '', '(', ')', '*', ';', '<', '>', '?', '\\', '[', ']', '^', '`', '{', '}', '|', ',', '\0'

A comma separated excluded suffix list appends a string based on the suffixes and excluded suffixes to the path. Using the keyword 'replacespaces' uses underscores (_) in place of spaces ' ' in the resulting path name. The keyword 'keepspaces' retains spaces in the resulting path name.

'keepspaces'|'replacespaces' is an optional argument (includes quotes) that uses either of the following keywords: replacespaces or keepspaces. The argument is accepted if the third argument is a list of suffixes. The action is the same as described for the third argument

Version number is specified by %Rn% where n is an integer from 0 to 9 (default 2), as follows:

n=	Description
0	no revision digits are appended (all names must be uniquely constructed without these revision digits).
1 to **9**	revision number is padded with leading zeroes to form an n-digit number. If more places are needed than specified, more zeroes are used.
>9 (more than one digit)	Rnn is still used as a search string in the sampleinfo file. %Rn% must be specified at the end of the `name_template` string. The revision digits are always appended except if %R0% is used.
no %Rn%	default of %R2% is used

See also *NMR Spectroscopy User Guide, VnmrJ Automation User Guide, VnmrJ Walkup*

Related
`autoname` Determines path for data storage during an automation run (C)
`autoname` Temple determining the path where is data stored (P)
`sqname` Study queue parameter template (P)
`svfname` Specifies the filename template (P)

`svfname` Filename parameter template for non-study data (P)

Description Specifies the filename template where data is saved when not using a study in VnmrJ. The template is constructed using the same keywords and delimiter, dollar sign ($) and percent sign (%), as `autoname`.

Examples If `svfdir=userdir+'/data'`, the result from fidsave is:
```
svfname='$pslabel$_$tn$_' ->
userdir+'/data/Proton_H1_01.fid'
svfname='%DATE%/t%TIME%%R0%' ->
userdir+'/data/20040501/t113005.fid'
```

See also *NMR Spectroscopy User Guide, VnmrJ Walkup*

Related
`fidsave` Save data (M)
`Svfname` Create path for data storage (C)
`sqname` Study queue parameter template (P)
`save` Save data (M)
`svfname` Filename parameter template for non-study data ((P)

`svimg` Generate and Save images as FDF files. (macro)

Syntax `svimg('directory_name'[,'outfmt'])`

Applicability VnmrJ 3.1

Description The "`svimg`" command generates images from the current experiment and saves them into the specified directory as Flexible Data Format (fdf) files. It will save one image or a number of images in the case of multislice experiments. Currently the specified directory is made in the user's data directory, and will be appended with a ".dat". Image files will be created under this directory as "image0001.fdf", "image0002.fdf", and so on. A "procpar" file will also be saved into this directory.

Arguments The '`outfmt`' parameter is an optional character which defines the type of image data. It can take two character values:

- 'f' - Outputs the data in floating point format.
- 'm' - Outputs the data in 12 bit integer values in 16 bit words.

The default is 'f' (floating point) and currently ImageBrowser only accepts data in floating point values. The macro only saves images with the new imaging parameters that support oblique imaging. Unlike "`svsis`" the macro does not care about the name of the sequence. It does however format the header according to the following parameters.

- `seqcon - Sequence loop control flag`
- `nD - Data dimension assumed to be 2.`
- `tn,dm - Transmitter Nucleus (string)`
- `sfrq,dfrq - Spectrometer frequency (MHz)`
- `lro - Size of FOV for read out axis (cm)`
- `lpe - Size of FOV for phase encode axis (cm)`
- `pro - Position of image center on the read out axis (cm)`
- `ppe - Position of image center on 2D phase encode axis (cm)`
- `thk - Slice thickness (mm)`
- `pss - Slice position (cm)`
- `psi,phi,theta - Euler angles determining direction.`

The macro uses a Vnmr command "`svsdfd`" to dump the transformed data out to the data file. After dumping the headers out a unix shell command "`fdfgluer`" is called to glue the headers to the data. The "`svsdfd`" command dumps the data in such a way that the (0,0) coordinates are the first data point in the file.

NOTES: Modifications to the macro should be made in the user's maclib. The output values of the direction cosines may not be correct.

See also `svsis`

`svllj` Save peak listing in JCAMP-DX X,Y or X,Y,M format (M)

Syntax `svllj<(filename<,'all'><,'noll'>)>`

Applicability VnmrJ 3.1

Description `"svllj"` saves a peak listing in X,Y format. If a file "dept.out" exists in the current experiment, peak multiplicities are added to the output as well (X,Y,M format).

Arguments "all" ("svllj" only) causes solvent signals to be included in the peak listing (multiplicity marked as "U" = unassigned)

"noll" ("svllj" only) causes "svllj" NOT to re-evaluate the line listing - the contents of the parameters "`llfrq`" and "`llamp`" are used instead.

Examples
```
svllj
svllj('myspectrum')
svllj('myspectrum','all')
svllj('myspectrum','noll')
svllj('myspectrum','noll','all')
select(3) svllj('myspectrum_3')
```

See also svfj

Related
`listparam` list parameters in simple format (UNIX)
`writetrace` write ascii file from phasefile (f1 or f2) trace (M)

svlsj Save large dynamic range spectrum in JCAMP-DX format (M)

Syntax `svlsj<(filename<,opt>)>`

Applicability VnmrJ 3.1

Description `"svlsj"` is the same as "`svsj`", except that the spectrum is saved with 8 extra bits of digital precision ("`svsj`" saves spectra with 16-bit precision), for spectra with very large dynamic range.

Arguments "filename" is the name of the target file. If no filename is supplied, the software checks if "file" differs from "`exp`" (i.e., you have used "`rt`" to load the dataset from disk). If "file<>'exp'" and if the FID file is writable, then the JCAMP-DX data are saved as "{file}/dx_name.dx", where "{file}" has ".fid" added, if necessary, and "dx_name.dx" is the "basename" part of "file" (minus the ".fid" extension). If "file='exp'" or if the FID directory is not writable, the user is prompted for the filename.

The resulting ASCII file by default has a ".dx" extension, unless the specified filename has an alternative extension. (the default can be altered in the header of the macro). NOTE: ALL files created - whether they are FIDs, spectra or line lists - have the same (default) extension (the distinction between FIDs and spectra is made within the JCAMP-DX format); it is up to the user to avoid overwriting files! Where multiple arguments are specified / allowed, "filename"MUST be the first argument.

"opt" is an optional argument that permits controlling the compression in the data part of the JCAMP-DX file. Possible options and their effects are:

Table 4. Option

	tbl	fix	pac	sqz	dup	dif	difdup	Effect	Description
X,Y	X	-	-	-	-	-	-	X,Y list	
FIX	-	X	-	-	-	-	-	X(Y..Y)	readable
PAC	-	-	X	-	-	-	-	X(Y..Y)	packed
SQZ	-	-	-	X	X	X	X	X(Y..Y)	squeezed
DIF	-	-	-	-	-	X	X	SQZ	differences
DUP	-	-	-	-	X	-	X	SQZ	DUP mode

The default mode / option is "'difdup'" which usually gives the best compression. For human-readable data use the "'fix'" option, for programs expecting output in X Y format (one ordinate value per line) use the "'tbl'" option. All format options comply with the JCAMP-DX format and should be usable.

Examples
```
svlsj
svlsj('myspectrum')
svlsj('myspectrum','tbl')
select(3) svlsj('myspectrum_3')
```

See also svfj
svsj

Related listparam list parameters in simple format (UNIX)
writetrace write ascii file from phasefile (f1 or f2) trace (M)

svp Save parameters from current experiment (M)

Syntax `svp(file) <(file<,'force'><,'nodb'>)>`

Description Saves parameters from current experiment to a file. The parameter set can be retrieved with the `rtp` and `rt` macros. `svp` reflects any changes made in parameters up to the moment of entering `svp`, including acquisition parameters (unlike macro `svf`).

Arguments `file` is the name of the file, with the suffix `.par` added, to be created to save the parameters. The default is the system prompts for a file name. You are warned if you attempt to overwrite a parameter set that already exists.

If `force` is given, you are not warned and the older parameter set is removed.

`nodb` is a keyword to prevent `svp` from adding information to a database. This prevention is useful if temporary parameter files are saved that will soon be removed.

Examples
```
svp('/vnmr/stdpar/P31')
svp('/usr/george/testdata')
```

See also *NMR Spectroscopy User Guide*

Related		
	rt	Retrieve FID (M)
	rtp	Retrieve parameters (M)
	svf	Save FIDs in current experiment (M)

svpdp

Description Compares current workspace parameters to the parameter file. Any current workspace parameter values that are different from the parameter file are updated in the parameter file.

Syntax `svpdp<(parlib)>`

Arguments target parameter library

svs Save shim coil settings (C)

Syntax `svs(file)<:status>`

Description Saves all shim coil settings except Z0 to a file.

Arguments `file` is the name of a file for saving the shim coil settings. If the file name is an absolute path, `svs` uses it with no modifications. Otherwise, `svs` saves the shim in the first application directory for which it has write permission.

The `svs` command reports where it stored the shims, unless it is requested to return the status.

`status` is a return variable with one of the following values after `svs` finishes:

- 0 indicates `svs` failed to store shim file.
- 1 indicates `svs` stored the shim file, either as an absolute path or in the `shims` directory of the first application directory.
- >=2 indicates `svs` stored the file in `shims` directory of the second, third, or later application directory.

Examples
```
svs('acetone')
svs('bb10mm'):r1
```

See also *NMR Spectroscopy User Guide*

Related		
	rts	Retrieve shim coil settings (C)

svs Spin simulation vertical scale (P)

Description Vertical scale for simulated spectrum.

Values 0 to 1e10. A typical value is 200.

See also *NMR Spectroscopy User Guide*

Related
spins Perform spin simulation calculation (C)
spsm Enter spin system (M)

svsis Generate and Save images as FDF files. (macro)

Syntax `svsis('directory_name'[,'outfmt'])`

Applicability VnmrJ 3.1

Description The "svsis" command generates images from the current experiment and saves them into the specified directory as Flexible Data Format (fdf) files. It will save one image or a number of images in the case of multislice experiments. Currently the specified directory is made in the user's data directory, and will be appended with a ".dat". Image files will be created under this directory as "image0001.fdf", "image0002.fdf", and so on. A "procpar" file will also be saved into this directory.

Arguments The 'outfmt' parameter is an optional character which defines the type of image data. It can take two character values:

- 'f' - Outputs the data in floating point format.
- 'm' - Outputs the data in 12 bit integer values in 16 bit words.

The default is 'f' (floating point) and currently ImageBrowser only accepts floating point data.

The macro only saves images from the standard SISCO imaging sequences: "image", "shorte", "stecho", "multiecho", "csi2D", and "ssfp". However, it can be easily modified to produce images from users own sequences provided the sequences use standard SISCO parameters, slice select pulse shapes, and generate data in the same manner as the standard SISCO sequences.

To easily modify the macro to use a user's sequence the user need only add a line similar to the following in the "Valid Sequences" section:

$k=$k+1 $seqfil[$k]='t1image' $seq[$k]='ncsnn' $thk[$k]='image'

The new sequence name is 't1image'. Its reconstruction properties are given by $seq whose values are similar to the parameter "seqcon". "seqcon"'s characters are defined as follows:

- First character: multiecho looping
- Second charcter: multislice looping
- Third charcter: 2D phase encode loop
- Fourth character: 3D phase encode loop
- Fifth character: 4D phase encode loop

The values of each character are:

- 'n': null loop
- 's': standard loop
- 'c': compressed loop

In this case 'ncsnn' is a standard 2D image with compressed multislice. The $thk value is the slice thickness type defined by the type of acquisition which in this case is the standard 'image' sequence.

More detailed modifications can be made to the macro but it is left to the user to make these adjustments. The macro uses a Vnmr command "svsdfd" to dump the transformed data out to the data file. After dumping the headers out a unix shell command "fdfgluer" is called to glue the headers to the data. The "svsdfd" command dumps the data in such a way that the (0,0) coordinates are the first data point in the file.

NOTE: Modifications to the macro should be made in the user's maclib.

See also svimg

svsj Save spectrum in JCAMP-DX format (M)

Syntax svsj<(filename<,opt>)>

Applicability VnmrJ 3.1

Description "svsj" saves the current 1D spectrum in JCAMP-DX format. "svsj" creates a temporary file "/vnmr/tmp/jdxspec"; it calls two external C programs "listparam" and "jdxspec". Only a single 1D trace (the current trace in the case of an arrayed experiment) is saved; "svsj" does not work on 2D data after "wft1d" or "wft2d", but 2D data can be treated as arrayed 1D data sets using "wft" / "ft", which again permits saving traces.

Arguments "filename" is the name of the target file. If no filename is supplied, the software checks if "file" differs from "exp" (i.e., you have used "rt" to load the dataset from disk). If "file<>'exp'" and if the FID file is writable, then the JCAMP-DX data are saved as "{file}/dx_name.dx", where "{file}" has ".fid" added, if necessary, and "dx_name.dx" is the "basename" part of "file" (minus the ".fid" extension). If "file='exp'" or if the FID directory is not writable, the user is prompted for the filename.

The resulting ASCII file by default has a ".dx" extension, unless the specified filename has an alternative extension. (the default can be altered in the header of the macro). NOTE: ALL files created - whether they are FIDs, spectra or line lists - have the same (default) extension (the distinction between FIDs and spectra is made within the JCAMP-DX format); it is up to the user to avoid overwriting files! Where multiple arguments are specified / allowed, "filename"MUST be the first argument.

"opt" is an optional argument that permits controlling the compression in the data part of the JCAMP-DX file. Possible options and their effects are:

Table 5. Option

	tbl	fix	pac	sqz	dup	dif	difdup	Effect	Description
X,Y	X	-	-	-	-	-	-	X,Y list	
FIX	-	X	-	-	-	-	-	X(Y..Y)	readable
PAC	-	-	X	-	-	-	-	X(Y..Y)	packed
SQZ	-	-	-	X	X	X	X	X(Y..Y)	squeezed
DIF	-	-	-	-	-	X	X	SQZ	differences
DUP	-	-	-	-	X	-	X	SQZ	DUP mode

The default mode / option is "'difdup'" which usually gives the best compression. For human-readable data use the "'fix'" option, for programs expecting output in X Y format (one ordinate value per line) use the "'tbl'" option. All format options comply with the JCAMP-DX format and should be usable.

Examples
```
svsj
svsj('myspectrum')
svsj('myspectrum','fix')
select(3) svsj('myspectrum_3')
```

Related
`listparam` list parameters in simple format (UNIX)
`writetrace` write ascii file from phasefile (f1 or f2) trace (M)

`svtmp` Move experiment data into experiment subfile (M)

Syntax `svtmp<(file)>`

Description Moves the experiment data (parameters, FID, and transformed spectrum) from current experiment into a subdirectory inside `curexp+'/subexp'`. Unlike the macro `cptmp`, the experiment data is no longer accessible in the current experiment; only a copy of the parameters is still present.

Arguments `file` is the name of the subfile that receives the experiment data. The default name is either the transmitter nucleus (if `seqfil='s2pul'`) or the pulse sequence name.

Examples
```
svtmp
svtmp('cosy')
```

See also *NMR Spectroscopy User Guide*

Related
`cptmp` Copy experiment data into experiment subfile (M)
`curexp` Current experiment directory (P)
`rttmp` Retrieve experiment data from experiment subfile (M)
`seqfil` Pulse sequence name (P)

svxyj Save spectrum in JCAMP-DX X,Y format (M)

Syntax `svxyj<(filename)>`

Applicability VnmrJ 3.1

Description "`svxyj`" is similar to "`svsj`", except that the spectrum is written out in X,Y (2-column) format, with referenced X values and Y values directly in mm (the other JCAMP-DX formats use a simple integer X and Y values, the scaling and referencing information is stored in header fields. NOTE: most JCAMP-DX import software expects "`svsj`" / "`svlsj`" output. "`svxyj`" output uses no compression - the resulting files are much bigger than with any of the output options of the other JCAMP-DX conversion macros for full spectra.

Arguments "filename" is the name of the target file. If no filename is supplied, the software checks if "file" differs from "`exp`" (i.e., you have used "`rt`" to load the dataset from disk). If "`file<>'exp'`" and if the FID file is writable, then the JCAMP-DX data are saved as "`{file}/dx_name.dx`", where "`{file}`" has ".fid" added, if necessary, and "dx_name.dx" is the "basename" part of "file" (minus the ".fid" extension). If "file='exp'" or if the FID directory is not writable, the user is prompted for the filename.

The resulting ASCII file by default has a ".dx" extension, unless the specified filename has an alternative extension. (the default can be altered in the header of the macro). NOTE: ALL files created - whether they are FIDs, spectra or line lists - have the same (default) extension (the distinction between FIDs and spectra is made within the JCAMP-DX format); it is up to the user to avoid overwriting files! Where multiple arguments are specified / allowed, "filename"MUST be the first argument.

Examples
```
svxyj
svxyj('myspectrum')
select(3) svxyj('myspectrum_3')
```

Related
`listparam` list parameters in simple format (UNIX)
`writetrace` write ascii file from phasefile (f1 or f2) trace (M)

sw Spectral width in directly detected dimension (P)

Description Sets the total width of the spectrum to be acquired, from one end to the other. All spectra are acquired using quadrature detection. The spectral width determines the sampling rate for data, which occurs at a rate of `2*sw` points per second (actually `sw` pairs of complex points per second). Note that the sampling rate itself is not entered, either directly or as its inverse (known on some systems as the *dwell time*).

If a value of `sw` is entered whose inverse is not an even multiple of the time base listed above, `sw` is automatically adjusted to a slightly different value to give an acceptable sampling rate.

To enter a value in ppm, append the character `p` (e.g., `sw=200p`).

If a DSP facility is present in the system (i.e., dsp='i' or dsp='r') and oversampling in the experiment has not been turned off by setting oversamp='n', then the oversampling factor will be recalculated.

Values Number, in Hz. The range possible is based on the system:

100 Hz to 500 kHz.

solids systems: up to 5 MHz.

See also *NMR Spectroscopy User Guide*

Related

dp	Double precision (P)
dsp	Type of DSP for data acquisition (P)
oversamp	Oversampling factor for acquisition (P)
setlp0	Set parameters for zero linear phase (M)
sw1	Spectral width in 1st indirectly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

sw1 Spectral width in 1st indirectly detected dimension (P)

Description Analogous to the sw parameter except that sw1 applies to the first indirectly detected dimension of a multidimensional data set. The increment of the variable evolution time d2 is automatically calculated from sw1. The number of increments for this dimension is set by ni. To create sw1 in the current experiment, as well as ni and phase, enter addpar('2d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d2	Incremented delay in 1st indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
phase	Phase selection (P)
sw	Spectral width in directly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

sw2 Spectral width in 2nd indirectly detected dimension (P)

Description Analogous to the sw parameter except that sw2 applies to the second indirectly detected dimension of a multidimensional data set. The increment of the variable evolution time d3 is automatically calculated from sw2. The number of increments for this dimension is set by ni2. To create sw2 in the current experiment, as well as d3, ni2, and phase2, enter addpar('3d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d3	Incremented delay for 2nd indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
phase2	Phase selection for 3D acquisition (P)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 2nd indirectly detected dimension (P)
sw3	Spectral width in 3rd indirectly detected dimension (P)

sw3 — Spectral width in 3rd indirectly detected dimension (P)

Description Analogous to the sw parameter except that sw3 applies to the third indirectly detected dimension of a multidimensional data set. The increment of the variable evolution time d4 is automatically calculated from sw3. The number of increments for this dimension is set by ni3. To create sw3 in the current experiment, as well as d4, ni3, and phase3, enter addpar('4d').

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
d4	Incremented delay for 3rd indirectly detected dimension (P)
ni3	Number of increments in 3rd indirectly detected dimension (P)
par4d	Create 4D acquisition parameters (C)
phase3	Phase selection for 4D acquisition (P)
sw	Spectral width in directly detected dimension (P)
sw1	Spectral width in 1st indirectly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

sysgcoil — System gradient coil (P)

Description Specially reserved string parameter that specifies which physical gradient set is currently installed, and allows convenient updating of important gradient characteristics when one gradient set is interchanged for another. The value to sysgcoil is assigned to the parameter gcoil when joining experiments or retrieving parameter sets.

This parameter is set in the Spectrometer Configuration window to the name of the gradient set in use. Once set, it is then available to all experiments and to all users.

See also *VnmrJ Installation and Administration*; *VnmrJ Imaging NMR*

Related
config	Display current configuration and possibly change it (M)
gcoil	Current gradient coil (P)
gmax	Maximum gradient strength (P)
setgcoil	Assign sysgcoil configuration parameter (M)

system System type (P)

Description A global parameter that sets the basic type of system: spectrometer or data station. The value is set using the System Type label in the Spectrometer Configuration window.

Values `'spectrometer'` is a spectrometer system (Spectrometer choice in Spectrometer Configuration window).

`'datastation'` is a system used as a data station (Data Station choice in Spectrometer Configuration window). Acquisition is not allowed in this setting.

See also *VnmrJ Installation and Administration*

Related
config	Display current configuration and possibly change it (M)
Console	System console type (P)

systemdir VnmrJ system directory (P)

Description Contains path to VnmrJ system directory, typically `/vnmr`. The UNIX environmental variable `vnmrsystem` initializes `systemdir` at bootup.

See also *NMR Spectroscopy User Guide*

T

t1	T_1 exponential analysis (M)
t1s	T_1 exponential analysis with short output table (M)
t2	T_2 exponential analysis (M)
t2s	T_2 exponential analysis with short output table (M)
tabc	Convert data in table order to linear order (M)
tan	Find tangent value of an angle (C)
tape	Read tapes from VXR-style system (M,U)
tape	Control tape options of files program (P)
target_bval	Adjust gdiff to achieve target b-value (M)
tcapply	Apply Table Conversion Reformatting to Data (C)
tchan	RF channel number used for tuning (P)
tcl	Send Tcl script to Tcl version of dg window (C)
tcclose	Table Convert Close (C)
tcopen	Table Convert Open (C)
temp	Open the Temperature Control window (C)
temp	Sample temperature (P)
tempcal	Temperature calculation (C)
tempcalc	Measure approximate sample temperature in Cold Probes (M)
testacquire	Test acquire mode (P)
testct	Check ct for resuming signal-to-noise testing (M)
testsn	Test signal-to-noise of a spectrum (M)
teststr	Find which array matches a string M)
text	Display text or set new text for current experiment (C)
textis	Return the current text display status (C)
textvi	Edit text file of current experiment (M)

th	Threshold (P)
th2d	Threshold for integrating peaks in 2D spectra (P)
thadj	Adjust threshold for peak printout (M)
time	Display experiment time or recalculate number of transients (M)
tin	Temperature interlock (P)
tlt	First-order baseline correction (P)
tmove	Left-shift FID to time-domain cursor (M)
tmsref	Reference 1D proton or carbon spectrum to TMS (M)
tn	Nucleus for observe transmitter (P)
tncosyps	Set up parameters for TNCOSYPS pulse sequence (M)
tndqcosy	Set up parameters for TNDQCOSY pulse sequence (M)
tnmqcosy	Set up parameters for TNMQCOSY pulse sequence (M)
tnnoesy	Set up parameters for TNNOESY pulse sequence (M)
tnroesy	Set up parameters for TNROESY pulse sequence (M)
tntocsy	Set up parameters for TNTOCSY pulse sequence (M)
Tocsy	Convert the parameters to a TOCSY experiment (M)
Tocsy1d	Convert the parameter set to a Tocsy1d experiment (M)
tocsyHT	Set up the tocsyHT experiment (M)
tof	Frequency offset for observe transmitter (P)
tpwr	Observe transmitter power level with linear amplifiers (P)
tpwrf	Observe transmitter fine power (P)
tpwrm	Observe transmitter linear modulator power (P)
trace	Mode for *n*-dimensional data display (P)
traymax	Sample changer tray slots (P)
trfunc	Translates screen co-ordinates to hertz or centimeters depending upon the axis parameter
trfuncd	Translates a screen distance into centimeters in a real image
troesy	Set up parameters for TROESY pulse sequence (M)
trtune	Allows the user to view multiple tuning traces apparently simultaneously
trunc	Truncate real numbers (O)
tshift	Adjust tau2 to current cursor position (M)
tugain	Amount of receiver gain used by qtune (P)

tune	Assign a frequency to a channel for probe tuning (C)
tunehf	Tune both H1 and F19 on an HFX probe (M)
tunematch	Default match target, in percent of optimum (P)
tunemethod	Method to use for tuning (P)
tuneResult	Message indicating how well the tuning succeeded (P)
tunerp	A pulse sequence for pulse tuning through the directional couplers in the VNMRJ display
tunesw	Width of the tuning sweep in Hz (P
tupwr	Transmitter power used in tuning (P)
typeof	Return identifier for argument type (O)

t1 — T_1 exponential analysis (M)

Description: Processes data obtained using an array of values of the parameter d2 for a T_1 experiment. It runs expfit, which does an exponential curve fitting that determines the value of T_1. The output is matched to the equation:

```
M(t)=(M(0)-M0)*exp(-t/T1)+M0
```

where *M0* is the equilibrium Z magnetization and *M(0)* is the magnetization at time zero (e.g., immediately after the 180° pulse for an inversion recovery T_1 experiment). Notice that this equation will fit inversion recovery data (for which *M(0)* is approximately equal to –*M0*) or saturation recovery data (for which *M(0)* is 0).

The required input is the file fp.out from fp and the values of the arrayed parameter. The T_1 analysis is done for all the peaks listed in fp.out. Peaks are selected for analysis by entering fp(index1,index2,...) before running the analysis. The output file is the analyze.list in the current experiment. The file analyze.out is used by exp1 to display the results. The output of the analysis program shows T_1 and its standard deviation, but does not explicitly show *M(0)*, *M0*, or their standard deviations. The *M(0)* and *M0* values can be found in “raw” form in analyze.out in the current experiment, but their standard deviations are not part of the program output.

See also: *NMR Spectroscopy User Guide*

Related:

d2	Incremented delay in 1st indirectly detected dimension (P)
expfit	Make least squares fit to polynomial or exponential curve (C)
fp	Find peak heights (C)
t1s	T_1 exponential analysis with short output table (M)
t2	T_2 exponential analysis (M)
t2s	T_2 exponential analysis with short output fable (M)

t1s T_1 exponential analysis with short output table (M)

Description Performs the same analysis as t1 but produces a short output table showing only a summary of the measured relaxation times.

See also *NMR Spectroscopy User Guide*

Related t1 T_1 exponential analysis (M)

t2 T_2 exponential analysis (M)

Description Processes data obtained using an array of values for the base time parameter bt for a T_2 experiment. It runs expfit, which does an exponential curve fitting that determines the value of T_2. The output is matched to the equation:

$M(t) = (M(0) - M(inf))*exp(-t/T2) + M(inf)$

where *M(0)* is the magnetization at time zero (i.e., the full magnetization excited by the observe pulse) and *M(inf)* is the xy-magnetization at infinite time (zero unless the peak is sitting on an offset baseline).

The required input is the file fp.out from fp and the values of the arrayed parameter. The T_2 analysis is done for all the peaks listed in fp.out. Peaks are selected for analysis by entering fp(index1,index2,...) before running the analysis. The output file is the file analyze.list in the current experiment. The file analyze.out is used by exp1 to display the results. The output of the analysis program shows T_2 and its standard deviation, but does not explicitly show *M(0)*, *M(inf)*, or their standard deviations. The *M(0)* and *M(inf)* values can be found in “raw” form in analyze.out in the current experiment, but their standard deviations are not part of the program output.

See also *NMR Spectroscopy User Guide*

Related

expfit	Make least squares fit to polynomial or exponential curve (C)
fp	Find peak heights (C)
t1	T_1 exponential analysis (M)
t1s	T_1 exponential analysis with short output table (M)
t2s	T_2 exponential analysis with short output fable (M)

t2s T_2 exponential analysis with short output table (M)

Description Performs the same analysis as t2 but produces a short output table showing only a summary of the measured relaxation times.

See also *NMR Spectroscopy User Guide*

Related `t2` T_2 exponential analysis (M)

tabc — Convert data in table order to linear order (M)

Syntax `tabc<(dimension)>`

Description Converts arbitrarily ordered data obtained under control of an external AP table to linear monotonic order, suitable for processing in VnmrJ. The data must have been acquired according to a table in the `tablib` directory.

Imaging and other 2D experiments are normally acquired so that the order of the incremented acquisition parameter, such as the phase-encode gradient, is linear and monotonic. For a standard imaging experiment, this linear order means that the phase-encode gradient progresses from a starting negative value monotonically up through zero to a positive value (e.g., –64, –63, –62, ... , –1, 0, 1, ... , 62, 63). The `ft2d` program assumes this structure in its operation.

Data from table-driven 2D pulse sequences is used by entering `tabc` *only once* before normal 2D processing and/or parameter storage. In this situation, `tabc` takes no arguments and is executed by entering `tabc` in the command window. A simple check is done by `tabc` to prevent it from being executed more than once on the same data set.

2D data is expected to be in the standard VnmrJ format, but if the 2D data is in the compressed format, setting `dimension` to `1` converts the data. `tabc` supports all 2D data types recognized by VnmrJ: arrayed, compressed multislice, and arrayed compressed multislice,

3D data is expected to be in the compressed/standard format, in which there are `ni` standard 2D planes of data (the third dimension), each consisting of `nf` compressed FIDs (the second dimension). Setting `dimension` to `3` reorders 3D data acquired with an external table.

`tabc` reads the file `fid` in the `acqfil` subdirectory of the current experiment. Before the data is reordered, this file is written to the file `fid.orig` in the same `acqfil` directory. If for any reason `tabc` fails or results in an unpredictable or undesired transformation, the original raw data can be recovered by moving `fid.orig` back to `fid`. To gain more disk space, you can delete `fid.orig` after you are satisfied that conversion is successful.

Use `tabc` on saved data that has been loaded into an experiment or on data in an experiment that has just been acquired but not yet saved. In the first case, converted data must be resaved for the saved data set to reflect conversion.

`tabc` requires that data must have the same number of “traces” as the table elements. It does not support any of the advanced features of table expansion (e.g., the entire table must be explicitly listed in the table file), and expects to find only one table in a file; whether the table is t1 or t60 is unimportant.

Arguments `dimension` specifies the type of data to be converted: `1` for 2D compressed data, `2` for 2D standard data, or `3` for 3D compressed/standard data. The default is 2.

Examples
```
tabc
tabc(1)
tabc(3)
```

See also *VnmrJ Imaging NMR*

Related

`flashc`	Convert compressed 2D data to standard 2D format (C)
`ft2d`	Fourier transform 2D data (C)
`ni`	Number of increments in 1st indirectly detected dimension (P)
`nf`	Number of FIDs (P)

tan — Find tangent value of an angle (C)

Syntax `tan(angle)<:n>`

Description Finds the tangent of an angle.

Arguments `angle` is an angle, in radians.

`n` is the return value giving the tangent of `angle`. The default is to display the tangent value in the status window.

Examples
```
tan(.5)
tan(val):tan_val
```

See also *User Programming*

Related

`atan`	Find arc tangent value of a number (C)
`cos`	Find cosine value of an angle (C)
`exp`	Find exponential value of a number (C)
`ln`	Find natural logarithm of a number (C)
`sin`	Find sine value of an angle (C)

tape — Read tapes from VXR-style system (M,U)

Syntax (From VnmrJ) `tape(<-d device,><type,>option <,file1,file2,...>)`
(From UNIX) `tape <-d device> <type> <option> <file1> <file2>...`

Description Displays the contents of a VXR-style (Gemini, VXR-4000, or XL) 9-track tape for use with VnmrJ or reads one or several files from the tape into the current directory. Note that the *write* option is not supported (i.e., VnmrJ only *reads* tapes in a VXR-style format and does not write to a tape).

Arguments `device` is the tape drive device name. The default value is `/dev/rst8`. For AIX systems, `device` should be `/dev/rmt0`.If the default value is not set properly or another device name is wanted, be sure to type `-d` and a space before the device name you want to input.

`type` is the type of tape to be accessed. `'-q'` or `'-s'` select the 1/4-inch tape unit ("streaming" or cartridge tape); this is the default. `'-9'`, `'-h'`, or `'-n'` select the 1/2- inch tape unit (open reel tape drive).

`option` is one of the following:

- `'help'` is a keyword to display help on the use of the system.
- `'cat'` is a keyword to display a catalog of files on tape.
- `'read'` is a keyword to read one or more files. This option requires that the files be listed as the next argument.
- `'rewind'` is a keyword to rewind tape (1/2-inch tape only).
- `'quit'` is a keyword to release the tape drive (1/2-inch tape only).

`file1, file2, ...` are the names of one or more files to be read. Wildcard characters (* and ?) can be used.

Examples

```
tape('cat')
tape('-h','read','mydata')
tape -h read mydata
tape -d /dev/rmt/0lb read mydata
```

Related

Related	`decomp`	Decompose a VXR-style directory (C)
	`vxr_unix`	Convert VXR-style text files to UNIX format (M,U)

`tape` Control tape options of files program (P)

Description Defines device that `files` program accesses when it is instructed to read or write to a tape. The parameter `tape` is in the user's global parameter tree.

Values Name of a device. The default device is `/dev/rst8`. If `tape` does not exist or is set to the null string (two single quotes with no space between), `files` uses its default device value. Notice that different computers define tape drives differently. For VnmrSGI, `tape='/dev/tapens'` is appropriate. For Solaris, `tape='/dev/rmt/0mb'`.

Related	`files`	Interactively handle files (C)

`target_bval` Adjust `gdiff` to achieve target b-value (M)

Applicability Imaging Systems

Syntax `target_bval(value)`

Description This macro iteratively adjusts gdiff and calls the sequence (go('check')) to achieve the target b-value. The sequence is evoked because the contributions from the imaging gradients must be taken into account backwards calculation of b is not possible because the relationship between gdiff and b-value is not simple. The macro defaults to getting within 1 s/mm2 of the target or maximum of 20 iterations and exits if either condition is met.

Arguments value, the target b-value in s/mm2.

Examples target_bval(1000)

See also *VnmrJ Imaging User's Guide*

tcapply Apply Table Conversion Reformatting to Data (C)

Syntax tcapply([<filename>])

Applicability VnmrJ 3.1

Description "tcapply" rearranges the spectra in a 2D dataset that reside in the current datafile. Using values from an AP table, it arranges the spectra corresponding to the value in the AP table from low value to high value. The values may have already been read in by the "tcopen" command or if the optional <filename> argument has been provided the values will be read in from $vnmruser/tablib/<filename>.

As mention before, this command uses spectra from the current datafile; which means that a "ft1d" should have been done on the data before using this command. To give an example, for a standard imaging experiment the phase encode gradients will progress from a starting negative value monotonically up through zero to a positive value, e.g.:

-64, -63, -62, ... , -1, 0, 1, ... , 62, 63.

It is possible to acquire the equivalent data in non-monotonic order, either by explicitly coding the desired progression into a pulse sequence, or by using an external AP table to control the order. In either case, "ft2d" will not be able to properly process the resulting data. "tcapply" and "tabc" are functions which reconstruct a properly ordered data set from any arbitrarily ordered data which has been acquired under control of an external AP table. The data must have been acquired according to a table in the "tablib" directory. The different between "tcapply" and "tabc" is that "tcapply" works on the first dimension transformed spectra residing in Vnmr's data memory and "tabc" works on and changes the raw data in the fid file.

Arguments 'filename' optional argument specifying the AP table to be read which resides in $vnmruser/tablib/<filename>.

Examples
```
ft1d(2)
tcapply(petable)
ft2d(2)
```

Related

tcclose	Table Convert Close
tcopen	Table Convert Open
tabc	

tchan RF channel number used for tuning (P)

Description Set by the protune macro.

See also *NMR Spectroscopy User Guide*

Related
protune	Macro to start ProTune (M)
atune	ProTune Present (P)
mtune	Tune probe using swept-tune graphical display (M)
tugain	Receiver gain used in tuning (P)
tunesw	Width of the tuning sweep in Hz (P
tupwr	Transmitter power used in tuning (P)

tcl Send Tcl script to Tcl version of dg window (C)

Syntax `tcl(script)`

Description Sends a Tcl (Tool Command Language) script to the Tcl version of the `dg` window. If this window is not active, this command does nothing.

Arguments `script` is any legal Tcl script.

See also *User Programming*

Related `dg` Display group of acquisition/processing parameters (C)

tcclose Table Convert Close (C)

Syntax `tcopen(<filename>)`
`tcclose`

Applicability VnmrJ 3.1

Description "`tcopen`" explicitly reads, sorts, and stores in memory a table convert file from `$vnmruser/tablib/<filename>` which it will then use when "`tcapply`" is called. Once the table has been read in "`tcclose`" command must be used to remove the table and free the memory used.

"`tcclose`" removes the table and frees the memory used to store the sorted table indices read in with a "`tcopen`" command.

Arguments 'filename' argument specifying the file to be read which resides in `$vnmruser/tablib/<filename>`.

Examples `tcopen(petable)`
`tcclose`

Related `tcapply` Apply Table Conversion Reformatting to Data

`temp` Open the Temperature Control window (C)

Applicability Systems with a variable temperature (VT) controller.

Description Opens the Temperature Control window, which has the following capabilities:

- Turn temperature control off.
- Set temperature control on at a specified temperature in degrees C.
- Enable temperature control from within an experiment using the `temp` parameter and the `su`, `go`, `ga`, or `au` macros. This mode is the default.
- Alternatively, turn off experiment control of the temperature and allow only the Temperature Control window (and `sethw`) to set the temperature. This mode has the advantage that, often times, `temp` is different between experiments. Joining a different experiment and entering `go` can unexpectedly change the temperature. This mode prevents this problem.
- Resetting the temperature controller when the temperature cable is reconnected to a probe.

See also *NMR Spectroscopy User Guide*

Related

`acqi`	Interactive acquisition display process (C)
`au`	Submit experiment to acquisition and process data (M)
`ga`	Submit experiment to acquisition and FT the result (M)
`go`	Submit experiment to acquisition (M)
`readhw`	Read current values of acquisition hardware (C)
`sethw`	Set values for hardware in acquisition system (C)
`su`	Submit a setup experiment to acquisition (M)
`temp`	Sample temperature (P)
`tin`	Temperature interlock (P)

`temp` Sample temperature (P)

Applicability Systems with a variable temperature (VT) module.

Description Sets the temperature of sample.

Values `'n'` or `-150` to `+200`, in steps of 0.1°C. `'n'` instructs the acquisition system not to change the VT controller and to ignore temperature regulation throughout the course of the experiment.

See also *NMR Spectroscopy User Guide*

Related

`readhw`	Read current values of acquisition hardware (C)
`temp`	Open the Temperature Control window (C)
`tempcal`	Temperature calculation (C)
`tin`	Temperature interlock (P)
`vtc`	Variable temperature cutoff point (P)

tempcal Temperature calculation (C)

Applicability Systems with a variable temperature (VT) module.

Syntax `tempcal(solvent)<:temperature>`

Description For exact determination of sample temperature when using the VT unit, a temperature calibration curve must be made for each probe used. All data, such as gas flow, must be noted. Use samples of ethylene glycol for high-temperature calibration, and use samples of methanol for low-temperature calibration. To make the calculation:

- Bring the sample to the desired temperature and allow sufficient time for equilibration, then obtain a spectrum.
- Next, align two cursors on the two resonances in the spectrum, then enter `tempcal('e')` for ethylene glycol, or enter `tempcal('m')` for methanol. The temperature is calculated based on the difference frequency between the cursors.

Arguments `solvent` is the sample solvent: `'glycol'`, `'e'`, or `'g'` for ethylene glycol, or `'methanol'` or `'m'` for methanol.

`temperature` returns the calculated value of the sample temperature. The default is the system displays the value.

Examples `tempcal('glycol')`
`tempcal('m'):temp`

See also *NMR Spectroscopy User Guide*

tempcalc Measure approximate sample temperature in Cold Probes (M)

Applicability Systems with Agilent, Inc. Cold Probes

Description Measure the approximate sample temperature and the actual sample temperature gradient and generate a report. Requires a ~1% HOD CH_3CN sample.

testacquire Test acquire mode (P)

Description Allows test acquisitions to be done while a study queue is active, without using the study queue. When this mode is enabled, acquisitions do not update the status of the currently loaded experiment in the study queue, and data is not saved in the study queue. This mode is set from the Test mode check box in the Acquisition menu or from the command line.

Syntax `testacquire=<'y' or 'n'>`

Values `'y'` test acquire mode enabled

'n' test acquire mode disabled

Related	acquire	Acquire data (M)
	save	Save data (M)

testct Check ct for resuming signal-to-noise testing (M)

Description Used by the testsn macro to decide when to resume testing of signal-to-noise. See the description of testsn for details.

See also *NMR Spectroscopy User Guide*

Related	ct	Completed transients (P)
	testsn	Test signal-to-noise of a spectrum (M)

testsn Test signal-to-noise of a spectrum (M)

Description Part of the automatic periodic signal-to-noise testing that occurs during various automated acquisitions, most notably c13. Transforms the data using fn=16000, and then baseline corrects, setting the left-most 10% of the spectrum and the right-most 2% as baseline. After the baseline correction, testsn uses getsn to calculate the signal-to-noise.

- If signal-to-noise exceeds the desired goal in parameter sn (found in the standard carbon parameter set /vnmr/stdpar/c13), testsn aborts the experiment using the command halt, which initiates processing according to the wexp parameter.
- If signal-to-noise is not reached, testsn estimates the signal-to-noise ratio at the end of the experiment. If signal-to-noise target will not be reached by then, it cancels subsequent signal-to-noise testing, but allows the experiment to proceed.
- If the signal-to-noise target will be reached before the end of the experiment, it saves the estimated number of transients required to reach the goal in the parameter r7 (using a conservative estimate), and then sets the processing at future blocks to be only testct, which simply tests if ct is greater than r7, and, if so, resumes testing of signal-to-noise with testsn.

See also *NMR Spectroscopy User Guide*

Related	c13	Automated carbon acquisition (M)
	fn	Fourier number in directly detected dimension (P)
	getsn	Get signal-to-noise estimate of a spectrum (M)
	halt	Abort acquisition with no error (C)
	r1–r7	Real parameter storage for macros (P)
	sn	Signal-to-noise ratio (P)
	testct	Check ct for resuming signal-to-noise testing (M)
	wexp	Specify action when experiment completes (C)

teststr — Find which array matches a string M)

Syntax `teststr(parameter,string <,tree>):$ret`

Description The `teststr` command requires at least two arguments. The first is the name of a string parameter. The first argument must generally be enclosed in single quotes. The `teststr` command needs the name of the parameter, not its values. The second is a string. The optional third argument is the parameter tree. The default is current.

Macro parameters can be used as the first argument. In this case, the third argument must be `'local'`.

This command sets `$ret` to the index of the array element that matches the second argument. If none of the array values of the parameter match the second argument, a zero is returned.

Examples `n1='hello','labas','gidday','hola','bonjour','ciao'`
`teststr('n1','labas'):r1`
sets `r1=2`, since `'labas'` matches element 2 of the `n1` array.

The elements do not need to be single words. For example,
`n1='good night','labanaktis','bonne nuit','gute Nacht','boa noite','buonas noces'`

`teststr('n1','boa noite'):r1`
sets `r1=5`. The strings must match exactly, including upper and lower case

`teststr('n1','gute nacht'):r1`
sets `r1=0`, since the lower case n in nacht does not match the upper case N in Nacht.

For local dollar variables, the `'local'` argument must be used. Again, enclose the name of the local parameter in single quotes.

`$greet='hello','labas','gidday','hola', 'ciao'`

`teststr('$greet','labas','local'):r1`

text — Display text or set new text for current experiment (C)

Syntax `text<(text_string)><:string_variable>`

Description Associated with each experiment is a text file, consisting of a block of text, that can be used to describe the sample and experiment. `text` allows displaying the text file and changing the text file for the current experiment. A UNIX text editor, such as `vi`, or the macro `textvi` can also be used to edit the text file of the current experiment.

Arguments `text_string` is a string of text that replaces the existing text file. The default is to display the text file in the current experiment. The characters `\\` or `\n` can be used in the string to denote a new line, and the characters `\t` can be used to denote a tab (see example below).

`string_variable` returns the text in `text_string` as a string variable. Thus, for example, the `text:n1` and `text(n1+'cosy` experiment') commands, where n1 is a string, can be used in a macro to add a "cosy experiment" to the text. An equivalent operation using the atext command would be atext('cosy experiment').

Examples `text('Sample 101\tCDCl3\\13 February')`

See also *NMR Spectroscopy User Guide*

Related		
	atext	Append string to the current experiment text (M)
	ctext	Clear the text of the current experiment (C)
	curexp	Current experiment directory (P)
	dtext	Display a text file in the graphics window (C)
	puttxt	Put text file into another file (C)
	textvi	Edit text file of current experiment (M)
	vnmrprint	Print text files (U)

textis Return the current text display status (C)

Syntax (1) `textis(command):$yes_no`
(2) `textis:$display_command`

Description Determines if a command given by the user currently controls the text window (syntax 1) or returns the name of the command currently controlling the text window (syntax 2).

Arguments command is the name of a command that potentially may be controlling the text window.

$yes_no returns 1 if command controls the text window, or 0 if it does not.

$display_command returns the name of the command currently controlling the text window.

Examples
```
textis:$display
if ($display = 'dg') then . . . endif
```

See also *User Programming*

Related		
	graphis	Return the current graphics display status (C)

textvi Edit text file of current experiment (M)

Description Edits the text file of the current experiment using the UNIX text editor vi. textvi is equivalent to the command vi(curexp+'/text').

See also *NMR Spectroscopy User Guide*

Related		
	edit	Edit a file with user-selectable editor (M)
	text	Display text or set new text for current experiment (C)
	vi	Edit text file with vi editor (M)

th — Threshold (P)

Description: Sets threshold for printout of peak frequencies so that peaks greater than th on the plot appear on any peak listings. th is always bipolar (i.e., negative peaks greater in magnitude than th also appear in peak listings).

Values: 0 to 1e9, in mm.

See also: *NMR Spectroscopy User Guide*

Related:

thadj	Adjust threshold for peak printout (M)

th2d — Threshold for integrating peaks in 2D spectra (P)

Description: Used by ll2d when determining the bounds of a peak and calculating its volume. To create the 2D peak picking parameters th2d and xdiag in the current experiment, enter addpar('ll2d').

Values: From 0.0 to 1.0. If th2d=1.0, ll2d integrates all points in the peak that are above the current threshold for the spectrum (i.e., the portion of the peak that can be seen in a contour plot of the spectrum). A smaller value causes ll2d to integrate a larger area when determining the volume of a peak. If th2d=0.5, for example, ll2d integrates all points in a peak that are above 0.5 times the current threshold.

See also: *NMR Spectroscopy User Guide*

Related:

addpar	Add selected parameters to the current experiment (M)
ll2d	Automatic and interactive 2D peak picking (C)
xdiag	Threshold for excluding diagonal peaks when peak picking (P)

thadj — Adjust threshold for peak printout (M)

Syntax: `thadj<(max_peaks<,noise_mult<,llarg1<,llarg2>>>)>`

Description: Adjusts the threshold th so that no more than a specified maximum number of peaks are found in a subsequent line listing (see nll) and so that th is at least a specified noise multiplier times the root-mean-square noise level.

Arguments: max_peaks is the maximum number of peaks in the displayed spectral range. The default is wc/4 (i.e., the threshold is adjusted such that ppf will produce a “reasonable” number of lines with any width of plot).

noise_mult is a noise multiplier used to calculate the minimum value for th from the size of the root-mean-square noise.

llarg1 is the noise_mult argument (the default is 3) to the nll command used inside this macro `llarg2` is the keyword argument (`'pos'`, `'neg'`, `'all'`; the default is `'all'`.) to the `nll` command used inside this macro.

Examples
```
thadj
thadj(50)
thadj(200,4)
thadj(200,4,2)
thadj(200,4,2,'pos')
```

See also *NMR Spectroscopy User Guide*

Related

nll	Find line frequencies and intensities (C)
ppf	Plot teak frequencies over spectrum (M)
th	Threshold (P)
vsadj	Automatic vertical scale adjustment (M)
vsadj2	Automatic vertical scale adjustment by powers of two (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
vsadjh	Automatic vertical scale adjustment for ^{1}H spectra (M)
wc	Width of chart (P)

time — Display experiment time or recalculate number of transients (M)

Syntax `time<(<hours,>minutes)>`

Description Estimates the acquisition time or recalculates the number of transients so that the total acquisition time is approximately the requested time. The parameters looked at when calculating the time per transient are `d1`, `d2`, `d3`, `at`, `ni`, `sw1`, `ni2`, and `sw2`.

Arguments `hours` and `minutes` are numbers making up a time to be used by the system to recalculate the parameter `nt` so that the total acquisition time is approximately the time requested; the default (no arguments) is for the system to estimate the acquisition time for a 1D, 2D, or 3D experiment using the parameters in the current experiment.

Examples
```
time
time(2,45)
```

See also *NMR Spectroscopy User Guide*

Related

at	Acquisition time (P)
d1	First delay (P)
d2	Incremented delay in 1st indirectly detected dimension (P)
d3	Incremented delay in 2nd indirectly detected dimension (P)
exptime	Display experiment time (C)
ni	Number of increments in 1st indirectly detected dimension (P)
ni2	Number of increments in 2nd indirectly detected dimension (P)
nt	Number of transients (P)

sw1	Spectral width in 1st indirectly detected dimension (P)
sw2	Spectral width in 2nd indirectly detected dimension (P)

tin Temperature interlock (P)

Description Controls error handling based on temperature regulation. If temperature regulation is lost, `tin` can be used to select whether an error is generated and acquisition is halted or whether a warning is generated and acquisition continues. In both cases, the lost regulation will cause `werr` processing to occur, thus providing a user-selectable mechanism to respond to VT failure.

Values `'n'` turns off the temperature interlock feature

`'w'` indicates the variable temperature regulation light is monitored during the course of the experiment and, if it starts to flash (regulation lost), a warning is generated; however, acquisition is not stopped.

`'y'` indicates the variable temperature regulation light is monitored during the course of the experiment and, if it starts to flash (regulation lost), the current data acquisition is stopped. The acquisition will not resume automatically if regulation is regained.

See also *NMR Spectroscopy User Guide*

Related

in	Lock and spin interlock (P)
werr	When error (P)

tlt First-order baseline correction (P)

Description When spectral display is active, the command `dc` turns on a linear drift correction (baseline correction). The result of this operation includes calculating a first-order baseline correction parameter `tlt`. The calculation is made by averaging of a small number of points at either end of the display and drawing a straight line baseline between them.

See also *NMR Spectroscopy User Guide*

Related

cdc	Cancel drift correction (C)
dc	Calculate spectral drift correction (C)
lvl	Zero-order baseline correction (P)

tmove Left-shift FID to time-domain cursor (M)

Description Provides an alternative method of left shifting time-domain data. To use this method, position the right time cursor at the place that should be the start of the FID, then enter `tmove`. This adjusts `lsfid` to left-shift the FID.

See also *NMR Spectroscopy User Guide*

Related `lsfid` Number of complex points to left-shift *np* FID (P)

`tmsref` Reference 1D proton or carbon spectrum to TMS (M)

Syntax `tmsref:tms_found`

Description Tries to locate a TMS line. If found, `tmsref` re-references the spectrum to the TMS line and returns a 1 to the calling macro; if not found, `tmsref` returns 0 and the referencing is left as it was. In the case of other signals (e.g., from silicon grease) immediately to the left of the TMS line (even if they are higher than the reference line), `tmsref` tries avoiding those by taking the rightmost line in that area, as long as it is at least 10% of the main Si-CH_3 signal. Large signals within 0.6 ppm for ^{1}H (or 6 ppm for ^{13}C) to the right of TMS may lead to misreferencing.

Arguments `tms_found` returns 1 if a TMS line was located or returns 0 if not.

See also *NMR Spectroscopy User Guide*

Related	`c13`	Automated carbon acquisition (M)
	`h1`	Automated proton acquisition (M)

`tn` Nucleus for observe transmitter (P)

Description Changing the value of `tn` causes a macro (`_tn`) to be executed that extracts values for `sfrq` and `tof` from lookup tables. The tables, stored in the directory `/vnmr/nuctables`, are coded by atomic weights.

Values In the lookup tables, typically given by `'H1'`, `'C13'`, `'P31'`, etc. The value `tn='lk'` sets the deuterium frequency, and also holds the lock current and switches the relay in the automated deuterium gradient shimming module, if present, so that deuterium signal may be observed without disturbing lock. The frequency is the same as `tn='H2'`.

See also *NMR Spectroscopy User Guide*

Related	`dn`	Nucleus for first decoupler (P)
	`dn2`	Nucleus for second decoupler (P)
	`dn3`	Nucleus for third decoupler (P)
	`sfrq`	Transmitter frequency of observe nucleus (P)
	`tof`	Frequency offset for observe transmitter (P)

`tncosyps` Set up parameters for TNCOSYPS pulse sequence (M)

Description Sets up a homonuclear correlation experiment (phase-sensitive version) with water suppression.

See also *NMR Spectroscopy User Guide*

tndqcosy Set up parameters for TNDQCOSY pulse sequence (M)

Applicability Systems with a linear amplifier on the observe channel and a T/R switch.

Description Sets up a 2D J-correlation experiment with water suppression.

See also *NMR Spectroscopy User Guide*

tnmqcosy Set up parameters for TNMQCOSY pulse sequence (M)

Applicability Systems with hardware digital phaseshifter for transmitting with direct- synthesis rf; otherwise, software small-angle phaseshifter for transmitting with the old-style rf is used.

Description Sets up a multiple-quantum filtered COSY experiment with water suppression.

See also *NMR Spectroscopy User Guide*

tnnoesy Set up parameters for TNNOESY pulse sequence (M)

Applicability Systems with a linear amplifier on the observe channel and a T/R switch.

Description Sets up a 2D cross-relaxation experiment with water suppression.

See also *NMR Spectroscopy User Guide*

tnroesy Set up parameters for TNROESY pulse sequence (M)

Description Sets up a rotating-frame NOE experiment with water suppression.

See also *NMR Spectroscopy User Guide*

tntocsy Set up parameters for TNTOCSY pulse sequence (M)

Applicability Systems with T/R switch, computer-controlled attenuators, and linear amplifiers on observe channel.

Description Sets up a total-correlation spectroscopy experiment (HOHAHA) with water suppression.

See also *NMR Spectroscopy User Guide*

Tocsy — Convert the parameters to a TOCSY experiment (M)

Description Convert parameters to a TOCSY experiment.

See also *NMR Spectroscopy User Guide*

Related

ft1dac	Combined arrayed 2D FID matrices (M)
ft2dac	Combined arrayed 2D FID matrices (M)
wft1dac	Combined arrayed 2D FID matrices (M)
wft2dac	Combined arrayed 2D FID matrices (M)

Tocsy1d — Convert the parameter set to a Tocsy1d experiment (M)

Description Convert the parameter set to a Tocsy1d experiment.

See also *NMR Spectroscopy User Guide*

Related

Proton	Set up parameters for ^{1}H experiment (M).
sel1d	Selective 1D protocols to set up (M).

tocsyHT — Set up the tocsyHT experiment (M)

Description Sets up parameters for a Hadamard-encoded tocsy experiment.

See also *NMR Spectroscopy User Guide*

Related

htofs1	Hadamard offset in `ni` (P)
fn1	Fourier number in 1st indirectly detected dimension (P)
ni	Number of increments in 1st indirectly detected dimension (P)
ft2d	Fourier transform 2D data (C)
sethtfrq1	Set Hadamard frequency list from a line list (M)
Tocsy	Set up parameters for a TOCSY pulse sequence (M)
htfrq1	Hadamard frequency list in `ni` (P)

tof — Frequency offset for observe transmitter (P)

Description Controls the exact positioning of the transmitter. As the value assigned to `tof` increases, the transmitter moves to a higher frequency (toward the left side of the spectrum). The minimum step size of `tof` is determined by the type of rf hardware in the spectrometer. The limit is specified using the Step Size label in the Spectrometer Configuration window. Systems with broadband style rf (rftype='b') generally have 100-Hz resolution; all other systems have 0.1 Hz resolution.

Values: Approximate, depends on frequency–100000 to 100000, in Hz.

See also: *NMR Spectroscopy User Guide*

Related:

config	Determine current configuration and possibly change it (M)
dof	Frequency offset for first decoupler (P)
dof2	Frequency offset for second decoupler (P)
dof3	Frequency offset for third decoupler (P)
rftype	Type of rf generation (P)

tpwr — Observe transmitter power level with linear amplifiers (P)

Applicability: Systems with a linear amplifier on the observe channel.

Description: Controls transmitter power. The value of the attenuator upper safety limit is set using the Upper Limit label in the Spectrometer Configuration window. Depending on hardware adjustments, the system may saturate at a given value of tpwr (i.e., values above a certain value may give equal output).

Values: On systems with 63-dB attenuator installed: 0 to 63 (63 is maximum power), in units of dB. About 55 to 60 is normal. Lower values (e.g., 49) might be used for water suppression experiments like 1-3-3-1.

On systems with 79-dB attenuator installed: –16 to 63 (63 is maximum power), in units of dB.

CAUTION

Continuous power greater than 2 watts in a switchable probe will damage the probe. Always carefully calibrate power to avoid exceeding 2 watts. The maximum value for tpwr on a 200-MHz, 300-MHz, or 400-MHz system with a linear amplifier on the decoupler channel has been set to 49, corresponding to about 2 watts of power. Before using tpwr=49 for continuous decoupling, ensure safe operation by measuring the output power. This should be done during system installation and checked periodically by the user.

See also: *NMR Spectroscopy User Guide*

Related:

cattn	Coarse attenuator (P)
config	Determine current configuration and possibly change it (M)
dpwr	Power level for first decoupler with linear amplifiers (P)
dpwr2	Power level for second decoupler (P)
dpwr3	Power level for third decoupler (P)
dpwrf	First decoupler fine power (P)
fattn	Fine attenuator (P)
tpwrf	Observe transmitter fine power (P)

`tpwrf` Observe transmitter fine power (P)

Applicability Systems with a fine attenuator on the observe transmitter channel.

Description Controls the transmitter fine attenuator. Systems with this attenuator are designated using the Fine Attenuator label in the Spectrometer Configuration window. The fine attenuator is linear and spans 60 dB or 6 dB. If `tpwrf` is not present, enter `create('tpwrf','integer')` `setlimit('tpwrf',4095,0,1)` to create it.

Values 0 to 4095, where 4095 is maximum power. If `tpwrf` does not exist in the parameter table, a value of 4095 is assumed.

See also *NMR Spectroscopy User Guide*

Related

`config`	Determine current configuration and possibly change it (M)
`dpwr`	Power level for first decoupler with linear amplifiers (P)
`dpwrf`	First decoupler fine power (P)
`fattn`	Fine attenuator (P)
`tpwr`	Observe transmitter power level with linear amplifier (P)
`tpwrm`	Observe transmitter linear modulator power (P)

`tpwrm` Observe transmitter linear modulator power (P)

Description Controls the power level on the observe transmitter linear modulator. The fine power control is linear and spans 0 to `tpwr`.

Values 0 to 4095, where 4095 is maximum power. If `tpwrm` does not exist in the parameter table, a value of 4095 is assumed.

See also *NMR Spectroscopy User Guide*

Related

`config`	Determine current configuration and possibly change it (M)
`dpwrf`	First decoupler fine power (P)
`fattn`	Fine attenuator (P)

`trace` Mode for *n*-dimensional data display (P)

Description Sets the multidimensional data display mode.

Values `'f1'` displays the f_1 axis horizontally and allows f_1 traces to be displayed.

`'f2'` displays the f_2 axis horizontally and allows f2 traces to be displayed.

`'f3'` displays the f3 axis horizontally and allows f_3 traces to be displayed if the data set is 3D.

See also *NMR Spectroscopy User Guide*

traymax Sample changer tray slots (P)

Applicability Systems with an automatic sample changer.

Description Specifies the type of sample changer. It also can be used to disable the sample changer. The value is set using the Sample Changer label in the Spectrometer Configuration window.

Values 0 is setting for no sample changer present or, if a sample changer is attached, to disable the changer (None choice in the Spectrometer Configuration window).

9, 50, 100, 96, 48 are `traymax` values that indicate the number of sample slots for the corresponding sample changer (9 is for Carousel, 50 is for SMS/ASM 50 Sample, 100 is for SMS/ASM 100 Sample, 96 is for VAST, and 48 is for NMS, 768 for 768AS).

See also *VnmrJ Installation and Administration; VnmrJ Walkup*

Related `config` Display current configuration and possibly change it (M)

trfunc Translates screen co-ordinates

Syntax trfunc($x,$y):$xincm,$yincm

Applicability VnmrJ 3.1

Description `trfunc` translates screen co-ordinates to hertz or centimeters depending upon the `axis` parameter.

Examples call trfunc($x,$y):$xincm,$yincm

trfuncd Translates a screen distance

Syntax `trfuncd`

Applicability VnmrJ 3.1

Description `trfuncd` translates a screen distance into centimeters in a real image. It is only useful in axis='cc' (aspect ratio constrained) images.

Examples `trfuncd($screenlength):$imagelength`

troesy Set up parameters for TROESY pulse sequence (M)

Description Sets up parameters for the transverse cross-relaxation experiment in a rotating frame.

See also *NMR Spectroscopy User Guide*

trunc Truncate real numbers (O)

Description In MAGICAL programming, an operator that truncates real numbers.

Examples `$3 = trunc(3.6)`

See also *User Programming*

Related

acos	Find arc cosine of number (C)
asin	Find arc sine of number (C)
atan	Find arc tangent of a number (C)
cos	Find cosine value of an angle (C)
exp	Find exponential value (C)
ln	Find natural logarithm of a number (C)
tan	Find tangent value of an angle (C)
sqrt	Return square root of a real number (O)
typeof	Return identifier for argument type (O)

trtune Allows the user to view multiple tuning traces apparently simultaneously

Syntax `trtune`

Applicability VnmrJ 3.1

Description "`trtune`" allows the user to view multiple tuning traces apparently simultaneously. A tune sweep executes on `tn` nucleus (typically H1), then the `dn` nucleus, the `dn2`, and so on. A color key is displayed to the right and above the axis on the display. The # traces selection (the nf parameter) controls how many traces are performed, the maximum number of traces is the number of `rf` channels present. If probeConnect is present, it is used. If not, the channel order is '12345' if `tn` is highband, and '21345' otherwise.

There is only one vertical scale control. The traces may be adjusted by independent gain control (gain, gaind, gaind2, gaind3 etc.) which are defined in the prarameter set. The power may be adjusted independently as well (`tupwr`, `tupwrd`, `tupwrd2` etc.). It is preferable to keep power levels low, and adjust gain. Adjusting the display is easiest setting number of traces to 1 and autoscale. `Trtune` does not support shared RF channel nor does it support quadrature tuning.

tshift Adjust tau2 to current cursor position (M)

Applicability Systems with a solids module.

Description Adjusts `tau2` to make the current time cursor position the start of acquisition. As the time-domain cursor can move between points, this macro allows the accurate adjustment of `tau2` so as to start another acquisition exactly at the top of an echo.

See also *User Guide: Solid-State NMR*

tugain — Receiver gain used in tuning (P)

Description Used internally by the protune macro to set the receiver gain.

See also *NMR Spectroscopy User Guide*

Related

protune	Macro to start ProTune (M)
atune	ProTune Present (P)
mtune	Tune probe using swept-tune graphical display (M)
tchan	RF channel number used for tuning (P)
tunematch	Default match target, in percent of optimum (P)
tunesw	Width of the tuning sweep in Hz (P
tupwr	Transmitter power used in tuning (P)

tune — Assign a frequency to a channel for probe tuning (C)

Syntax (1) `tune(freq1,<freq2,freq3,freq4>)`
(2) `tune(chan1,freq1,<chan2,freq2,...>)`

Description Assigns a frequency to a channel when tuning the probe. The frequency assignment remains in effect (as a tune frequency) until the next su or go command is executed. Although only the first synthesizer is connected to the tuning system, the console is programmed to set this synthesizer to the desired frequency based on the channel shown on the CHAN readout on the TUNE INTERFACE unit.

The `tune` program has two formats. If syntax 1 is used, frequencies are assigned to channels based on the order of the arguments. The first argument is interpreted and assigned to the first (observe) channel, the second argument is assigned to the second (decoupler) channel. A third or fourth argument would be interpreted and assigned in a similar manner.

If syntax 2 is used, the arguments are entered in pairs, with the first argument specifying the rf channel and the next argument specifying the frequency.

`tune` selects the format based on the first argument. If the first argument is a name for an rf channel, syntax 2 is assumed; otherwise, syntax 1 is used.

Arguments `freq1`, `freq2`, `freq3`, and `freq4` specify the frequency of the rf channel as a value in MHz (e.g., `200` or `300`) or indirectly using the nucleus for tuning the probe (e.g., `'H1'` or `'C13'`). If a nucleus is entered, it must be found in the nucleus table. The frequency of any channel without an argument is unaffected. For example, `tune('H1','C13','N15')` sets the first channel to tune at the ^{1}H, the second channel at ^{13}C, and the third channel at ^{15}N. If a fourth channel is present, it is not affected. Entering `tune('H1','C13',200)` assigns the same frequencies for the first and second channels but the third channel tunes to 200 MHz, regardless of the proton frequency.

`chan1`, `chan2`, `chan3`, and `chan4` specify the channel directly:

- `'todev'` or `'ch1'` specify channel 1 (observe transmitter).
- `'dodev'` or `'ch2'` specify channel 2 (first decoupler).
- `'do2dev'` or `'ch3'` specify channel 3 (second decoupler).
- `'do3dev'` or `'ch4'` specify channel 4 (third decoupler).

Only one of these keywords is used per channel (do not enter the channel using just its number). If a channel does not have a keyword entered as an argument, that channel is not affected (e.g., `tune('ch4','P31')` selects the frequency corresponding to ^{31}P on the fourth channel, but leaves the first three channels unaffected).

Examples
```
tune('H1','C13','N15')
tune('H1','C13',200)
tune('ch4','P31')
```

See also *NMR Spectroscopy User Guide*

Related

`dfrq`	Transmitter frequency of first decoupler (P)
`dfrq2`	Transmitter frequency of second decoupler (P)
`dfrq3`	Transmitter frequency of third decoupler (P)
`go`	Submit experiment to acquisition (C)
`mtune`	Tune probe using swept-tune graphical display (M)
`qtune`	Tune probe using swept-tune graphical tool (C)
`sfrq`	Transmitter frequency of observe nucleus (P)
`spcfrq`	Display frequencies of rf channels (M)
`su`	Submit a setup experiment to acquisition (C)
`tune`	Assign frequencies (C)

tunehf — Tune both H1 and F19 on an HFX probe (M)

Syntax `tunehf<('x')>`

Description Tune both H1 and F19 on an HFX probe. Including the optional argument, `tunehf('x')` also tunes the low band channel to `dn` (`dfrq`).

Arguments `'x'`— low band channel to `dn` (`dfrq`)

See also *NMR Spectroscopy User Guide*

Related

`protune`	Macro to start ProTune (M)

tunematch Default match target, in percent of optimum (P)

Description The default match target, in percent of optimum. This local real parameter must be created. It is used as the match criterion in calls of the form `protune(599.96)`

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

protune	Macro to start ProTune (M)
create	Create new parameter in a parameter tree (C)
atune	ProTune Present (P)
mtune	Tune probe using swept-tune graphical display (M)
tchan	RF channel number used for tuning (P)
tugain	Receiver gain used in tuning (P)
tunesw	Width of the tuning sweep in Hz (P
tupwr	Transmitter power used in tuning (P)

tunemethod Method to use for tuning (P)

Applicability Liquids, VnmrJ Walkup, Automation

Description Specify probe tuning method. Methods are located in:
`$home/vnmrsys/tune/methods` for local user or
`/vnmr/tune/methods` for access by all users.
The method determines the nucleus to tune and how coarse or fine the probe is tuned as a percentage of the optimal `pw`.

Values `'lohi'` –tune low band to medium criterion then tune high band to medium criterion

`'<name>'` – user defined method.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

atune	ProTune Present (P)
protune	Macro to start ProTune (M)
wtune	Specify when to tune (P)

tuneResult Message indicating how well the tuning succeeded (P)

Description Message indicating how well the tuning succeeded. This local string parameter is created by ProTune and set to a string describing the result of the tuning. The first word of the message will be "ok" if tuning is successful, "failed" if it fails, and "Warning:" if tuning was not done but the experiment should proceed.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

protune	Macro to start ProTune (M)

tunerp — A pulse sequence for pulse tuning through the directional couplers in the VNMRJ display

Syntax `tunerp`

Applicability VnmrJ 3.1

Description A pulse sequence for pulse tuning through the directional couplers in the VNMRJ display. Tunerp is used for high-power pulsed tuning and for characterization of phase transient. It provides a phase-detected output of the pulse that can be displayed in phased or absolute value mode. Launch `Tunerp` with the `Fidscan` button on the `Shims` page in `Setup` tab.

To use `Tunerp` it is preferable to be able to measure the ratio of forward to reflected power. The standard directional couplers are wired to measure reflected power only. The arrow on the side of the coupler should point back toward the Front End to measure reflected power. To measure forward power reverse the coupler so that the arrow points toward the probe.

A second optional bidirectional coupler is available on some systems. With this coupler reverse the direction of the arrow by turning the knob on the top.

Setup

Load a calibrated data set or load `Settancpx` into a workspace that will not be used to acquire data. Convert the data set with `Tunerp`. Set `Tunerp` to obs the desired channel with the procedure below. It is helpful to set up `Tunerp` for each of 1-4 channels in the first 1-4 workspaces and join each of them when tuning is needed.

To tune a particular channel, enter its number (1-4) in Channel entry box on the Sequence page. Also set the particular channel as observe on the channels page and choose the nucleus.

For two-channel experiments, where channels 1 and 2 are used as obs and dec, it is simply necessary to enter the desired nucleus in the observe nucleus entry box. The correct channel will be selected automatically and the channel number will be displayed.

For three-channel experiments or any time channels 3 and 4 are involved, it is necessary to configure `probeConnect` and `preAmpConfig` before tuning. See the instructions below for configuration of these parameters. Enter one of the nuclei designated in `probeConnect` in the observe transmitter-nucleus entry box. The correct channel will be selected automatically and the channel number will be displayed

Set aTune and `tpwr` to appropriate values. The amplitude of the tuning pulse is determined by aTune (not aX90) and `tpwr`. One should tune routinely with about 25 to 50 Watts of power or less. Sometimes it is necessary to retune with the precise amplitude to be used in the experiment.

Pulse Tuning

Before pulse-tuning always rough-tune the probe with the mtune function.

Press the Tune button to set pwTune at 300 us and select a full FID display. Note that Tune sets a 5.0 MHz spectral width and the appropriate acquisition time. Enter Fidscan on the Shims page of the Setup tab and select magnitude mode only. Adjust the repetition rate with `d1` as desired. Alternatively type `av` and collect one-scan displays with acquire.

Switch to forward power and measure the pulse shape. It may be necessary to reduce reciever gain to avoid receiver overload. For high-power tuning it may be necessary to put attenuation between the coupler and the Front End.

Switch to reflected power and tune the probe to minimize the central component of the pulse. Characterize the forward/reflected ratio by recording two traces with the same value of vertical fid scale `vf`. Good tuning is a ratio of > 30/1.

Phase Transient

Minimization of phase transient on the proton channel is needed for multiple-pulse proton experiments such as `Hetcorlgcp2d`. Minimization of phase transient on the X channel is needed for Pisema2d and is desired for multiple-pulse X experiments such as C7inad2d.

Be sure the probe is tuned before measuring phase transient.

To characterize phase transient press the button labeled Transient to set a 10 us pulse whose rise and fall are clearly visible. Note that the Transient button sets a 5.0 MHz spectral width and the appropriate acquisition time. Enter Fidscan on the Shims page of the Setup tab and select real and imaginary modes. Adjust the repetition rate with `d1` as desired. Alternatively type `av` and collect one-scan displays with acquire.

Set the coupler for forward power. Collect a trace and phase it so that the real channel is 90 degrees out of phase and the imaginary channel is in phase. In this mode one will see zero amplitude with two transients of opposite phase at the beginning and ends of the pulse.

These transients are the phase transient and represent pulse amplitudes during the rise and fall times that are 90 degrees out of phase from the pulse.

Adjust probe tuning or cable lengths so as to minimize the amplitude of the two transients. Note that removal of phase transient with the probe tuning alone will detune the probe and increase reflected power. Generally one cannot achieve both good tuning and no transient by changing only the probe.

To remove phase transient by probe tuning adjust the Tune knob on the probe to move the tuning dip either up or down in frequency. Rephase the display and note whether the transients have gained or lost amplitude. Choose a value that minimizes the transient.

Phase transient can be removed permanently by adjusting the cable length between the probe and the directional coupler. Phase transient is a minimum for cable lengths that are multiples of 1/2 wavelength plus a constant. To find the correct length it is helpful to have a set of short cables and connectors and experiment with different lengths. The high-band channel can be adjusted with a set of elbow connectors. Once the correct length is found it is desirable to have a single permanent length made. Be sure that the probe remains tuned during this process. Note that one must have a different cable length for each different nucleus.

Three and Four Channel Experiments

To tune on channels 3 and 4 one must set `probeConnect` and `preAmpConfig`. These two parameters are "Global" strings than must be created manually by the system manager or user. As global parameters these strings apply to all workspaces in a user and do not affect other users. Note that these parameters are NOT created in the "Update User" function of the VNMRJ administrator interface or by the "makeuser" function.

`probeConnect` is a global string whose entries are the nuclei to be assigned to each channel. Create it with the command create `'probeConnect','string','global'`). Type display(`'probeConnect','global')` to verify its exisitence. Type display('`probeConnect`') alone to verify that a "current" version of probeConnect does not exist. The result should be negative.

Set [rpbeConnect equal to the nuclei for channels 1 to the number of channels, `numrfch`, in order, separated by spaces. For example:

`probeConnect` = 'H1 C13 F19 N15'

sets up a four channel spectrometer with an HFXY probe tuned to the indicated nuclei. Note that the first entry is always highband and the second always low band. On three-channel spectrometers the third entry must match the band of channel-three amp. On four channel spectrometers a second highband amp is always placed on channel 3 if it is present.

`preAmpConfig` is a global string whose entries indicate the receiver function attached to each channel. Create it with the command create('preAmpConfig','string','global'). Type display('preAmpConfig','global') to verify its exisitence. Type display('`preAmpConfig`') alone to verify that a "current" version of preAmpConfig does not exist. The result should be negative.

The characters of preAmpConfig can be "H" for highband, "L" for lowband and "X" for no preamp. The band of the preamp on a channel must match the band of the amplifier. A channel must have a preamp to be selected as the observe function.

An example for preAmpConfig is:

`preAmpConfig = 'HLHL'`

for the four-channel machine above.

`probeConnect` and `preAmpConfig` are present on the Channels page of all sequences. An output of "---" means that parameter does not exist. An Output of " " means that the parameter exists but has null value.

Parameter Groups

tune: Module: no

Sequence: tunerp.c

Description: Implements a directional-coupler pulse on a selected hardware channel for pulse tuning.

Parameters: Sequence Page

Arguments atune: the amplitude of the tune pulse.

chtune: the hardware channel to be tuned.

pwtune: the length of the tune pulse.

`tunesw` Width of the tuning sweep in Hz (P)

Description Sets the width of the tuning sweep in Hz and is set by the protune macro.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

`protune`	Macro to start ProTune (M)
`atune`	ProTune Present (P)
`mtune`	Tune probe using swept-tune graphical display (M)
`tchan`	RF channel number used for tuning (P)
`tugain`	Receiver gain used in tuning (P)
`tunematch`	Default match target, in percent of optimum (P)
`tupwr`	Transmitter power used in tuning (P)

`tupwr` Transmitter power used in tuning (P)

Description The transmitter power used in tuning. The `aptune` pulse sequence uses this to set the transmitter power. Set by the `protune` macro.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

`protune`	Macro to start ProTune (M)
`atune`	ProTune Present (P)
`mtune`	Tune probe using swept-tune graphical display (M)
`tchan`	RF channel number used for tuning (P)
`tugain`	Receiver gain used in tuning (P)
`tunematch`	Default match target, in percent of optimum (P)
`tunesw`	Width of the tuning sweep in Hz (P

`typeof` Return identifier for argument type (O)

Syntax `typeof`

Description In MAGICAL programming, an operator that returns an identifier (0 or 1) for the type (real or string) of an argument.

Examples `if typeof('$1') then $arg=1 else $arg=$1 endif`

See also *User Programming*

Related

`isreal`	Utility macro to determine a parameter type (M)
`isstring`	Utility macro to determine a parameter type (M)
`on`	Make a parameter active or test its state (C)
`size`	Return number of elements in an arrayed parameter (O)

U

ultra8	Selects the Ultra 8 shim configuration (M)
ultra18	Selects the Ultra 18 shim configuration (M)
undospins	Restore spin system as before last iterative run (M)
undosy	Restore original 1D NMR data from sub experiment (M)
undosy3D	Restores 2D DOSY data stored by the dosy macro in 3D DOSY(M)
unit	Define conversion units (C)
unixtime	Return marker for current time to a Magical variable
unlock	Remove inactive lock and join experiment (C)
updatepars	Update all parameter sets saved in a directory (M)
updateprobe	Update probe file (M)
updaterev	Update after installing new VnmrJ version (M)
updtgcoil	Update gradient coil (M)
updtparam	Update specified acquisition parameters (C)
usemark	Use "mark" output as deconvolution starting point (M)
userdir	VnmrJ user directory (P)
usergo	Experiment setup macro called by go, ga, and au (M)
userfixpar	Macro called by fixpar (M)

ultra8 selects the Ultra 8 shim configuration (M)

Syntax ultra8

Description The ultra8 macro selects the Ultra 8 shim configuration and selects an appropriate template for the dgs command and manual shim panel.

Administrator privilege is required to change the shim configuration. The shims are: z1c z2c x1 y1 xz yz xy x2y2.

Related ultra18 selects the Ultra 18 shim configuration (M)

ultra18 — Select 18 shim configuration for Ultra 18 shim power supply (M)

Syntax `ultra18`

Description Selects the 18 shim configuration for the Ultra 18 shim power supply and selects an appropriate template for the `dgs` command and manual shim panel. Administrator privilege is required to change the shim configuration.

The shims are: `z1 z1c z2 z2c z3c z4c x1 y1 xz yz xy x2y2 x3 y3 xz2 yz2 zxy zx2y2`

Related `ultra8` selects the Ultra 8 shim configuration (M)

undospins — Restore spin system as before last iterative run (M)

Description Returns the values of the line assignments and the chemical shifts and coupling constants existing before the last iterative adjustment with `spins('iterate')`, and then runs `spins`. The parameters are returned from the file `spini.inpar` and the transitions from the file `spini.savela` in the current experiment.

See also *NMR Spectroscopy User Guide*

Related `spins` Perform spin simulation calculation (C)

undosy — Restore original 1D NMR data from sub experiment (M)

Description Restores the 1D DOSY data stored by the `dosy` macro (if data exists) by recalling the data stored in the file `subexp/dosy2Ddisplay` in the current experiment. `undosy` and `redosy` enable easy switching between the 1D DOSY data (spectra as a function of `gzlvl1`) and the 2D DOSY display (signal as a function of frequency and diffusion coefficient).

See also *NMR Spectroscopy User Guide*

Related `dosy` Process DOSY experiments (M)
`redosy` Restore 2D DOSY display from subexperiment (M)

undosy3D

Syntax `undosy3D`

Applicability VnmrJ 3.1

Description undosy3D restores 2D DOSY data stored by the dosy macro (if they exist), recalling the data stored in the file subexp/original2d in the current experiment.

See also dosy

unit Define conversion units (C)

Syntax
```
unit<(suffix,label,m<,tree><,'mult'|'div'>   \
   ,b<,tree><,'add'|'sub'>)>
```

Description Defines a linear relationship that can be used to enter parameters with units. The unit is applied as a suffix to the numerical value (e.g., 10k, 100p). The definition of the linear relations follows the traditional y=mx+b equation, where x is the input value and y is the converted result.

Entering the unit command with no arguments displays all currently defined units. To remove a unit, define the unit with a 0 for the slope.

A convenient place to put unit commands for all users is in the bootup macro. Put private unit commands in a user's login macro.

Arguments suffix is a string identifying the name for the unit. The length of the string is limited to 12 characters.

label is a string for the name to be displayed when the axis parameter is set to the value of the suffix (if the suffix is only a single character). The length of the string is limited to 12 characters.

m is the slope of the linear relationship, defined either as a numerical value or as the name of a parameter. If a parameter name is used, it may be optionally followed with the parameter tree to use (argument tree) and by another optional keyword that specifies whether the parameter value should be a multiplier (keyword 'mult') or divisor (keyword 'div').

tree is the parameter tree to use (i.e.,'current', 'processed', 'global', or 'systemglobal'). The default tree is 'current'.

'mult' is a keyword that specifies that a parameter value used for the slope should be a multiplier. This is the default for the slope.

'div' is a keyword that specifies that a parameter value used for the slope should be a divisor.

b is the intercept of the linear relationship, defined either as a numerical value or as the name of a parameter. If a parameter name is used, it may be optionally followed with the parameter tree to use (argument tree) and by another optional keyword that specifies whether the parameter value should be added (keyword 'add') or subtracted (keyword 'sub').

'add' is a keyword that specifies that a parameter value used for the intercept should be a added. This is the default for the intercept.

'sub' is a keyword that specifies that a parameter value used for the intercept should be a subtract.

Examples `unit`
Displays all currently defined units

`unit('k','kHz',1000)`
`r1=10k` will set `r1` to `10000`

`unit('p','ppm','reffrq','processed')`
`r1=10p` will set `r1` to `10*reffrq`, where `reffrq` from processed tree

`unit('p','',0)`
`r1=10p` will set `r1` to `10` and give an error “`unknown unit p`”

`unit('F','degF',5/9,-32*5/9)`
`r1=212F` will set `r1` to `100` (degrees C)

`unit('C','degC',9/5,32)`
`r1=100C` will set `r1` to `212` (degrees F)

See also *NMR Spectroscopy User Guide, User Programming*

Related	`axis`	Axis label for displays and plots (P)
	`bootup`	Macro executed automatically when VnmrJ is activated (M)

unixtime — Return marker for current time to a Magical variable

Syntax `unixtime:r1,r2- Return marker for current time to a Magical variable`
`systemtime:r1,r2- synonym for unixtime`

Applicability VnmrJ 3.1

Description `unixtime` and systemtime are two names for the same function. They determine the current date and time as a system-dependent integer. The return value is in seconds. This value is usually defined as the elapsed time from an "`epoch`", which is often 1970. A second return value will give a microsecond value, for higher resolution.

The unixtime command helps time the execution of commands. It returns a marker representing the current time, in seconds. Call `unixtime` at the start and the end of a sequence of operation and then subtract the starting from the ending time to get the elapsed time.

`unixtime` accesses only the wall clock time, not the CPU time or any other statistic connected with the current process. The units for values returned are seconds and values should be accurate to within a few milliseconds.

Be aware that unixtime cannot time operations that run in background, for example, the `ft3d` command or go and its aliases.

The following Magical code fragment illustrates how you time something:

```
$t1 = 0

$t2 = 0

$t3 = 0

unixtime:$t1
```

```
ft2d
unixtime:$t2
$t3=$t2-$t1
write('line3','elapsed time for ft2d is %f secs',$t3
```

For more information, consult the UNIX manual entries time and get time of day.

unlock Remove inactive lock and join experiment (C)

Syntax `unlock(exp_number,'force')`

Description In attempting to join another experiment, the `jexp` command may abort claiming the experiment is locked. This feature prevents two users from processing the same experimental data at the same time, which could corrupt the data (a "user" can also be a background operation invoked by the same user, such as in `wexp` processing). This lock can be left behind if the program or the computer crashes.

The `unlock` command removes the lock if it is inactive and joins the unlocked experiment. The command will fail if the lock is still active (i.e., the process that made the lock is still executing) or if the lock was placed on the experiment by a remote host. The latter situation can only occur when one or more nodes are sharing the same file system (and experimental data).

Arguments `exp_number` is the number of the experiment from 1 to 9 to be unlocked.

`force` unlocks an experiment under all circumstances and joins the unlocked experiment.

Examples `unlock(3)`

See also *NMR Spectroscopy User Guide*

Related `jexp` Join existing experiment (C)

updatepars Update all parameter sets saved in a directory (M)

Syntax `updatepars(directory)`

Description Corrects saved parameter sets. Starting with VNMR version 4.2, all parameters, upper limit, lower limit, and step sizes have been tightened. Further additions were made in VNMR 4.3. `updatepars` searches a directory for parameter and FID files and corrects the `procpar` files found. This macro overwrites parameters in the current experiment. The corrections applied to the parameter sets are defined by the `parfix` macro. Because `updatepars` uses the current experiment to process the parameter sets, the experiment chosen for running `updatepars` should not contain a valuable data set.

Arguments `directory` is the name of the directory to be searched.

Examples `updatepars('myparlib')`
`updatepars('mydata')`

See also *NMR Spectroscopy User Guide*

Related `parfix` Update parameter sets (M)
`parversion` Version of parameter set (P)

updateprobe Update probe file (M)

Syntax `updateprobe(<probe|'tmplt'><,'system'>)`

Description Updates the current existing probe file or probe template.

Arguments `probe` is the probe parameter to update. The default is the current probe parameter value.

`'tmplt'` is a keyword to update the local probe template. The default is the current probe file.

`'system'` is a keyword to update the system template or probe file, providing you have write permission to the file. The default is to update the local template or probe file.

Examples
```
updateprobe
updateprobe('autosw')
updateprobe('autosw','system')
updateprobe('tmplt')
```

See also *NMR Spectroscopy User Guide*

Related `addparams` Add parameter to current probe file (M)
`getparam` Receive parameter from probe file (M)
`setparams` Write parameter to current probe file (M)

updaterev Update after installing new VnmrJ version (M)

Description Updates experiment parameters and the global file following installation of a new VNMR software version. `updaterev` is called by the `makeuser` command during the installation process.

See also *VnmrJ Installation and Administration*

updtgcoil Update gradient coil (M)

Applicability Systems with three-axis gradients.

Description Creates the `gcoil` parameter, if it does not exist, and sets it to the current value of the system gradient coil `sysgcoil`. `updtgcoil` only executes if gradients are configured in the system.

The `updtgcoil` macro is called when a new experiment is joined or new parameters are read into an experiment; however, it is only called at these times if the `gcoil` parameter exists. If `sysgcoil` is set to a gradient table name and if the values of `sysgcoil` and `gcoil` are different, a message is displayed in the Status window to let the user know that the gradient coil parameters have been updated.

`updtgcoil` can be called directly if the user wants to update the parameter set with the `gcoil` and gradient table parameters.

See also *NMR Spectroscopy User Guide; User Programming; VnmrJ Imaging NMR*

Related

`gcoil`	Read data from gradient calibration tables (P)
`sysgcoil`	System gradient coil (P)

`updtparam` Update specified acquisition parameters (C)

Description Enables interactive updating of specified acquisition parameters.

See also *SpinCAD*

Related

`psgupdateoff`	Prevent update of acquisition parameters (C)
`psgupdateon`	Enable update of acquisition parameters (C)

`usemark` Use “mark” output as deconvolution starting point (M)

Description In some cases it is not possible to produce a line list that is a suitable starting point for a deconvolution (e.g., lines may overlap so severely that a line list does not find them). In this case, or in any case, the results of a “mark” operation during a previous spectral display (`ds`) may be used to provide a starting point. If the “mark” has been made with a single cursor, the information in the file `mark1d.out` contains only a frequency and intensity, and the starting linewidth is taken from the parameter `slw`.

If the “mark” is made with two cursors, placed symmetrically about the center of each line at the half-height point, `mark1d.out` contains two frequencies and an intensity. In this case, the starting frequency is taken as the average of the two cursor positions; the starting linewidth is taken as their difference (thus allowing different starting linewidths for each line).

See also *NMR Spectroscopy User Guide*

Related

`ds`	Display a spectrum (C)
`slw`	Spin simulation linewidth (P)

userdir VnmrJ user directory (P)

Description Stores the full UNIX path of the directory that contains a user's private VnmrJ files. These include a user's private maclib, menulib, shims, psglib, experiments, etc. This parameter is initialized at bootup by the UNIX environmental variable vnmruser.

Values Typical value is /home/vnmr2/vnmrsys

See also *NMR Spectroscopy User Guide*

Related	curexp	Current experiment directory (P)
	systemdir	VnmrJ system directory (P)

usergo Experiment setup macro called by go, ga, and au (M)

Description Called by macros go, ga, or au before starting an experiment. The user typically creates usergo as a means to set up general experiment conditions.

See also *NMR Spectroscopy User Guide*

Related	au	Submit experiment to acquisition and process data (M)
	ga	Submit experiment to ac acquisition and FT the result (M)
	go	Submit experiment to acquisition (M)
	go_	Pulse sequence setup macro called by go, ga, and au (M)

userfixpar Macro called by fixpar (M)

Description Called by the macro fixpar to provide an easy mechanism to customize parameter sets.

See also *NMR Spectroscopy User Guide*

Related	fixpar	Correct parameter characteristics in experiment (M)

V

vast1d	Set up initial parameters for VAST experiments (M)
vastget	Selects and displays VAST spectra (M)
vastglue	Assemble 1D datasets into a 2D (or pseudo-2D) datasets (M)
vastglue2	Assemble 1D datasets into a 2D (or pseudo-2D) datasets (M)
vastgo	Turn off LC stop flow automation, start VAST automation (M)
vbg	Run VNMR processing in background (U)
vf	Vertical scale of FID (P)
vi	Edit text file with vi text editor (M)
vibradd	Display relative amplitudes of Cold Probe vibrations (M)
vjhelp	Display VnmrJ help (U)
vn	Start VNMR directly (U)
vnmr	Start VNMR in current windowing system (U)
vnmr2sc	VNMR to SpinCAD pulse sequence translator (M)
vnmr_accounting	Open Accounting window (U)
vnmremail	Utility to Send Files via Email
vnmrexit	Exit from the VNMR system (C)
vnmrj	Start VnmrJ (U)
vnmrjcmd()	Commands to invoke the GUI popup (C)
vnmrjOptions	Installer for passworded VnmrJ options (C)
vnmrplot	Plot files (U)
vnmrprint	Print text files (U)
vo	Vertical offset (P)
vp	Vertical position of spectrum (P)
vpaction	Set initial state for multiple viewports (M)
vpf	Current vertical position of FID (P)

vpfi	Current vertical position of imaginary FID (P)
vpset3def	Set the viewport state to three default viewports (M)
vpsetup	Set new viewports (M)
vs	Vertical scale (P)
vs2d	Vertical scale for 2D displays (P)
vsadj	Automatic vertical scale adjustment (M)
vsadj2	Automatic vertical scale adjustment by powers of 2 (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
vsadjh	Automatic vertical scale adjustment for ^{1}H spectra (M)
vsproj	Vertical scale for projections and traces (P)
vtairflow	Variable Temperature Air Flow (P)
vtairlimits	Variable Temperature Air Flow Limits (P)
vtc	Variable temperature cutoff point (P)
vtcomplvl	Variable temperature compensation for gradient shimming (P)
vttype	Variable temperature controller present (P)
vtwait	Variable temperature wait time (P)
vxr_unix	Convert VXR-style text files to UNIX format (M,U)

vast1d Set up initial parameters for VAST experiments (M)

Applicability Systems with VAST accessory.

Description Sets up initial VAST parameters from the `/vnmr/stdpar` directory or from the user's `stdpar` directory if the appropriate file exists there. Any changes made to the files in these directories are reflected in the setup. The file `/vnmr/stdpar/vast1d.par` contains the "default" parameters for VAST spectra and should be modified as needed to produce spectra under desirable conditions. After running `vast1d`, the solvent parameter can be set by choosing it from the list of solvents listed in `/vnmr/solvents`.

See also *NMR Spectroscopy User Guide*

vastget Selects and displays VAST spectra (M)

Applicability Systems with VAST accessory.

Syntax `vastget(<well>,<well>, ...)>`

Description — Selects and displays the spectra from any arbitrary well or wells using the well label(s) as arguments. the spectra are displayed in a `dss` stacked plot.

Arguments — `well` is the well label from which you want to select and display spectra. The wells are labeled [A->H][1-8].

Examples — `vastget('B6','B7','C11','G3')`

See also — *NMR Spectroscopy User Guide*

`vastglue` Assemble 1D datasets into a 2D (or pseudo-2D) datasets (M)

Applicability — Systems with the VAST accessory.

Syntax — `vastglue(<rack,<zone>)`

`vastglue(<glue order>,<plate>)`

Description — Used to artificially reconstruct a 2D datasets from a series of 1D data sets having similar filenames. It is crucial to ensure that the format of the file names of each of the 1D data sets is identical. `vastglue` reads in each 1D file, in succession, and adds it to the previous data, but in a 2D format. It assumes that file names are of the format obtained when using the default setting of `autoname` (`autoname=''`). If `autoname` has been redefined, use a macro like `vastglue2`. Save the resulting reconstructed 2D datasets in the normal manner using `svf`.

Arguments — `rack` is the rack number; the default is 1. If you enter a `rack` number, you must also enter a `zone` number.

`zone` is the zone number; the default is 1. If you want to specify a `zone` number, you must enter a `rack` number.

`glue order` is the specific glue order to be defined based on the order defined in a `plate_glue` file. If `glue order` is specified, you can provide a `plate` number as the second argument and used with the `glue order` argument.

See also — *NMR Spectroscopy User Guide*

Related

`autoname`	Prefix for automation data file (P)
`vastglue2`	Assemble related 1D datasets into a 2D (or pseudo-2D) datasets (M)

`vastglue2` Assemble 1D datasets into a 2D (or pseudo-2D) datasets (M)

Applicability — Systems with the VAST accessory

Syntax — `vastglue2<(number)>`

Description — Used to artificially reconstruct a 2D data set from a series of 1D datasets having similar filenames. It is crucial to ensure that the format of the file names of each of the 1D datasets is identical. `vastglue2` reads in each 1D file, in succession, and adds it to the previous data, but in a 2D format. It assumes that file names are of the format obtained using a nondefault setting of `autoname` (`autoname='filename_R%RACK:%_Z%ZONE:%_S%SAMPLE#:%_'`). This definition must be hard coded into the macro by the user. If `autoname` has not been redefined, use a macro like `vastglue`. Save the resulting reconstructed 2D data set in the normal manner using `svf`.

Arguments: `number` is used to specify that only spectra from 1 through `number` are to be glued. The default is to glue all the spectra stored in the current directory that have the proper file name format (from 1 through `arraydim`).

See also: *NMR Spectroscopy User Guide*

Related:

`autoname`	Prerix for automation data file (P)
`vastglue`	Assemble related 1D datasets into a 2D (or pseudo-2D) data set (M)

vastgo — Turn off LC stop flow automation, start VAST automation (M)

Applicability: Systems with the LC-NMR and VAST accessory

Description: Turns off LC stopped flow use of automation and starts VAST automation run.

vbg — Run VNMR processing in background (U)

Syntax: (From UNIX) `vbg exp_number command_string <prefix>`

Description: Enables user to perform VNMR tasks in the background. `vbg` (for “VNMR background processing”) must be run from within a UNIX shell, and *no* foreground or other background processes can be active in the designated experiment (e.g., if you are working in `exp2` in VNMR (in the foreground), you cannot execute background processing in `exp2` as well).

Foreground processing causes a lock file to be placed in the appropriate experiment. The file has a format such as `f.1268`, where `1268` indicates the process number in the process table (accessed in UNIX by entering the command `ps -e`). Background processing causes a lock file to be in the appropriate experiment as well. This file has a format such as `b.4356`, where `4356` indicates the process number. By displaying the files within an experiment, the user can readily determine whether any foreground or background processes are active in that experiment.

Arguments `exp_number` is the number of the experiment, from 1 to 9, in the user's directory in which the background processing is to take place.

`command_string` is the command string to be executed by VNMR in the background. Double quotes enclosing the string are mandatory (e.g., `"fn=4096 fn1=2048 wft2da"`).

`prefix` is a prefix to be added to the name of the log file, making the name `prefix_bgf.log`. The default name is `exp_number_bgf.log`, where `exp_number` is the experiment number. The log file is placed in the experiment in which the background processing takes place.

Examples (From UNIX) `vbg 1 "wft2da bc('f1')"`
(From UNIX) `vbg 3 "vsadj pl pscale pap page" plotlog`

See also *User Programming*

vf — Vertical scale of FID (P)

Description In normalized intensity (`nm`) mode, `vf` is the height of the largest FID. In absolute intensity (`ai`) mode, `vf` is a multiplier that is adjusted to produce a desired vertical scale, using the appearance on the display screen as a guide (full scale on the screen gives full scale on the plotter).

`vf` can be entered in the usual way or interactively controlled by clicking the middle mouse button in the graphics window during a FID display (click above the FID to increase `vf` or below the FID to decrease it).

Values 1e–6 to 1e9, in mm (in `nm` mode) or as a multiplier (in `ai` mode).

See also *NMR Spectroscopy User Guide*

Related

`ai`	Select absolute intensity mode (C)
`df`	Display a single FID (C)
`nm`	Select normalized intensity mode (C)
`sf`	Start of FID (P)
`wf`	Width of FID (P)

vi — Edit text file with vi text editor (M)

Syntax `vi(file)`

Description Invokes the UNIX text editor `vi` for editing the file name given. On the Sun workstation, a popup screen contains the editing window. On the GraphOn terminal, the main screen becomes the editing window. `vi` is a powerful text editor, but its user interface is limited: the mouse is not used, menus are not available, and status information is virtually nonexistent.

`vi` operates in three modes: the *command mode* (for moving the cursor and editing text), the *insert mode* (for inserting text into the file), and the *last line mode* (for special operations). Each mode is described below.

Command mode

`vi` starts up in the command mode. In this mode, user commands consist mostly of a single character, sometimes in combination with another character, or a number, or both. A number preceding a command typically defines how many times a command should be executed (e.g., `3dd` means delete three lines). The commands available include the following:

`G`	go to the start of the last line in the file
`3G`	go to the start of line 3
`0`	(zero) go to the start of the current line
`$`	go to the end of the current line
Return or `+`	go to start of next line
`-`	(hyphen) go to start of previous line
Ctrl-d	scroll down (forward) half a screen
Ctrl-f	scroll forward by a full screen
Ctrl-u	scroll up (back) half a screen
Ctrl-b	scroll back by a full screen
`/expression`	find next `expression` and jump to its first character
`?expression`	find previous `expression`, jump to its first character
`n`	find next `expression` (from the last search)
`N`	find previous `expression` (from the last search)
`dd`	delete one line and put it into the buffer
`3dd`	delete three lines and put them into the buffer
`dw`	delete word
`x`	erase one character forward (under cursor)
`X`	erase one character backwards (before cursor)
`3x`	erase three characters forward
`rcharacter`	erase character and replace with `character`
`ZZ`	write if necessary and quit *vi*
`.`	(period) repeat the last command
`u`	undo the last command
`J`	join the next line to the current line
`yy` or `Y`	yank one line and put into a buffer (called yank buffer)
`p`	put contents of yank buffer after the cursor
`P`	put contents of yank buffer before the cursor
`"aY`	yank line into buffer `a` (buffers `b` to `z` also available)

`"ap`	put contents of buffer a below current line
`"aP`	put contents of buffer a above current line

Because there is no command line, these commands do not show up on the screen but are *executed immediately* (without pressing the Return key).

Insert mode

In the insert mode, characters typed on the keyboard (except for the Esc key) show up in the text. The insert mode is entered by typing one of the following commands from the command mode:

`a text` Esc	append `text` after the current cursor position
`A text` Esc	append `text` to the end of current line
`i text` Esc	insert `text` before current cursor position
`cw word` Esc	change `word` from current cursor position to end
`2cw words` Esc	change two `words` from current cursor position to end
`o text` Esc	open line below current line and append `text`
`O text` Esc	open line above current line and append `text`

The only way to exit the insert mode is by pressing the Esc key, which leads back to the command mode. Unfortunately, there is no indication on the screen whether `vi` is in the command mode or in the insert mode. Inexperienced users often press the Esc key to make sure they are still in the command mode. The Esc key can also be used to avoid execution of commands that have been typed partially (e.g., the number has been typed, but not the last character).

You can insert special (normally nondisplayable) characters into the text if they are preceded by a Ctrl-v (e.g., entering Ctrl-v Ctrl-q is displayed in the text as `^Q`).

Changing selected occurrences

The following actions find one or more occurrences of a particular word and change it to another word:

- First, type `/word` and press Return, where / is a forward slash and `word` is word you want to change.
- Next, press `n` as necessary until you reach the occurrence of the `word` you want to change.
- Finally, type `cw newword` and press Esc, where `newword` is replacement word.
- To repeat for another occurrence of `word`, press `n` as necessary to scan forward, and then type `.` (a period) to repeat `cw newword` (or whatever was the last change)

Changing selected occurrences of an expression (one or more words) is similar. To change two words, for example, take the same actions as above but use the command `2cw` (or `c2w`) instead.

Last line mode

The last line mode is initiated with a colon; thereafter, commands such as the following can be used (press Return to execute these commands):

`:r filename`	read file named `filename` (insert in currently open file)
`:w`	write (save) file
`:w filename`	write under a new file named `filename`
`:e filename`	edit a different file named `filename`
`:q`	quit `vi` (only possible if file has been written back)
`:wq`	write back file (save changes) and quit `vi`
`:q!`	quit `vi` without saving changes

Exiting from `vi` is accomplished by using the `ZZ` command in the command mode, or with the `:q`, `:wq`, or `:q!` commands in the last line mode.

This description lists only a selection of the most important commands. For more information on `vi`, refer to UNIX books and manuals.

Examples `vi(userdir+'/psglib/apt.c')`
`vi(curexp+'/text')`

See also *User Programming*

Related

`edit`	Edit a file with user-selectable editor (M)
`paramvi`	Edit a parameter and its attributes with `vi` text editor (M)
`macrovi`	Edit a user macro with the `vi` text editor (C)
`menuvi`	Edit a menu with the `vi` text editor (M)
`textvi`	Edit text file of current experiment (M

vibradd Display relative amplitudes of Cold Probe vibrations (M)

Applicability Systems with Agilent, Inc. Cold Probes

Description Display the relative amplitudes of the vibrations reaching the probe. Requires a doped HOD sample.

vjhelp Display VnmrJ help (U)

Syntax `vjhelp file:///vnmr/jhelp/jhelp.html`

Description Displays the VnmrJ help in a Web browser.

vn — Start VNMR directly (U)

Syntax (From UNIX) `vn <-display Xserver> <-fn font> &`

Description Starts the VNMR application directly without checking the operating system and attempting to run the window manager.

Arguments `-display Xserver` specifies X server display (e.g., `hostname:0.0`). The default is the environment set by the `DISPLAY` variable.

`-fn font` specifies the size of the font displayed (e.g., `9x15`, `8x13`, or `7x13`). The default is the font set in the `.Xdefaults` file. Note that the size of the font affects the size of the VNMR window.

Examples
```
vn &
vn -display hostname:0.0 &
vn -font 8x13 &
```

See also *NMR Spectroscopy User Guide*

Related `vnmr` Start VNMR (U)

vnmr — Starts VnmrJ (U)

Applicability VnmrJ

Syntax `vnmr`

Description Starts the VnmrJ application

See also *NMR Spectroscopy User Guide*

Related `vnmrj` Start VnmrJ (U)

vnmr2sc — VNMR to SpinCAD pulse sequence translator (M)

Syntax `vnmr2sc<('sequence_name'<,rfchannels<,gradchannels>>)>`

Description Converts the pulse sequence pointed to by the `seqfil` parameter in the current VNMR parameter set from a C program into a SpinCAD pulse sequence. The conversion result is stored in the local `spincad/psglib` under the same name as the C pulse sequence (i.e., the name stored in the `seqfil` parameter), but without the `.c` extension.

`vnmr2sc` uses `dps` output to generate the SpinCAD code, i.e., the pulse sequence must be compiled and must be displayable with `dps`. Pulse sequences that do not compile with the `dps` option cannot be translated. For the same reason, `vnmr2sc` cannot translate features that do not show up in `dps`. This means that `go`-time decisions (such as flag-based C `if` constructs) will *not* show up in the translated SpinCAD sequence. In such cases, you have two options:

- Translate the sequence several times, once for each of the relevant flag settings. That is, generate several (simpler) SpinCAD pulse sequences from a single C sequence.
- Translate the sequence once (preferably with all options turned on), then manually insert the necessary `if` statements and other missing elements using SpinCAD.

Arguments `sequence_name` is an optional argument that permits the name of the resulting SpinCAD pulse sequence to be specified. By default, `vnmr2sc` creates a SpinCAD sequence with the name specified in the `seqfil` parameter (i.e., the SpinCAD sequence has the same name as the C pulse sequence). `sequence_name` is particularly useful if a C sequence is to be translated into multiple SpinCAD sequences; see the examples.

`rfchannels` is an optional numeric argument specifying the number of rf channels. Use it when you want the SpinCAD sequence to address more rf channels. By default, `vnmr2sc` determines the number of rf channels from the source sequence. You can only *increase* the number of rf channels. If you specify 0 rf channels, the number of rf channels is left unchanged.

`gradchannels` is a second optional numeric argument specifying the number of gradient channels or axes. Use it when you want to convert a nongradient sequence to a gradient sequence or when you want the SpinCAD sequence to address more gradient axes than the source sequence. By default, `vnmr2sc` determines the number of gradient axes from the source sequence. You can only *increase*, not decrease, the number of gradient axes.

Examples
```
vnmr2sc
setup('H1','CDCl3') hmqc null=0.2 vnmr2sc
null=0 mbond='y' vnmr2sc('hmbc')
vnmr2sc('gcosy',2,3)
nt=256 vnmr2sc
vnmr2sc(4,1)
vnmr2sc(0,1)
```

See also *SpinCAD Manual*

Related		
	`dps`	Display pulse sequence (C)
	`spincad`	Run SpinCAD program (C)

`vnmr_accounting` Open Accounting window (U)

Description Opens a window for creating and maintaining cost accounting data for groups of users on a spectrometer system. The program accommodates multiple rate schedules for spectrometer usage. A calendar tool can be used to define holidays for holiday rates. There is no limit on the number of rates that can be defined. Multiple printers can be selected.

Any user can view the accounting information (enter `cd /vnmr/bin` followed by `./vnmr_accounting`), but to update information, the user must have `root` privileges.

See also	*System Installation and Administration*	
Related	`operator`	Operator name (P)
	`operatorlogin`	Sets work space and parameters for the operator (M)

`vnmremail` Utility to Send Files via Email

Description	Sends a file to an email address. Files are sent after `uuencode`. Directories are converted into tar files or zip files and sent.
Syntax	`vnmremail(<'-m'>,filename,address)`
Examples	`vnmremail('myfile','nmr@agilent.com')`
Arguments	The `-m` option is used to concatenate the specified file to the body of the email.

`vnmrexit` Exit from the VNMR system (C)

Description: Exits from the VNMR system in a graceful manner by writing parameters and data to the disk, removing lock files, and restoring the terminal (if on a GraphOn). To provide flexibility when exiting VNMR, the macro `exit` calls `vnmrexit` to exit from VNMR.

CAUTION

When you exit from the VNMR user interface on your X display system, whether you are using an X terminal or a Sun computer, and whether you are using OpenWindows, CDE, or Motif, you must first exit from any copy of VNMR running on your system. Failure to do this can cause current parameter values and even current data to be lost.

`vnmrj` Start VnmrJ (U)

Applicability	VnmrJ	
Syntax	`vnmrj`	
Description	Starts the VnmrJ application	
See also	*NMR Spectroscopy User Guide; VnmrJ Walkup*	
Related	`vnmr`	Starts VnmrJ (U)

vnmrjcmd() Commands to invoke the GUI popup (C)

Syntax `vnmrjcmd('command1','command2',..., parametername)`

`vnmrjcmd('command1','command2',...<, callback>)`

Description The `vnmrjcmd()` commands are needed in order to invoke the GUI popup in which the user enters the parameters.

Note that `vnmrbg` and VnmrJ cannot be easily synchronized. When a macro invokes VnmrJ via `vnmrjcmd`, the VnmrJ thread runs independently and the macro continues on and takes action without otherwise having knowledge of VnmrJ. In order to have events associated with required parameters occur in the proper order, a callback strategy was devised. In simple terms, the vnmrj commands can have a `callback` string such that when the required parameters are established in VnmrJ, `vnmrbg` can be re- invoked - the foremost example of this is re-entering the '`go`' macro after the parameters are established in VnmrJ.

Examples Sends parameters one at a time to VnmrJ to be eventually displayed in an entry popup:

`vnmrjcmd('reqpar','warngui','set', 'real', parametername)`

`vnmrjcmd('reqpar','warngui','set', 'string', parametername)`

Display a GUI panel listing required parameters sent from `vnmrbg` in the previous '`set`' option above:

`vnmrjcmd('reqpar','warngui','show')`

`vnmrjcmd('reqpar','warngui','show', callback)`

The callback is a command string to be sent back to `vnmrbg`, if needed. See the `reqpartest` macro source code for examples of how to use callback.

See also *VnmrJ User Programing*

Related		
	`go`	Submit experiment to acquisition (M)
	`reqpartest`	Tests whether required parameters are set (M)

vnmrjOptions Installer for passworded VnmrJ options (C)

Applicability VnmrJ 3.2

Description VnmrJ passworded options can be installed after the VnmrJ software in installed from the distribution media. This tool provides the mechanism to specify the passwords and install the options. If you run this tool and do not have permission to write to the /vnmr system directory, it will show you what options are currently loaded.

vnmrplot Plot files (U)

Syntax (From UNIX) `vnmrplot <file>`

Description A UNIX command that plots files from inside VNMR commands. To plot a file, you should use the page command, which uses `vnmrplot` internally.

Arguments `file` is the name of the file to be plotted.

See also *NMR Spectroscopy User Guide*

Related `vnmrprint` Print text files (U)

vnmrprint Print text files (U)

Syntax (From UNIX) `vnmrprint printfile <printcap>`
`<printer_type <clear|file>>`

Description A UNIX command installed as part of the VNMR system to print text files. The `printon` and `printoff` commands use `vnmrprint` to print files. `vnmrprint` can also be used to delete a print file or save a print file to a different name.

Arguments `printfile` is the name of the text file to be printed.

`printcap` is a UNIX printcap entry (e.g. `LaserJet_300`) for the printer to print the text file. The default is the printer selected by the `-p` option of the UNIX `lp` command.

`printer_type` is the type of printer from the list of VNMR printers (e.g., `LaserJet_300`). `printer_type` is required as an argument when it is desired to clear the printer file or save the printer file to another name.

`clear` is a keyword to delete the current print file. Deleting this file also requires that the `printfile`, `printcap`, and `printer_type` arguments be entered so that `clear` is the fourth argument.

`file` is the name of the file to use in saving the `printfile`. If a file with the name specified already exists, it is overwritten. Saving the file also requires that the `printfile`, `printcap`, and `printer_type` arguments be entered so that `file` is the fourth argument.

Examples
```
vnmrprint /vnmr/psglib/tocsy.c LaserJet_300
vnmrprint myfile LaserJet_300 LaserJet_300 clear
vnmrprint myfile ps PS_AR yourfile
```

See also *NMR Spectroscopy User Guide*

Related
`printoff` Stop sending text to printer and start print operation (C)
`printon` Direct text output to printer (C)
`vnmrplot` Plot files (U)

vo Vertical offset (P)

Description Sets the vertical offset, for 1D data sets, of the each spectrum in a *stacked display* with respect to the previous spectrum. The parameter ho sets the horizontal offset. For a “left-to-right” presentation, ho is typically negative; for a “bottom-to-top” presentation, vo is positive.

For 2D data sets, the parameter wc2 sets the distance between the first and last trace and the *vo* parameter is inactive.

Values Number, in mm.

See also *NMR Spectroscopy User Guide*

Related

ho	Horizontal offset (P)
wc2	Width of chart in second direction (P)

vp Vertical position of spectrum (P)

Description Contains vertical position of spectrum with respect to the bottom of the display or plotter.

Values –200 to +200, in mm.

See also *NMR Spectroscopy User Guide*

Related

vpf	Current vertical position of FID (P)
vpfi	Current vertical position of imaginary FID (P)

vpaction Set initial state for multiple viewports (M)

Applicability *VnmrJ Walkup*

Description Sets the initial state for multiple viewports. Used by the viewport editor dialog under **Edit -> Viewports**.

See also *User Programming*

Related

jcurwin	Work space numbers of all viewports (P)
jviewportlabel	Work space labels for all viewport buttons (P)
jviewports	Viewport layout (P)

vpf Current vertical position of FID (P)

Description Contains the current vertical position of an FID. To create this parameter and the other FID display parameters axisf, crf, deltaf, dotflag, and vpfi (if the parameter set is older and lacks these parameters), enter addpar('fid').

Values Number, in mm. If `vpf=0`, the FID is positioned in the middle of the screen.

See also *NMR Spectroscopy User Guide*

Related

`addpar`	Add selected parameters to the current experiment (M)
`axisf`	Axis label for FID displays and plots (P)
`crf`	Current time-domain cursor position (P)
`deltaf`	Difference of two time-domain cursors (P)
`dotflag`	Display FID as connected dots (P)
`vp`	Vertical position of spectrum (P)
`vpfi`	Current vertical position of imaginary FID (P)

vpfi Current vertical position of imaginary FID (P)

Description Contains the current vertical position of the imaginary part of an FID. To create this parameter and the other FID display parameters `axisf`, `crf`, `deltaf`, `dotflag`, and `vpf` (if the parameter set is older and lacks these parameters), enter `addpar('fid')`.

Values Number, in mm. In `vpfi`=0, the imaginary part is positioned in the middle of the screen.

See also *NMR Spectroscopy User Guide*

Related

`addpar`	Add selected parameters to the current experiment (M)
`axisf`	Axis label for FID displays and plots (P)
`crf`	Current time-domain cursor position (P)
`deltaf`	Difference of two time-domain cursors (P
`dotflag`	Display FID as connected dots (P)
`vp`	Vertical position of spectrum (P)
`vpf`	Current vertical position of FID (P)

vpset3def Set the viewport state to three default viewports (M)

Description Sets the number of viewports to three, and resets the viewport button labels.

See also *User Programming*

Related

`jcurwin`	Work space numbers of all viewports (P)
`jviewportlabel`	Work space labels for all viewport buttons (P)
`jviewports`	Viewport layout (P)

vpsetup Set new viewports (M)

Description Sets the viewports from the selections made in the viewport editor dialog. For each viewport, it checks the work space number to join, then joins the appropriate work space.

See also *User Programming*

Related

jcurwin	Work space numbers of all viewports (P)
jviewportlabel	Work space labels for all viewport buttons (P)
jviewports	Viewport layout (P)

vs Vertical scale (P)

Description In normalized (nm) mode, vs is the height of the largest peak in the spectrum. In absolute intensity (ai) mode, vs is a multiplier that is adjusted to produce a desired vertical scale, using the appearance on the display screen as a guide (full scale on the screen gives full scale on the plotter). vs can be entered in the usual way or interactively controlled by clicking the middle mouse button.

Values 1e–6 to 1e9, in mm (in nm mode) or as a multiplier (in ai mode).

See also *NMR Spectroscopy User Guide*

Related

ai	Select absolute intensity mode (C)
isadj	Adjust integral scale (M)
nm	Select normalized intensity mode (C)
thadj	Adjust threshold for peak printout (M)
vsadj	Automatic vertical scale adjustment (M)
vsadj2	Automatic vertical scale adjustment by powers of two (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
vsadjh	Automatic vertical scale adjustment for ^{1}H spectra (M)

vs2d Vertical scale for 2D displays (P)

Description Sets a multiplier for 2D spectra and images that is adjusted to produce a desired vertical scale for display or plotting. vs2d takes the place of vs for 2D data display and can be adjusted by explicitly setting it to a value or by clicking the middle mouse button when pointing to a point on a 2D display. If vs2d does not exist, it can be created by running par2d.

See also *NMR Spectroscopy User Guide*

Related

par2d	Create 2D acquisition, processing, and display parameters (M)
vs	Select vertical scale (C)
vsproj	Adjust vertical scale for projections and traces (M)

vsadj Automatic vertical scale adjustment (M)

Syntax `vsadj<(height)>`

Description Automatically sets the vertical scale vs in the absolute intensity (ai) mode so that the largest peak is at the requested height.

Arguments `height` is the desired height, in mm, of the largest signal in the displayed portion of the spectrum. The default is `0.9*(wc2max-vp-sc2)`.

Examples
```
vsadj
vsadj(100)
```

See also *NMR Spectroscopy User Guide*

Related

ai	Select absolute intensity mode (C)
isadj	Adjust integral scale (M)
thadj	Adjust threshold for peak printout (M)
vs	Vertical scale (P)
vsadj2	Automatic vertical scale adjustment by powers of two (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
vsadjh	Automatic vertical scale adjustment for ^{1}H spectra (M)
wc2max	Maximum width of chart in second direction (P)

vsadj2 Automatic vertical scale adjustment by powers of 2 (M)

Syntax `vsadj2<(height)>:scaling_factor`

Description Adjusts the vertical scale by powers of two as required for expansion plots (see aexppl for more information).

Arguments `height` is desired height of largest (or largest relevant) signal in displayed portion of the spectrum. The default is `0.9*(wc2max-vp-sc2)`.

`scaling_factor` returns to the calling macro the ratio of the new compared to the old value of vs.

Examples
```
vsadj2
vsadj2(50):r1
```

See also *NMR Spectroscopy User Guide*

Related

aexppl	Automatic expansions plot (M)
isadj	Adjust integral scale (M)
sc2	Start of chart in second direction (P)
thadj	Adjust threshold for peak printout (M)
vp	Vertical position of spectrum (P)
vs	Vertical Scale (P)
vsadj	Automatic vertical scale adjustment (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)
vsadjh	Automatic vertical scale adjustment for H1 spectra (M)
wc2max	Maximum width of chart in second direction (P)

vsadjc Automatic vertical scale adjustment for 13C spectra (M)

Syntax `vsadjc<(height)>`

Description Functionally the same as the macro vsadj, except excludes solvent and TMS signals from the carbon spectra for the adjustment of vs.

Arguments `height` is desired height of largest (or largest relevant) signal in displayed portion of the spectrum. The default is `0.9*(wc2max-vp-sc2)`.

Examples
```
vsadjc
vsadjc(wc2max-sc2-wc2-5)
```

See also *NMR Spectroscopy User Guide*

Related

isadj	Adjust integral scale (M)
thadj	Adjust threshold for peak printout (M)
vs	Vertical Scale (P)
vsadj	Automatic vertical scale adjustment (M)
vsadj2	Automatic vertical scale adjustment by powers of two (M)
vsadjh	Automatic vertical scale adjustment for H1 spectra (M)

vsadjh Automatic vertical scale adjustment for ^{1}H spectra (M)

Syntax `vsadjh<(height<,do_not_ignore_solvent>)>`

Description Works as the same as the macro vsadj, except disregards solvent and TMS signals from proton spectra and, if from the remaining spectrum the highest line is more than three times as high as the second highest line, the spectrum is scaled to this second highest signal (otherwise the highest signal is taken as relevant).

Arguments `height` is desired height of largest (or largest relevant) signal in displayed portion of the spectrum. If `height` is 0 or a negative value, it defaults to `0.9*(wc2max-vp-sc2)`, which is also the default with no arguments.

`do_not_ignore_solvent` is any second argument. If present, it signals `vsadjh` to not ignore the solvent line and regard the solvent line as normal signal (i.e, only exclude the TMS line). This argument was added for the situation where frequently there are high “real” signals at the position of the solvent line. Such signals could otherwise be regarded as solvent line and would then be ignored. This could then lead to overscaling in the result.

Examples
```
vsadjh
vsadjh(0.7*wc2max)
```

See also *NMR Spectroscopy User Guide*

Related

isadj	Adjust integral scale (M)
sc2	Start of chart in second direction (P)
thadj	Adjust threshold for peak printout (M)
vs	Vertical scale (P)

vsadj	Automatic vertical scale adjustment (M)
vsadj2	Automatic vertical scale adjustment by powers of two (M)
vsadjc	Automatic vertical scale adjustment for ^{13}C spectra (M)

vsproj Vertical scale for projections and traces (P)

Description Sets a multiplier that is adjusted to produce a desired vertical scale for projections or traces of 2D data sets. vsproj can be explicitly adjusted by setting it to a value or by clicking the middle mouse button when pointing at the projection or trace. When interactively adjusting the scale with the mouse, the higher the pointer is in the trace display, the larger the vertical scale. If the parameter does not exist, it can be created by running the par2d macro.

See also *NMR Spectroscopy User Guide*

Related

par2d	Create 2D acquisition, processing, and display parameters (M)
vs	Select vertical scale (C)
vs2d	Adjust vertical scale for 2D displays (M)

vtairflow Variable Temperature Air Flow (P)

Description This global parameter sets the VT air flow, in l/min. The adjustment is coarse, +/- 1 l/min. If there is not enough air flow available it may not reach the requested value.

Values 0 - 25

Related

pin	Pneumatics router interlock (P)
vtairlimits	Variable temperature air flow limits (P)

vtairlimits Variable Temperature Air Flow Limits (P)

Description This global parameter determines the range of safe VT air flow, as indicated by the LEDs on the flow meter. It sets the LEDs on the air flow meter, upper and lower LEDs are orange, in between are green. As long as the ball in the air flow meter is next to a green LED the air flow is considered safe. If the air flow drops or increases such that the ball is next to an orange LED, the pneumatics box will turn the VT Controller off and notify the experiment, provided the switch is in the 'run' position. A bit value of 1 sets an unsafe orange state, a bit value of 0 sets a safe green state.
To create the parameter:

```
create('vtairlimits','integer','global')
setlimit('vtairlimits',1023,0,1,'global')
```

Examples a value of 775 or 0x307 will set the two lower and the three upper LEDs (orange) and clear the remaining 5 in between (green). Note that the upper bits determine the lower LEDs. If the parameter does not exist the value defaults to 0x307 for liquids; 0x200 for solids.

Values 0 - 1023

Related
`pin` Pneumatics router interlock (P)
`tin` Temperature interlock (P)
`vtairflow` Variable temperature air flow (P)

vtc Variable temperature cutoff point (P)

Applicability Systems with a variable temperature (VT) module.

Description Sets a VT cutoff point. Above this temperature, VT air flows straight into the probe, past the heater, then past the sample. Below this temperature, air goes first through the heat exchange bucket, for cooling by the heat exchange fluid, and then into the probe and past the heater.

Values 0 to 50, in degrees celsius. `vtc` is typically set 5°C higher than the supply gas used for VT regulation.

See also *NMR Spectroscopy User Guide*

Related
`temp` Sample temperature (P)
`tin` Temperature interlock (P)

vtcomplvl Variable temperature compensation for gradient shimming (P)

Description Specifies the level of VT compensation used by gradient shimming.

Values
0, disable VT compensation.
1, enable VT compensation
2, enable VT compensation with extra gradient dephasing.

Related
`gmapz` Get parameters and files for gmapz pulse sequence (M)
`gmapsys` Run gradient autoshimming, set parameters, map shims (M)
`gzsize` Number of z-axis shims used by gradient shimming (P)
`temp` Sample temperature (P)
`vttype` Variable temperature controller present (P)

vttype Variable temperature controller present (P)

Description In the Spectrometer Configuration window, this parameter specifies whether a variable temperature (VT) controller is present or not on the system. The value is set using the VT Controller label in the Spectrometer Configuration window.

When entered from command line in VNMR, control of the variable temperature (VT) controller from the current experiment is either engaged (*vttype=2*) or disengaged *(vttype=0)*. The current state of the variable temperature (VT) controller is not changed when *vttype* is set in the command window.

The variable temperature (VT) controller setting in Spectrometer Configuration is not affected by entering *vttype* on the command line.

Values 2 is setting for VT controller (Present choice in Spectrometer Configuration window).

0 is setting for no VT controller (Not Present choice in Spectrometer Configuration window).

Examples If *temp='some temperature'* while *vttype=2* and *vttype* is then changed to *vttype=0* on the command line, the variable temperature (VT) controller will continue regulate the sample at the value set by *temp*. While *vttype=0* changes to *temp* will have no effect.

See also *VnmrJ Installation and Administration*; *NMR Spectroscopy User Guide*

Related

`config`	Display current configuration and possibly change values (M)
`masvt`	Type of variable temperature system (P)

vtwait Variable temperature wait time (P)

Applicability Systems with a variable temperature (VT) module.

Description Sets a time for establishing temperature regulation. If temperature interlock `tin` is set and regulation is not established after the time set by `vtwait`, VNMR displays the message “`VT FAILURE`” and aborts the experiment.

Values Number, in seconds, A typical value is 180 seconds.

See also *NMR Spectroscopy User Guide*

Related

`pad`	Preacquisition delay (P)
`tin`	Temperature interlock (P)

vxr_unix Convert VXR-style text files to UNIX format (M, U)

Syntax (From VNMR) `vxr_unix(VXR_file<,UNIX_file>)`
(From UNIX) `vxr_unix VXR_file UNIX_file`

Description Converts a VXR-style text file (from a Gemini, VXR, or XL system) to the UNIX format.

Arguments `VXR_file` is the name of the input file, which must be a text file.

`UNIX_file` is the name of the output file after conversion. The names of the input and output files must be different.

Examples (From VNMR) `vxr_unix('oldtextfile','newtextfile')`
(From UNIX) `vxr_unix oldtextfile newtextfile`

See also *NMR Spectroscopy User Guide*

Related `convert` Convert data set from a VXR-style system (C,U)
`decomp` Decompose a VXR-style directory (C)

W

w	Who is using system (C)
walkup	Walkup automation (M)
walkupQ_runtime	Macro to Control Study Queue
waltz	WALTZ decoupling present (P)
warmprobe	Tells the system a warm probe is present
wbs	Specify action when bs transients accumulate (C)
wbs	When block size (P)
wc	Counts Words in a String
wc2	Width of chart in second direction (P)
wcmax	Maximum width of chart (P)
wc2max	Maximum width of chart in second direction (P)
wdone	Specify action when experiment is done (C)
wdone	Specify action when experiment is done (P)
wds	
werr	Specify action when error occurs (C)
werr	When error (P)
wet	Flag to turn on or off wet solvent suppression ((P)
Wet1d	Set up parameters for wet ^{1}H experiment (M)
wetdqcosy	Set up parameters for a WETDQCOSY pulse sequence (M)
wetgcosy	Set up parameters for a WETGCOSY pulse sequence (M)
wetghmqcps	Set up parameters for a WETGHMQCPS pulse sequence (M)
wetghsqc	Set up parameters for a WETGHSQC pulse sequence (M)
wetgmqcosy	Set up parameters for a WETGHSQC pulse sequence (M)
wetit	Set up and create pulse shapes for Wet1d experiment (M)
wetnoesy	Set up parameters for a WETNOESY pulse sequence (M)

wetpeaks	Number of peaks for wet solvent suppression (P)
wetpwxcal	Set up parameters for a WETPWXCAL pulse sequence (M)
wettntocsy	Set up parameters for a WETTNTOCSY pulse sequence (M)
wetshape	Shape for pwwet pulses (P)
wexp	Specify action when experiment completes (C)
wexp	When experiment completes (P)
wf	Width of FID (P)
wf1	Width of interferogram in 1st indirectly detected dimension (P)
wf2	Width of interferogram in 2nd indirectly detected dimension (P)
wfgtest	Waveform generator test (M)
wft	Weight and Fourier transform 1D data (C)
wft1d	Weight and Fourier transform f_2 for 2D data (C)
wft1da	Weight and Fourier transform phase-sensitive data (M)
wft1dac	Combine arrayed 2D FID matrices (M)
wft2d	Weight and Fourier transform 2D data (C)
wft2da	Weight and Fourier transform phase-sensitive data (M)
wft2dac	Combine arrayed 2D FID matrices (M)
wftt3	Process f_3 dimension during 3D acquisition (M)
which	Display which command or macro is used (M)
wnt	Specify action when nt transients accumulate (C)
wnt	When number of transients (P)
wp	Width of plot in directly detected dimension (P)
wp1	Width of plot in 1st indirectly detected dimension (P)
wp2	Width of plot in 2nd indirectly detected dimension (P)
write	Write formatted text to a device (C)
writefid	Write numeric text file using a FID element (C)
writejxy	Create x,y ascii file from phasefile for JCAMP-DX conversion (M)
writeparam	Write one of more parameters to a file (C)
writespectrum	Write a spectrum to a binary file (C)
writetrace	Create ascii file from phasefile (f1 or f2) trace (M)
writexy	Create x,y ascii file from phasefile (f1 or f2) trace (M)
wrtp	Command string executed after rtp command (P)

wsram	Send hardware configuration to acquisition console (C)
wstart	
wshim	Conditions when shimming is performed (P)
wtfile	User-defined weighting in directly detected dimension (P)
wtfile1	User-defined weighting in 1st indirectly detected dimension (P)
wtfile2	User-defined weighting in 2nd indirectly detected dimension (P)
wtgen	Compile user-written weighting functions (M,U)
wti	Interactive weighting (C)
wtia	Interactive weighting for 2D absorptive data (M)
wtune	Specify when to tune (P)
wtunedone	What to do after ProTune tuning is done (P)
wysiwyg	Set plot display or full display (P)

w — Who is using system (C)

Description: Displays information about users currently on the system. It functions like the UNIX command of the same name.

See also: *User Programming*

walkup — Walkup automation (M)

Description: Enables using sample changers for continuous “walk-up” operation. Click on Utilities -> New automation run to run this macro from the VnmrJ Walkup interface. The macro creates a new automation directory each day with the name `auto_yyyy.mm.dd`, where `yyyy` is the year, `dd` is the day of the month, and `mm` is the month (e.g., `auto_20040601`). The automation directory is saved in a directory specified by the global parameter globalauto. `walkup` creates the directory `globalauto` and the parameter globalauto, and then sets the globalauto parameter.

See also: *VnmrJ Walkup*

Related:

enter	Enter sample information for automation run (M,U)
globalauto	Automation directory name (P)

walkupQ_runtime Macro to Control Study Queue

Syntax `""" walkupQ_runtime - this is typically used at runtime by ""`

```
"" CMD protocols to modify the queue as appropriate ""
"" Usage: walkupQ_runtime(keyword1,experiment/node,keyword2,..) ""
""   keyword1 = add / delete /customize ""
""    arg2 = '' is interpreted as ALL experiments in the queue ""
"" arg2='' and keyword1='add' is invalid ""
""   keyword2 = next / last / all / night / '' / node    ""
"" keyword2='node' interprets arg2 as nodename  ""
""   else arg2 is experimentname  ""
"" keyword2='node' and arg2='' is invalid combination ""
"" keyword2='all' and keyword1='add' is invalid  ""
"" keyword2='' is same as keyword2='next'   ""
"" keyword2='night' and arg2='' include an implicit 'all' ""
""    4th argument is required for customize option
```

Description this is typically used at runtime by ""

"" CMD protocols to modify the queue as appropriate

Arguments """" Number of Arguments: 3 ""

"" walkupQ_runtime('add','gHSQCAD','next') ""

"" Adds gHSQCAD as next experiment in the queue ""

"" walkupQ_runtime('add','gHSQCAD','gCOSY_02') ""

"" Add gHSQCAD after gCOSY_02 node ""

"" walkupQ_runtime('add','gHSQCAD','last') ""

"" Add gHSQCAD as last experiment in the queue ""

"" walkupQ_runtime('add','gHSQCAD','night') ""

"" Add gHSQCAD to the night queue ""

""***
*******************""

"" Number of Arguments: 3   ""

""   walkupQ_runtime('delete','','next')   ""

"" Delete the next experiment in the queue   ""

""   walkupQ_runtime('delete','','last')   ""

"" Delete the last experiment in the queue   ""

""   walkupQ_runtime('delete','','all')   ""

"" Delete all pending experiments in the queue ""

""   walkupQ_runtime('delete','HSQCAD','next or last or all') ""

"" Delete next(last or all) HSQCAD experiments   ""

""   walkupQ_runtime('delete','gHSQCAD_02','node') ""

"" Delete gHSQCAD_02 in the queue   ""

""***
*******************""

"" Number of Arguments: 4   ""

"" walkupQ_runtime('customize','','keyword','nt=32') ""

""   keyword='next' or 'last' or 'all'   ""

""   Set `nt`=32 for the next/last/all experiments   ""

"" walkupQ_runtime('customize','HSQCAD','keyword','`nt`=32') ""

"" keyword='next' or 'last' or 'all'   ""

"" Set `nt`=32 for the next/last/all HSQCAD experiments ""

"" walkupQ_runtime('customize','gHSQCAD_02','node','`nt`=32') ""

"" Set `nt`=32 for the gHSQCAD_02 experiment in the queue ""

"

`waltz` WALTZ decoupling present (P)

Description Sets whether system is equipped for WALTZ decoupling. The value is changed by normal parameter entry rather than using the Spectrometer Configuration window.

Values 'n' sets WALTZ decoupling not present.

'y' sets WALTZ decoupling present.

See also *VnmrJ Installation and Administration*

warmprobe Tells the system a warm probe is present

Applicability VnmrJ 3.1

Description If a C13 observe coldprobe is being used, the value of rof2 should not be less than 350 usec. The coldprobe macro tells the system that a coldprobe is present so that the rof2 rule is enforced.
The warmprobe macro tells the system that a warm probe is present so that the rof2 rule is not enforced.

Related coldprobe Tells the system a coldprobe is present

wbs Specify action when bs transients accumulate (C)

Syntax wbs(string)

Description Specifies what action to take when bs transients accumulate. The *command* wbs sets the corresponding *parameter* wbs. Using the command, rather than setting the parameter value explicitly, notifies the acquisition process that the associated parameter value has changed. Thus, the desired operation can be effected even if the experiment has already started.

Arguments string is a string argument containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. If single quotes are required *within* the text string, place a backslash character before each of the interior single quotes (\'). Maximum length of the string is 256 characters. To turn off wbs processing, enter wbs(''), where the argument is two single quotes with no space between.

Syntax
```
wbs('dg wft')
wbs('mf(3)')
wbs('')
```

See also *NMR Spectroscopy User Guide*

Related

bs	Block size (P)
makefid	Make a FID element using numeric text input (C)
phfid	Zero-order phasing constant for np FID (P)
wbs	When block size (P)
werr	Specify action when error occurs (C)
wexp	Specify action when experiment completes (C)
wnt	Specify action when nt transients accumulate (C)

wbs When block size (P)

Description Invokes an action to occur automatically after each bs block of transients is completed. For example, `wbs='wft'` results in an automatic weighting and Fourier transformation after each bs transients. To specify no wbs processing, set wbs to the null string. If the acquisition has already started, the wbs *command* must be used to change this parameter.

Values Command, macro, or null string (`wbs=''`, where the value is given by two single quotes with no space between them).

See also *NMR Spectroscopy User Guide*

Related

bs	Block size (P)
wbs	Specify action when bs transients accumulate (C)

wc Counts Words in a String

Syntax `wc(string)`

Description Utility to emulate the "`wc -w`" command in Unix. Called on a string variable, it returns the number of words in the string.

Examples `wc('textfile'):r1`

wc2 Width of chart in second direction (P)

Description Specifies width of chart (plotting or printing area) along the second axis (or *y* axis) of a 2D contour plot or 2D "stacked display." For plots made in the cutoff mode, wc2 specifies the width of the plotted area along the *y*-axis.

Values Width, in mm.

See also *NMR Spectroscopy User Guide*

Related

cutoff	Data truncation limit (P)
ho	Horizontal offset (P)
sc2	Start of chart in second direction (P)
wcmax	Maximum width of chart (P)
wc2max	Maximum width of chart in second direction (P)

wcmax Maximum width of chart (P)

Description Specifies the maximum width of a chart (plotting or printing area). Set when plotter or printer is installed.

Values Width, in mm.

See also	*NMR Spectroscopy User Guide*	
Related	wc	Width of chart (P)
	wc2	Width of chart in second direction (P)

wc2max Maximum width of chart in second direction (P)

Description	Specifies the maximum width of a chart (plotting or printing area) in the second direction (*y*-axis). Set when the plotter or printer is installed.
Values	Width, in mm.
See also	*NMR Spectroscopy User Guide*

Related	wc2	Width of chart in second direction (P)
	wcmax	Maximum width of chart (P)

wdone Specify action when experiment is done (C)

Syntax	`wdone(string)`
Description	Specifies the action to take when the experiment is done, after `wexp` has been executed. The `wdone` command sets the corresponding parameter `wdone`. Using the command, rather than setting the parameter value explicitly, notifies the acquisition process that the associated parameter value has changed and the desired operation is effected even if the experiment has already started.
Arguments	The `string` argument contains the command or macro to be executed when the experiment is done. The string must be enclosed in single quotes. If single quotes are required within the text string, place a backslash character before each of the interior single quotes (`\'`). Maximum length of the string is 256 characters. `''` (null string) turns off `wdone` processing.

Related	wexp	Specify action when experiment completes (C)

wdone Specify action when experiment is done (P)

Syntax `wdone'<command, macro, or null string >'`

Description Invokes a single action to occur just after `wexp` is executed. As with `wexp`, it is executed automatically after the experiment is finished, which can occur at the end of a single FID or after the last fid in a multi-FID experiment. To specify no `wdone` processing, set `wdone` to the null string. If the acquisition has already started, the `wdone` command must be used to change the `wdone` parameter. For wdone to execute after an experiment finishes and after wexp has executed, start the experiment with the au command.

If the wexp action sets the wdone parameter, the new value of the wdone parameter will be executed and the old value will be ignored.

werr — Specify action when error occurs (C)

Syntax `werr(string)`

Description Specifies what action to take if an error occurs during acquisition. The *command* werr sets the corresponding *parameter* werr. Using the command, rather than setting the parameter value explicitly, notifies the acquisition process that the associated parameter value has changed. Thus, the desired operation can be effected even if the experiment has already started.

Arguments `string` is a string argument containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. If single quotes are required *within* the text string, place a backslash character before each of the interior single quotes (\'). Maximum length of the string is 256 characters. To turn off werr processing, enter werr(''), where the argument is two single quotes with no space between them.

Examples
```
werr('react')
werr('')
```

See also *NMR Spectroscopy User Guide*

Related

wbs	Specify action when bs transients accumulate (C)
werr	When error (P)
wexp	Specify action when experiment completes (C)
wnt	Specify action when nt transients accumulate (C)

werr — When error (P)

Description Specifies a macro (e.g., werr='react') that will take appropriate action when an error occurs during acquisition. To specify no werr processing, set werr to the null string. If the acquisition has already been started, the werr *command* must be used to change the werr *parameter*. Arrayed parameter acqstatus provides the error code to werr in acqstatus[1] and acqstatus[2]. For a list of error codes, refer to the description of acqstatus or view the file acq_errors in directory /vnmr/manual.

Values Macro or null string (werr='', where the value is given by two single quotes with no space between them).

See also *NMR Spectroscopy User Guide*

Related acqstatus Acquisition status (P)

react	Recover from error conditions during werr processing (M)
werr	Specify action when error occurs (C)

wet — Flag to turn on or off wet solvent suppression ((P)

Description: Specifies if wet solvent suppression is turned on or off. It is now a standard option in many liquids pulse sequences, including Wet1d and sequences of apptype hetero2d and homo2d.

Related:

apptype	Application type (P)
hetero2d	Execute protocol actions of apptype hetero2d (M)
homo2d	Execute protocol actions of apptype homo2d (M)
std1d	Execute protocol actions of apptype std1d (M)
Wet1d	Set up parameters for a WET1D pulse sequence (M)

Wet1d — Set up parameters for wet ^{1}H experiment (M)

Description: Set up parameters for wet ^{1}H experiment.

wetdqcosy — Set up parameters for a WETDQCOSY pulse sequence (M)

Applicability: Systems with LC-NMR accessory.

Description: Sets up for a WETDQCOSY LC-NMR experiment.

See also: *NMR Spectroscopy User Guide*

wetgcosy — Set up parameters for a WETGCOSY pulse sequence (M)

Applicability: Systems with LC-NMR accessory.

Description: Sets up for a WETGCOSY LC-NMR experiment.

See also: *NMR Spectroscopy User Guide*

wetghmqcps — Set up parameters for a WETGHMQCPS pulse sequence (M)

Applicability: Systems with LC-NMR accessory.

Description: Sets up for a WETHMQCPS LC-NMR experiment.

See also: *NMR Spectroscopy User Guide*

wetghsqc Set up parameters for a WETGHSQC pulse sequence (M)

Applicability Systems with LC-NMR accessory.
Syntax `wetghsqc('nucleus')`
Description Sets up for a WETGHSQC LC-NMR experiment.
See also *NMR Spectroscopy User Guide*

wetgmqcosy Set up parameters for a WETGHSQC pulse sequence (M)

Applicability Systems with LC-NMR accessory.
Description Sets up for a WETGMQCOSY LC-NMR experiment.
See also *NMR Spectroscopy User Guide*

wetit Set up and create pulse shapes for Wet1d experiment (M)

Applicability *VnmrJ Walkup*
Description A macro to set up and create pulse shapes for a `Wet1d` experiment. It is based on suppressing the largest N peaks found in a spectrum.
Related `wetpeaks (P)`

wetnoesy Set up parameters for a WETNOESY pulse sequence (M)

Applicability Systems with LC-NMR accessory.
Description Sets up for a WETNOESY LC-NMR experiment.
See also *NMR Spectroscopy User Guide.*

wetpeaks Number of peaks for wet solvent suppression (P)

Applicability *Walkup*
Description Sets the number of peaks to be suppressed by wet solvent suppression for the `Wet1d` protocol. The `wetit` macro suppresses the N tallest peaks found in the scout spectrum, where `N` is specified by `wetpeaks`. The parameter is set by the *Number of peaks to suppress menu* on the Prescan page.

Values	1 to 7 for DirectDrive or UnityInova systems; 3 for Mercury systems are the default values.	
Related	Wet1d	Set up parameters for wet 1H experiment (M)
	wetit	Set up and create pulse shapes for Wet1d experiment (M)

wetpwxcal Set up parameters for a WETPWXCAL pulse sequence (M)

Applicability Systems with LC-NMR accessory.

Description Sets up for a WETPWXCAL LC-NMR pulse width calibration.

See also *NMR Spectroscopy User Guide*

wettntocsy Set up parameters for a WETTNTOCSY pulse sequence (M)

Applicability Systems with LC-NMR accessory.

Description Sets up for a WETTNTOCSY LC-NMR experiment.

See also *NMR Spectroscopy User Guide*

wetshape Shape for pwwet pulses (P)

Applicability Systems with LC-NMR accessory.

Description Sets the name of the shape used for `pwwet` pulses (e.g., `wetshape='wet'`).

See also *NMR Spectroscopy User Guide*

wexp Specify action when experiment completes (C)

Syntax `wexp(string)`

Description Specifies what action to take when the experiment completes. The `wexp` *command* sets the corresponding *parameter* `wexp`. Using the command, rather than setting the parameter value explicitly, notifies the acquisition process that the associated parameter value has changed. Thus, the desired operation can be effected even if the experiment has already started.

Arguments `string` is a string argument containing the command or macro to be executed when the experiment completes. The string must be enclosed in single quotes. If single quotes are required *within* the text string, place a backslash character before each of the interior single quotes (`\'`). Maximum length of the string is 256 characters. To turn off `wexp` processing, enter wexp(''), where argument is two single quotes with no space between them.

Examples	wexp('wft(\'all\') calcT1') wexp('')
See also	*NMR Spectroscopy User Guide*

Related

wbs	Specify action when bs transients accumulate (C)
werr	Specify action when error occurs (C)
wexp	When experiment completes (P)
wnt	Specify action when nt transients accumulate (C)

wexp When experiment completes (P)

Description Invokes a single action to occur automatically after the experiment is finished, which can occur after a single FID or after a number of FIDs in a multi-FID experiment. To specify no wexp processing, set wexp to the null string. If the acquisition has already started, the wexp *command* must be used to change the wexp *parameter*. For wexp to execute after an experiment finishes, start the experiment with the au command.

wexp processing occurs after wnt processing in a single FID experiment, and both can be used. wexp also occurs after wnt during the last FID of a multi-FID experiment. Thus, wnt='wft(\'all\')' wexp='calcT1' and wexp='wft(\'all\') calcT1' transforms each FID in a T_1 experiment as it is performed, and when each of the FIDs has been collected, performs the calculation of the T_1 using a hypothetical macro command calcT1. Notice the use of the backslash to include a single quotation mark inside the string.

Values Command, macro, or null string (wexp='', where the value is given by two single quotes with no space between them). If the command or macro uses a file name as an argument, specifying an absolute path is best. Be sure the path is valid and you have the appropriate write permission.

See also *NMR Spectroscopy User Guide*

Related

wexp	Specify action when experiment completes (C)
wnt	When number of transients (P)
au	Submit experiment to acquisition and process data (C)

wf Width of FID (P)

Description Width of the FID display. This parameter can be entered in the usual way or interactively controlled by selecting the sf wf button during a FID display.

Values 0 to the value of at, in seconds.

See also *NMR Spectroscopy User Guide*

Related
- at Acquisition time (P)
- dcon Display noninteractive color intensities map (C)
- dconi Interactive 2D data display (C)
- df Display a single FID (C)
- sf Start of FID (P)
- vf Vertical scale of FID (P)
- wf1 Width of interferogram in 1st indirectly detected dimension (P)
- wf2 Width of interferogram in 2nd indirectly detected dimension (P)

wf1 Width of interferogram in 1st indirectly detected dimension (P)

Description Sets the width of the interferogram display in the first indirectly detected dimension.

Values 0 to (2 × ni)/sw1, in seconds.

See also *NMR Spectroscopy User Guide*

Related
- ni Number of increments in 1st indirectly detected dimension (P)
- sf1 Start of interferogram in 1st indirectly detected dimension (P)
- sw1 Spectral width in 1st indirectly detected dimension (P)
- wf Width of FID (P)

wf2 Width of interferogram in 2nd indirectly detected dimension (P)

Description Sets the width of the interferogram display in the second indirectly detected dimension.

Values 0 to (2 × ni2)/sw2, in seconds.

See also *NMR Spectroscopy User Guide*

Related
- ni2 Number of increments in 2nd indirectly detected dimension (P)
- sf2 Start of interferogram in 2nd indirectly detected dimension (P)
- sw2 Spectral width in 2nd indirectly detected dimension (P)
- wf Width of FID (P)

wfgtest Waveform generator test (M)

Applicability Systems with a waveform generator.

Description Retrieves a parameter set and pulse sequence, and compiles the sequence, in order to set up an experiment to test the waveform generators.

See also *Waveform Generator Kit Installation*

wft Weight and Fourier transform 1D data (C)

Syntax (1) `wft<(<options,><'nf'><,start><,finish><,step>)>`
(2) `wft('inverse',exp_number,expansion_factor)`

Description Performs a Fourier transform on one or more 1D FIDs with weighting applied to the FID. The command executes a left-shift, zero-order phase rotation, and a frequency shift according to the parameters `lsfid`, `phfid`, and `lsfrq`, respectively, on the time-domain data prior to the weighting and Fourier transformation. The type of Fourier transformation to be performed is determined by `proc`. `wft` uses the same arguments as the command `ft`, and except for weighting, it functions the same as the `ft` command.

See also *NMR Spectroscopy User Guide*

Related

`ft`	Fourier transform 1D data (C)
`lsfid`	Number of points to left-shift `np` FID (P)
`lsfrq`	Frequency shift of the `fn` spectrum in Hz (P)
`phfid`	Zero-order phasing constant for `np` FID (P)
`proc`	Type of processing on `np` FID (P)

wft1d Weight and Fourier transform f_2 for 2D data (C)

Syntax (1) `wft1d(element_number)`
(2) `wft1d<(<options,><coefficients>)>`

Description Performs the first Fourier transformation along the dimension defined by `sw`, with weighting and matrix transposition. This allows the display of t_1 interferograms with the `dcon` and `dconi` commands.

Except for weighting, `wft1d` functions the same as the `ft1d` command. See the description of `ft1d` for further information.

Arguments Same as the arguments to `ft1d`. See the `ft1d` command for details.

See also *NMR Spectroscopy User Guide*

Related

`dcon`	Display noninteractive color intensity map (C)
`dconi`	Interactive 2D data display (C)
`ft1d`	Fourier transform along f_2 dimension (C)
`sw`	Spectral width in directly detected dimension (P)

wft1da   Weight and Fourier transform phase-sensitive data (M)

Values `wft1da<(options)>`

Description Processes 2D FID data as well as 2D planes at particular t_1 or t_2 times from a 3D data set for a pure absorptive display.

`wft1da` differs from `ft1da` only in that weighting of the time-domain data is performed prior to the Fourier transform. See the description of `ft1da` for further information.

Arguments Same as arguments to `ft2da`. See the `ft2da` command for details.

See also *NMR Spectroscopy User Guide*

Related		
	`ft1da`	Fourier transform phase-sensitive data (M)
	`ft2da`	Fourier transform phase-sensitive data (M)
	`wft2da`	Weight and Fourier transform phase-sensitive data (M)

wft1dac   Combine arrayed 2D FID matrices (M)

Syntax `wft1dac<(<mult1>,<mult2>, ,...<multn>)>`

Description Allows the ready combination of 2D FID matrices within the framework of the 2D Fourier transform program. Weighting is performed. This command requires that the data be acquired either without f_1 quadrature or with f_1 quadrature using the TPPI method. `wft1dac` is used with TOCSY (with multiple mixing times).

Arguments `mult1,mult2,...,multn` are multiplicative coefficients. The *n*th argument is a real number and specifies the multiplicative coefficient for the *n*th 2D FID matrix.

See also *NMR Spectroscopy User Guide*

Related		
	`ft1dac`	Combine arrayed 2D FID matrices (M)
	`Tocsy`	Set up parameters for TOCSY pulse sequence (M)
	`wft2dac`	Combine arrayed 2D FID matrices (M)

wft2d   Weight and Fourier transform 2D data (C)

Syntax `wft2d<(<options,>coefficients)>`

Description Performs a complete 2D transformation with weighting after 2D data has been acquired. If the first Fourier transformation has already been done using `ft1d`, `wft1d`, `ft1da`, or `wft1da`, then the `wft2d` command performs only the second transform.

For arrayed 2D experiments, a single array element can be transformed and weighted using the array element number as an argument. Interferograms can be constructed explicitly using the following coefficient table:
`wft2d(rr1,ir1,rr2,ir2,...ri1,ii1,ri2,ii2,...)`.

`wft2d('ptype',...)` transforms P-type spectra, and `wft2d('ntype',...)` transforms N-type spectra. The default is N-type.

`wft2d` also *completes* a 2D transform that has been started with `wft1d` (or related commands such as `wft1da`). The first transform will not be done again if it has already been performed. For phase-sensitive 2D experiments, the coefficients must be applied as part of the first transform (e.g., with `wft1da`) since the interferograms are formed at that stage. These coefficients need not be repeated when invoking the subsequent transform: a simple `wft2d` or `ft2d` can suffice.

See the `ft2d` command description for further information.

Arguments Same as the arguments to `ft2d`. See the `ft2d` command for details.

Examples
```
wft2d(1,0,0,0)
wft2d(2)
wft2d(1,0,1,0,0,1,0,1)
wft2d(.67,0,.33,0,0,.67,0,.33)
```

See also *NMR Spectroscopy User Guide*

Related

`dconi`	Interactive 2D data display (C)
`ft1d`	Fourier transform along f_2 dimension (C)
`ft1da`	Fourier transform "halfway" for pure absorption 2D data (M)
`ft2d`	Fourier transform 2D data (C)
`wft1d`	Weight and Fourier transform f_2 for 2D data (C)
`wft1da`	Weight and FT "halfway" for pure absorption 2D data (M)
`wft2da`	Weight and transform for pure absorption 2D data (M)

`wft2da` Weight and Fourier transform phase-sensitive data (M)

Syntax `wft2da<(options)>`

Description Processes 2D FID data, as well as 2D planes at particular t_1 or t_2 times, from a 3D data set for a pure absorptive display.

`wft2da` differs from `ft2da` only in that weighting of the time-domain data is performed prior to the Fourier transform. See the description of `ft2da` for further information.

Arguments Same as used with `ft2da`. See the `ft2da` command for details.

See also *NMR Spectroscopy User Guide*

Related

`ft1da`	Fourier transform phase-sensitive data (M)
`ft2da`	Fourier transform phase-sensitive data (M)
`wft1da`	Weight and Fourier transform phase-sensitive data (M)

`wft2dac` Combine arrayed 2D FID matrices (M)

Syntax `wft2dac<(<mult1><,mult2>,...<,multn>)>`

Description Allows the ready combination of 2D FID matrices within the framework of the 2D Fourier transform program. Weighting is performed. This command requires that the data be acquired either without f_1 quadrature or with f_1 quadrature using the TPPI method. `wft2dac` is used with TOCSY (with multiple mixing times).

Arguments `mult1,mult2,...,multn` are multiplicative coefficients. The `n`th argument is a real number and specifies the multiplicative coefficient for the *n*th 2D FID matrix.

See also *NMR Spectroscopy User Guide*

Related

`ft1dac`	Combine arrayed 2D FID matrices (M)
`ft2dac`	Combine arrayed 2D FID matrices (M)
`Tocsy`	Set up parameters for TOCSY pulse sequence (M)
`wft1dac`	Combine arrayed 2D FID matrices (M)

`wfttt3` Process f_3 dimension during 3D acquisition (M)

Description Allows f_3 processing of 3D data to be performed concurrently with data acquisition. To invoke this function, set `wnt='wftt3'` and use `au` to start the acquisition of the 3D data. When `wftt3` detects that all the FIDs comprising a (`t1,t2`) block have been acquired, it starts up the `ft3d` program in background to process that block of FIDs in f_3.

The 3D processing information file, created by entering `set3dproc` within VnmrJ, does not need to contain valid f_1 and f_2 processing information but only valid f_3 processing information. Once the f_3 processing is complete, a new 3D information file can be created for the f_1-f_2 processing stages that contains valid f_1 and f_2 processing information.

The non-standard string parameter `path3d` can be used to specify the directory into which the f_3 processed 3D data is to be stored. Normally, `path3d` is absent in the parameter set. If this is the case or if `path3d=''`, the f_3-processed 3D data is stored in the directory `curexp/datadir`. `path3d` can be created by entering `create('path3d','string') setgroup('path3d','display')`.

See also *NMR Spectroscopy User Guide*

Related

`au`	Submit experiment to acquisition and process data (C)
`create`	Create new parameter in a parameter tree (C)
`ft3d`	Perform a 3D Fourier transform (M,U)
`getplane`	Extract planes from a 3D spectral data set (M)
`path3d`	Path to currently displayed 2D planes from a 3D data set (P)
`select`	Select a spectrum or 2D plane without displaying it (C)
`set3dproc`	Set 3D processing (C)
`setgroup`	Set group of a parameter in a tree (C)
`wnt`	When number of transients (P)

which — Display which command or macro is used (M)

Syntax `which(name)`

Description Searches VnmrJ libraries and then displays on line 3 which VnmrJ command or macro with the given name will be executed. For macros, `which` displays the type of macro (user, local, application, or Agilent) and the path to the library.

Arguments `name` is the name of a command or macro.

Examples `which('wft')`

See also *User Programming*

Related		
	`exists`	Determine if a parameter, file, or macro exists (C)
	`hidecommand`	Execute macro instead of command with same name (M)

wnt — Specify action when nt transients accumulate (C)

Syntax `wnt(string)`

Description Specifies what action to take when `nt` transients accumulate. The `wnt` *command* sets the corresponding *parameter* `wnt`. Using the command, rather than setting the parameter value explicitly, notifies the acquisition process that the associated parameter value has changed. Thus, the desired operation can be effected even if the experiment has already started.

Arguments `string` is a string argument containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. If single quotes are required within the text string, place a backslash character before each of the interior single quotes (`\'`). Maximum length of the string is 256 characters. To turn off `wnt` processing, enter `wnt('')`, where the argument is two single quotes with no space between them.

Examples
```
wnt('wft(\'all\')')
wnt('')
```

See also *NMR Spectroscopy User Guide*

Related		
	`nt`	Number of transients (P)
	`wbs`	Specify action when `bs` transients accumulate (C)
	`werr`	Specify action when error occurs (C)
	`wexp`	When experiment completes (P)
	`wnt`	When number of transients (P)

wnt When number of transients (P)

Description Invokes a single action to occur automatically after the FID is finished (ct=nt) or after each FID in a multi-FID experiment involving an arrayed parameter. The most common processing to occur after an FID is an automatic weighting and Fourier transformation (i.e., wnt='wft'); however, this is normally not needed because the command ga is the exact equivalent of wnt='wft(\'acq\')' au (i.e., ga sets the wnt action automatically). To specify no wnt processing, set wnt to the null string. If the acquisition has already been started, the wnt *command* must be used to change this parameter.

Values Command, macro, or null string (wnt='', where the value is given by two single quotes with no space between them).

See also *NMR Spectroscopy User Guide*

Related
nt Number of transients (P)
wnt Specify action when nt transients accumulate (C)

wp Width of plot in directly detected dimension (P)

Description Sets the width of the displayed or plotted region of the spectrum.

Values Always stored in Hz, but can be entered in ppm by using the p suffix (e.g., wp=6p sets the width of plot to 6 ppm).

See also *NMR Spectroscopy User Guide*

Related
wp1 Width of plot in 1st indirectly detected dimension (P)
wp2 Width of plot in 2nd indirectly detected dimension (P)

wp1 Width of plot in 1st indirectly detected dimension (P)

Description Analogous to the wp parameter except that wp1 applies to the first indirectly detected dimension of a multidimensional data set.

See also *NMR Spectroscopy User Guide*

Related
wp Width of plot in directly detected dimension (P)
wp2 Width of plot in 2nd indirectly detected dimension (P)

wp2 Width of plot in 2nd indirectly detected dimension (P)

Description Analogous to the wp parameter except that wp2 applies to the second indirectly detected dimension of a multidimensional data set.

See also *NMR Spectroscopy User Guide*

Related	`wp`	Width of plot in directly detected dimension (P)
	`wp1`	Width of plot in 1st indirectly detected dimension (P)

`write` Write formatted text to a device (C)

Syntax
(1) `write('keywords'><,color|pen>`
`<,'reverse'>,x,y<,template>) <:height>`
(2) `write('alpha'|'printer'|'line3'|'error',template)`
(3) `write('reset'|'file'|'fileline',file<,template>)`
(4) `write('net',host,port, template)'`

Description
Writes text to a graphics screen or plotter in a given format (syntax 1), writes formatted text to another device (syntax 2), clears a file (syntax 3), or writes to a file (syntax 3). The input to the command comes from arguments in `template`, which can be parameters such as `n1` or `pw`.

Arguments
`'keywords'` identify the output device (`'graphics'|plotter'`) and the drawing mode (`'xor'|'normal'|'newovly'|'ovly'| 'ovlyC'`).

- `'graphics'|'plotter'` is a keyword selecting the output device. The default is `'plotter'`. The output selected is passed to subsequent `pen`, `move`, or `draw` commands and remains active until a different mode is specified.
- `''xor','normal'` is a keyword for the drawing mode when using the `'graphics'` output device. The default is `'normal'`. In the `'xor'` mode, if a line is drawn such that one or more points of the line are in common with a previous `'xor'` line, the common points are erased. In the `normal` mode, the common points remain. The mode selected is passed to subsequent `pen`, `move`, and `draw` commands and remains active until a different mode is specified.
- `'newovly'`, `'ovly'`, and `'ovlyC'` are keywords that specify an interactive drawing capability that is slightly slower than the `'xor'` mode but more consistent in color. `'newovly'` clears any previous draws, boxes, and writes made with the `'ovly'` modes and draws the figure. `'ovly'` draws without clearing so that multi-segment figures can be created. `'ovlyC'` clears without drawing.

`color` is the color of the text on a color display: `'red'`, `'yellow'`, `'green'`, `'cyan'`, `'blue'`, `'magenta'`, and `'white'`. The default is `'yellow'`.

`pen` is the plotter pen: `'pen1'`, `'pen2'`, etc.

`'reverse'` is a keyword specifying a sideways orientation of the output.

`x` and `y` are coordinates on the screen or plotter, in mm.

`template` is a string of formatting characters along with arguments to those characters. The format is the same as used with the UNIX `printf` command (for details, see any basic UNIX manual or enter `man printf` in UNIX). For example, `'pw = %12.5f'` is a template to format the parameter pw as fixed point with a field width of 12 spaces and 5 decimal places. The following format characters are implemented:

character	%c
integer	%d
hexadecimal	%h
exponential:	%e
fixed point	%f
exponential/fixed point	%g
octal	%o
string	%s
write a % character	use write(...'%s','%')

height returns the height of the characters on the screen or plotter. This is useful for positioning multiple-line displays. See the source code of the macro dtext in the maclib directory for an example of usage.

'alpha' is a keyword to write text to the alphanumeric screen.

'printer' is a keyword to print text on the printer

'line3' is a keyword to write text as a message on line 3.

'error' is a keyword to write text as an error on line 3 and sound a beep.

'reset' is a keyword to clear the file specified.

'file' is a keyword to append data to the file specified. Existing data in the file is not overwritten. By writing repeated 'file' calls, a formatted data file can be created (see the fifth example below). Each write command automatically appends a carriage return (line feed) to the end of the string defined by the template argument. To append data without the automatic line feed, use the 'fileline' keyword instead of 'file'. Also, two backslashes (\\) are interpreted as a new line.

'fileline' is a keyword to append data to the file specified, the same as using the 'file' keyword, but without automatically appending a carriage return (line feed) to the end of the data. Any line feeds desired must be explicitly defined (using \n) by the template argument (see the sixth example below). Furthermore, two backslashes (\\) output a single backslash into the file.

file is the name of the file used with the 'reset', 'file', and 'fileline' keywords.

'net' is a keyword for writing to a network program. The host name and port number must be supplied. The host name may also be an IP address, such as 10.190.x.y. The hostname of the local computer is stored in the instrument parameter. The command serverport may be used to get the port number for the currently executing VnmrJ program.

Examples

```
write('graphics',100,100):$ys
write('plotter',20,180, 'pw = %12.5f',pw)
```

```
write('line3', 'Too many arguments')
write('reset','temp1')
write('file','temp1','%10f %10.1f',n1,pw)
write('fileline','temp1','\nEnd of data\n\n')
serverport:$port
write('net',instrument,$port,'banner(`hello`)')
```

See also *User Programming*

Related		
	dtext	Display a text file in the graphics window (M)
	serverport	Returns the value of the VnmrJ network listening port (C)

writefid — Write numeric text file using a FID element (C)

Syntax `writefid(file<,element_number>)`

Description Writes a text file using data from the selected FID element. The program writes two values per line—the first is the value from the X (or real) channel and the second is the value from the Y (or imaginary) channel. `writefid` writes the raw FID data (i.e., FID data processing based on the parameters phfid, lsfid, and lsfrq does not occur).

Arguments `file` is the name of a text file to store the data.

`element_number` is an integer larger than 0 for the number of a FID element. The default is 1.

See also *NMR Spectroscopy User Guide, User Programming*

Related		
	lsfid	Number of complex points to left-shift `np` FID (P)
	lsfrq	Frequency shift of `fn` spectrum in Hz (P)
	makefid	Make a FID element using numeric text input (C)
	phfid	Zero-order phasing constant for `np` FID (P)
	writespectrum	Write a spectrum to a binary file (C)

writejxy — Create x,y ascii file from phasefile for JCAMP-DX conversion (M)

Syntax `writejxy<(traceno)>`

Applicability VnmrJ 3.1

Description "`writejxy`" does almost the same as "writexy", but in a mode that is adjusted for calls by the "svxyj" macro (JCAMP-DX X,Y data conversion).

writeparam Write one of more parameters to a file (C)

Syntax `writeparam(file,parlist[,tree]['add' | 'replace')`

Description The `writeparam` command will write one or more parameters to a specified file. The first argument is the name of the file. The second argument is a list of the names of the parameters to be written. It is a string parameter and the names can be separated either by a space or a comma. The optional third argument is the tree from which the parameters are copied.

The variable trees are `'current'`, `'global'`. `'processed'` and `'systemglobal'`.

An optional final argument is the keyword `'add'` or `'replace'`. The add keyword will cause the parameters to be appended to the specified file.

If they already exists in the file, their values will be updated. The replace keyword will replace the values in the file with the current values from the tree. The parameters must exist in both the file and the tree

A special case for the replace option occurs when the parameter list is an empty string. In this case, all the parameters in the file will be updated with the current values in the tree. If the parameter does not exist in the tree, no change will be made for that parameter.

This command may be used to store temporary values. For example, you may want to save `wexp`, `wbs`, `wnt`, etc. in order to run a setup acquisition. When it is done, you want to reset the original values. The `fread` command can to used to read the parameters back into an appropriate parameter tree.

Examples `writeparam(curexp+'/mypar','in')`

writes the parameter in into the file `mypar` in the current experiment directory.

`writeparam(curexp+'/mypar','sw ct np','processed')`

writes the parameters `sw`, `ct`, and `np` from the processed tree into the file `mypar` in the current experiment directory.

writespectrum write a spectrum to a binary file (C)

Description Writes out the current spectrum as a binary file. The file has no header information and is written in the native format (little-endian on Linux; big-endian on Solaris).

writespectrum scales the data by `vs`, determines the mode selected, `ph`, `av`, or `pwr`, and writes whatever is displayed by `ds.` The file is written in the current experiment as `specN`, where N is the element number.

Examples Write files spec1, spec2, spec3 ... spec{arraydim} in the current experiment directory:

```
wft $i=0 while ($i < arraydim) do $i = $i + 1 select($i)
writespectrum endwhile
```

Write the real and imaginary components if phase mode is selected.

```
wft
ph
$i=0
$index=''
while ($i < arraydim) do
     $i = $i + 1
     format($i,0,0):$index
     select($i)
     writespectrum
     mv(curexp+'/spec'+$index, curexp+'/
      spec'+$index+'.re')
     rp = rp + 90
     writespectrum
     mv(curexp+'/spec'+$index, curexp+'/
      spec'+$index+'.im')
     rp = rp - 90
endwhile
```

Related writefid Write numeric text file using a FID element (C)

writetrace Create ascii file from phasefile (f1 or f2) trace (M)

Syntax writetrace<(traceno)>

Applicability VnmrJ 3.1

Description "writetrace" creates an ASCII file from a phasefile trace in the current experiment. The argument indicates the number of the trace that is to be "asciified". The trace orientation depends on the orientation of the current data set (trace parameter). "writetrace" works on fids (1D, arrayed, 2D), interferograms and 1D/2D spectra. Trace counting starts at 1. The default trace is the current one. The output will be written into a file in the current experiment, using the trace number as filename extension:

- curexp+'/trace.1': 1D spectrum (can be 1st of an array)
- curexp+'/trace.8': 8th trace from arrayed 1D data set
- curexp+'/f2trace.13': 13th f2 trace from 2D data set
- curexp+'/f1trace.1024': 1024th f1 trace from 2D data set

NOTE: the data MUST have been displayed using the "ds" (1D) or "dcon" or related (2D) commands, otherwise the phased spectrum is not even generated, and "writetrace" can't work. For 2D data, also traces that are currently not on display must have been displayed in the current orientation once before, otherwise they may not exist in phasefile!

Examples `writetrace`
`writetrace(13)`
`writetrace(1024)`

writexy Create x,y ascii file from phasefile (f1 or f2) trace (M)

Syntax `writexy<(traceno)>`

Applicability VnmrJ 3.1

Description "`writexy`" does the same thing as "`writetrace`", except that it creates an output file with x and y pairs (one pair per line, x values in referenced Hz). Also here, the output will be written into a file in the current experiment, using the trace number as filename extension:

- curexp+'/xytrace.1': 1D spectrum (can be 1st of array)
- curexp+'/xytrace.8': 8th trace from arrayed 1D data set
- curexp+'/f2xytrace.13': 13th f2 trace from 2D data set
- curexp+'/f1xytrace.1024': 1024th f1 trace from 2D data set

Examples `writexy`
`writexy(13)`

wrtp Command string executed after rtp command (P)

Description Holds the command string that is executed after an rtp command finishes. It is mostly used to set frequency-dependent parameter values, such as sw, so that one parameter set can be used on all spectrometers.

Examples `wrtp='setsw(13p,-2p)'`

wsram Send hardware configuration to acquisition console (C)

Syntax `wsram<:$success>`

Description Sends new hardware configuration information to the acquisition console when `config` is used (e.g., to set `lockfreq`). `wsram` (write to static RAM) is not normally entered directly by the user.

Arguments `success` returns 1 if `wsram` is successful, or 0 otherwise.

See also *VnmrJ Installation and Administration.*

Related
`config`	Display current configuration and possibly change it (M)
`lockfreq`	Lock frequency (P)

wshim Conditions when shimming is performed (P)

Description Specifies when automatic shimming is to be used, according to the method specified by the parameter `method`.

Values `'n'` sets that no automatic shimming is performed. Even with `wshim` set to this value, the shimming procedure specified by the parameter `method` can be activated by using the `shim` command.

`'e'` or `'exp'` sets that automatic shimming is done before data acquisition.

`'s'` or `'samp'` sets that automatic shimming is done only at the beginning of the first experiment, following the change of a sample using the automatic sample changer.

`'g'` sets that automatic shimming using gradient shimming is done only at the beginning of the first experiment, following the change of a sample using the automatic sample changer. The parameter `method` is ignored. This option is only available in automation and is not used with the `go`, `ga`, or `au` commands.

`'f'` or `'fid'` set automatic shimming is done prior to the data collection of each new array member in a multi-FID experiment.

`'f`*n*`'`, where *n* is an integer, sets shimming is done prior to data collection of every *n*th FID (e.g., `wshim='f16'` shims prior to acquiring FIDs 1, 17, 33, etc.). This method is only relevant to arrayed or 2D experiments.

See also *NMR Spectroscopy User Guide*

Related `gf` Prepare parameters for FID/spectrum display in `acqi` (M)
`method` Autoshim method (P)

wtfile User-defined weighting in directly detected dimension (P)

Description Set to name of the file containing the user-written weighting function along the directly detected dimension. This dimension is referred to as the f_2 dimension in 2D data sets, the f_3 dimension in 3D data sets, etc. The shellscript `wtgen` is used to compile the user-written weighting module into an executable program. The source file is stored in the directory `vnmruser+'/wtlib'` with a `.c` file extension. The executable file is in the same directory and has the same name as the source file but has no file extension.

Values `file` is the name of the executable weighting function or the name of the weighting function text file.

`''` (two single quotes with no space in between) indicates `wtfile` is inactive and VnmrJ should not look for a user-written weighting function.

See also	*NMR Spectroscopy User Guide; User Programming*	
Related	wtfile1	User-defined weighting in 1st indirectly detected dimension (P)
	wtfile2	User-defined weighting in 2nd indirectly detected dimension (P)
	wtgen	Compile user-written weighting functions (C,U)

wtfile1 User-defined weighting in 1st indirectly detected dimension (P)

Description Set to the name of the file containing the user-written weighting function for the first indirectly detected dimension. This dimension is often referred to as the f_1 dimension of a multidimensional data set. Otherwise, `wtfile1` is analogous to wtfile.

See also *NMR Spectroscopy User Guide; User Programming*

Related	wtfile	User-defined weighting in directly detected dimension (P)
	wtfile2	User-defined weighting in 2nd indirectly detected dimension (P)

wtfile2 User-defined weighting in 2nd indirectly detected dimension (P)

Description Set to the name of the file containing the user-written weighting function along the second indirectly detected dimension. This dimension is often referred to as the f_2 dimension of a multidimensional data set. `wtfile2` can be set with wti on the 2D interferogram data. Otherwise, `wtfile2` is analogous to wtfile.

See also *NMR Spectroscopy User Guide; User Programming*

Related	wtfile	User-defined weighting in directly detected dimension (P)
	wtfile1	User-defined weighting in 1st indirectly detected dimension (P)
	wti	Interactive weighting (C)

wtgen Compile user-written weighting functions (M,U)

Syntax (From VnmrJ) `wtgen(file<.c>)`
(From UNIX) `wtgen file<.c>`

Description Allows compilation of a user-written weighting function that subsequently can be executed from within VnmrJ. `wtgen` performs the following functions:

- Checks for the existence of the /vnmr/bin directory and aborts if the directory is not found.
- Checks for files usrwt.o and weight.h in the /vnmr/bin directory and aborts if either of these two files cannot be found there.
- Checks for the existence of the user's directory and creates this directory if it does not already exist.
- Establishes in the wtlib directory soft links to usrwt.o and weight.h in the /vnmr/bin directory.
- Compiles the user-written weighting function, which is stored in the wtlib directory, link loads it with usrwt.o, and places the executable program in the same directory; any compilation and/or link loading errors are placed in the file errmsg in wtlib.
- Removes the soft links to usrwt.o and weight.h in the /vnmr/bin directory.

The name of the executable program is the same as that for the source file without a file extension (e.g., testwt.c is the source file for the executable file testwt).

Examples (From VnmrJ) wtgen('testwt')
(From UNIX) wtgen testwt.c

See also *User Programming*

Related

wtfile	User-defined weighting for t_2 (P)
wtfile1	User-defined weighting for t_1 (P)
wtfile2	User-defined weighting in ni2 dimension (P)

wti Interactive weighting (C)

Syntax wti<(element_number)>

Description Allows weighting parameters to be set interactively for both t_2 FIDs and t_1 interferograms. *wti* responds appropriately to phfid and lsfid for t_2 FIDs and to phfid1 and lsfid1 for t_1 interferograms. The following parameters can be interactively weighted:

- awc, awc1, and awc2 set the additive weighting constant; added in to the weighting function after the lb and sb (or sbs) contributions but before the gf (or gfs) contributions.
- gf, gf1, and gf2 set the Gaussian apodization constant, in seconds.
- gfs, gfs1, and gfs2 set the Gaussian function shift, in seconds; shifts the origin of the Gaussian function; active only if gf (or gf1) is active.
- lb, lb1, and lb2 set the line broadening factor, in Hz; a positive value gives sensitivity enhancement; a negative value gives resolution enhancement.
- sb, sb1, and sb2 set the sinebell time period, in seconds; a negative value give a sine squared bell.

- sbs, sbs1, and sbs2 set the sinebell shift, in seconds; shifts the origin of the sine bell; active only if sb (or sb1) is active.

  These parameters can be typed in or changed with the left mouse button in the proper field. The right mouse button turns off the spectrum for a faster response to changes in the weighting function.

Arguments: `element_number` specifies which FID element or interferogram trace is to be used in adjusting the weighting parameters. The default is the currently active element or trace.

Examples:
```
wti
wti(3)
```

See also: *NMR Spectroscopy User Guide*

Related:

lsfid	Number of complex points to left-shift `np` FID (P)
lsfid1	Number of complex points to left-shift `ni` interferogram (P)
phfid	Zero-order phasing constant for `np` FID (P)
phfid1	Zero-order phasing constant for `ni` interferogram (P)
wtia	Interactive weighting for 2D absorptive data (C)

wtia — Interactive weighting for 2D absorptive data (M)

Syntax: `wtia<(element_number)>`

Description: Allows weighting parameters to be set interactively for both t_2 FIDs and t_1 interferograms in 2D absorptive data. Refer to the description of the wti command for further information.

Arguments: `element_number` specifies which FID element or interferogram trace is to be used in adjusting the weighting parameters. The default is the currently active trace.

See also: *NMR Spectroscopy User Guide*

Related:

lsfid	Number of complex points to left-shift `np` FID (P)
lsfid1	Number of complex points to left-shift `ni` interferogram (P)
phfid	Zero-order phasing constant for `np` FID (P)
wti	Interactive weighting (C)

wtune — Specify when to tune (P)

Applicability: Liquids, VnmrJ Walkup, Automation

Description: Specify when automatic probe tuning will happen.

Syntax: `wtune ='value1<value2>...'`

Values:

`'s'` – when a new sample is inserted

`'e'` – before each experiment

`'o'` – change of operator

`'v'` – change of solvent

`'t'` – change of temperature

`'1'` – change of high band frequency (tn or dn)

`'2'` – change of low band frequency (dn or tn)

`'n'` – do not tune, if `'n'` is included in argument list, no tuning will occur.

Examples `wtune ='st12'`

The system will tune when a new sample is inserted (`s`) or the temperature changes for the current or new sample (`t`) or there is a change in the high band frequency (tn or dn) (`1`) or there is a change of low band frequency (dn or tn) (`2`).

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related		
	`tunemethod`	Method to use for tuning (P)
	`protune`	Macro to start ProTune (M)
	`wtunedone`	What to do after ProTune tuning is done (P)

`wtunedone` What to do after ProTune tuning is done (P)

Description Specific what to do after ProTune tuning is done. This is a local string parameter that does not exist by default and must be created to specify a command to be executed after tuning is finished.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related		
	`protune`	Macro to start ProTune (M)
	`create`	Create new parameter in a parameter tree (C)
	`wtune`	Specify when to tune (P)

`wysiwyg` Set plot display or full display (P)

Description Sets whether the window display is the same as the plot (“what you see is what you get,” or WYSIWYG) or is expanded to fill the window. This allows the user to scale the image to the full window, making it easier to view. This parameter is in the user’s global parameter file.

Values `'y'` makes the window picture size depend on the current plotter setting. Scaling the window does not change the ratio of the picture. This value is the default display condition.

`'n'` makes the window display expand, giving a full display.

See also *NMR Spectroscopy User Guide*

X

x0	X-zero position of HP pen plotter or Postscript device (P)
x1	X1 shim gradient (P)
x2y2	X2Y2 shim gradient (P)
x3	X3 shim gradient (P)
x4	X4 shim gradient (P)
xdiag	Threshold for excluding diagonal peaks when peak picking (P)
xgate	Load time counter (M)
xm1	Utility macro for study queue experiment manager (M)
xmaction	Perform study queue action (M)
xmactionw	Perform study queue action for walkup (M)
xmaddreq	Add a required protocol before the main protocol (M)
xmcheckreq	Check required protocol name (M)
xmconvert	Convert a temporarily stored study into a submitted study (M)
xmcopy	Copy protocols in a study queue (M)
xmdelete	Delete nodes in a study queue (M)
xmenablepanel	Enable or disable a parameter panel (M)
xmendq	End a chained study queue (M)
xmgetatts	Get study queue attributes (M)
xmHprescan	Set up and process Proton prescans (M)
xminit	Initialize an imaging study queue (M)
xmlockup	Move a study queue node up and lock it (M)
xmmakenode	Make a new study queue node (M)
xmnext	Find next prescan or next experiment in study queue (M)
xmprescan	Run prescans in study queue (M)
xmreact	Recover from error conditions during automation study (M)

xmreadnode	Read attributes from a study queue node (M)
xmrtpar	Retrieve parameters from a study queue node (M)
xmsample	Write enterQ entry for a sample for study queue – automation (M)
xmsara	Write sample enterQ entry for study queue– imaging (M)
xmsatfrq	Processing for Presat experiment (M)
xmselect	Action when study queue node is selected (M)
xmsetattr	Set an attribute for a study queue node (M)
xmsetatts	Set an attribute for a study queue node (M)
xmshowdata	Show data from a study queue node (M)
xmstartnightq	Start the night queue (M)
xmsubmit	Submit sample(s) to the study queue (M)
xmtime	Update the study queue time (M)
xmtune	Check tune parameter during automation (M)
xmwerr	Recover from acquisition error in study queue (M)
xmwexp	Processing macro for end of acquisition in study queue (M)
xmwritenode	Write study queue node attributes (M)
xmwritesq	Write study queue node order (M)
xpol	Cross-polarization (P)
xpolar1	Set up parameters for XPOLAR1 pulse sequence (M)
xy	XY shim gradient (P)
xz	XZ shim gradient (P)
xz2	XZ2 shim gradient (P)

x0 X-zero position of HP pen plotter or Postscript device (P)

Applicability Systems with a Hewlett-Packard pen plotter or a Postscript output device.

Description Adjusts the x-zero position on the chart. Use hpa to adjust x0 (and y0) to place the numbers in a pleasing position when filled in on the blank lines. x0 is part of vnmrsys/global and hence common to all experiments.

Values Number, in mm.

See also *NMR Spectroscopy User Guide*

Related	hpa	Plot parameters on special preprinted chart paper (C)
	y0	Y-zero position of HP plotter or Postscript device (P)

x1 X1 shim gradient (P)

Description Holds current setting of the X1 radial shim gradient.

Values If shimset is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related shimset Type of shim set (P)

x2y2 X2Y2 shim gradient (P)

Description Holds current setting of the X2Y2 radial shim gradient.

Values If shimset is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related shimset Type of shim set (P)

x3 X3 shim gradient (P)

Description Holds current setting of the X3 radial shim gradient.

Values If shimset is 1, 2, 10: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related shimset Type of shim set (P)

x4 X4 shim gradient (P)

Description Holds current setting of the X4 radial shim gradient.

Values –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related shimset Type of shim set (P)

xdiag — Threshold for excluding diagonal peaks when peak picking (P)

Description Used by the ll2d program to exclude diagonal peaks when peak picking.

To create the 2D peak picking parameters xdiag and th2d in the current experiment, enter addpar('ll2d').

Values Peaks within xdiag Hz of the diagonal will not be picked by ll2d. Setting xdiag to 0.0 will cause ll2d to pick all peaks, including diagonal peaks.

See also *NMR Spectroscopy User Guide*

Related

addpar	Add selected parameters to the current experiment (M)
ll2d	Automatic and interactive 2D peak picking (C)
th2d	Threshold for integrating peaks in 2D spectra (P)

xgate — Load time counter (M)

Applicability Systems with a solids module.

Syntax xgate(counts)

Description Loads the (12-bit) time counter on the pulse programmer with the specified number of counts and switches the counter to the external time base (the external trigger). On each trigger, the counter counts one unit down, and the next pulse sequence event starts when the count reaches zero. Often that time count will be just 1 (1.0, as the argument must be a floating point number). If the final pulse is to be performed after a longer delay, two options are available:

- Perform a normal delay, followed by the xgate(1.0) call.
- Calculate how many rotor cycles that delay would be (calculation is typically done based on a parameter srate) and then perform xgate with that calculated number of rotor triggers. Be aware that the only number of rotor cycles that can be counted this way is 4096, because the pulse programmer uses a 12-bit counter). At typical rotor speeds of 5 to 10 kHz, the “counted” delay is limited to 0.8 to 0.4 seconds.

Arguments counts is the number of counts to load into the time counter. The value must be a floating point number.

Examples xgate(5.0)

See also *User Guide: Solid-State NMR; VNMR Pulse Sequences*

Related

srate	Spinning rate for magic angle spinning (P)

xm1 Utility macro for study queue experiment manager (M)

Description A utility macro for setting study queue attributes and other study queue operations. Usually called from other macros, and not from the command line.

xmaction Perform study queue action (M)

Applicability *VnmrJ Walkup*, Imaging

Description Perform an action on an experiment node in the study queue. Usually called from study queue actions, and not from the command line.

xmactionw Perform study queue action for walkup (M)

Applicability *VnmrJ Walkup*

Description Perform an action on an experiment node in the study queue. Usually called from other macros, and not from the command line.

xmaddreq Add a required protocol before the main protocol (M)

Applicability *VnmrJ Walkup*, Imaging

Description Add a required protocol before the main protocol, when adding a protocol to the study queue. Usually called from other macros, and not from the command line.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related xmmakenode Make a new study queue node (M)

xmcheckreq Check required protocol name (M)

Applicability *VnmrJ Walkup*, Imaging

Description Check if a required protocol exists in the study queue, and return the full path filename to data, if data has been acquired. Usually called from plotting macros, and not from the command line.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related cqplot Macro to perform generic 2D plot (M)
plot2D Plot 2D spectra (M)

xmconvert Convert a temporarily stored study into a submitted study (M)

Applicability *VnmrJ Walkup*, Imaging

Description Convert a temporarily stored study into a submitted study. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related	xmsubmit	Submit sample(s) to the study queue (M)

xmcopy Copy protocols in a study queue (M)

Applicability *VnmrJ Walkup*, Imaging

Description Copy protocols within a study queue. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related	xmaction	Perform study queue action (M)
	xmactionw	Perform study queue action for walkup (M)

xmdelete Delete nodes in a study queue (M)

Applicability *VnmrJ Walkup*, Imaging

Description Delete nodes within a study queue. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related	sqfilemenu	Study queue file menu commands (M)
	xmaction	Perform study queue action (M)
	xmactionw	Perform study queue action for walkup (M)

xmenablepanel Enable or disable a parameter panel (M)

Description Enable or disable a parameter panel. Usually used to disable the Acquire panel for Imaging applications. Usually called only from a panel.

xmendq End a chained study queue (M)

Applicability *VnmrJ Walkup*

Description	End a chained study queue in the Walkup interface. Usually called by other macros.
See also	*VnmrJ Walkup*
Related	`xmnext` Find next prescan or next experiment in study queue (M)

`xmgetatts` Get study queue attributes (M)

Applicability	*VnmrJ Walkup*, Imaging
Description	Get study queue attributes.
See also	*VnmrJ Walkup, VnmrJ Imaging User's Guide*
Related	`xmaction` Perform study queue action (M)

`xmHprescan` Set up and process Proton prescans (M)

Applicability	*VnmrJ Walkup*
Description	A macro to set up and process prescans for Proton-type experiments (Proton, Presat, or Wet1d protocols). Usually called from other macros, and not from the command line.
See also	*VnmrJ Walkup, VnmrJ Imaging User's Guide*
Related	`Hprescan` Proton prescan (P) `std1d` Apptype macro for Standard 1D experiments (M)

`xminit` Initialize an imaging study queue (M)

Applicability	Imaging
Description	Initialize an imaging study queue. Usually called from other macros, and not from the command line.
See also	*VnmrJ Imaging User's Guide*
Related	`sqfilemenu` Study queue file menu commands (M)

`xmlockup` Move a study queue node up and lock it (M)

Applicability	*VnmrJ Walkup*, Imaging
Description	A macro to move a study queue node up above other completed nodes in the study queue, and lock it so it cannot be moved. This is usually done just prior to acquisition. Usually called from other macros, and not from the command line.

See also	*VnmrJ Walkup, VnmrJ Imaging User's Guide*	
Related	`acquire`	Acquire data (M)

`xmmakenode` Make a new study queue node (M)

Applicability	*VnmrJ Walkup*, Imaging	
Description	Create a new node in the study queue. Usually only called by other macros.	
See also	*VnmrJ Walkup, VnmrJ Imaging User's Guide*	
Related	`locaction`	Locator action (M)
	`xmaddreq`	Add a required protocol before the main protocol (M)

`xmnext` Find next prescan or next experiment in study queue (M)

Applicability	*VnmrJ Walkup*	
Description	Find the next prescan or next experiment in a study queue. It is used for chaining prescans and experiments. Usually only called by other macros.	
See also	*VnmrJ Walkup*	
Related	`acquire`	Acquire data (M)
	`startq`	Start a chained study queue (M)
	`xmprescan`	Run prescans in study queue (M)
	`xmwexp`	Processing macro for end of acquisition in study queue (M)

`xmprescan` Run prescans in study queue (M)

Applicability	*VnmrJ Walkup*	
Description	Run prescans in a study queue. Usually only called by other macros.	
See also	*VnmrJ Walkup*	
Related	`cqfindz0`	Run an experiment to find the value of z0 (M)
	`gmapshim`	Start gradient autoshimming (M)
	`prescan`	Study queue prescan (P)
	`xmnext`	Find next prescan or next experiment in study queue (M)

xmreact — Recover from error conditions during automation study (M)

Applicability: *VnmrJ Walkup*

Description: A macro to recover from error conditions during a study queue automated acquisition. Usually only called by other macros.

See also: *VnmrJ Walkup*

Related:

`acquire`	Acquire data (M)
`react`	Recover from error conditions during werr processing (M)

xmreadnode — Read attributes from a study queue node (M)

Applicability: *VnmrJ Walkup*, Imaging

Description: Read attributes from a study queue node. Usually only called by other macros

See also: *VnmrJ Walkup, VnmrJ Imaging User's Guide.*

Related:

`xmaction`	Perform study queue action (M)
`xmactionw`	Perform study queue action for walkup (M)
`react`	Recover from error conditions during werr processing (M)

xmrtpar — Retrieve parameters from a study queue node (M)

Applicability: Imaging

Description: Retrieve parameters from a study queue node after its parameters have been customized. Usually only called by other macros.

See also: *VnmrJ Imaging User's Guide*

Related:

`xmmakenode`	Make a new study queue node (M)
`xmselect`	Action when study queue node is selected (M)

xmsample — Write enterQ entry for a sample for study queue – liquids (M)

Applicability: *VnmrJ Walkup*, systems with automation such as sample changer or LC-NMR.

Description: Write the information required for a sample in the study queue when the sample is submitted. Usually only called by other macros.

See also: *VnmrJ Walkup*

Related:

`loc`	Location of sample in tray (P)
`xmsubmit`	Submit sample(s) to the study queue (M)

xmsara — Write enterQ entry for a sample for study queue – imaging (M)

Applicability Imaging

Description Halt or resume acquisition in the study queue, especially when using multiple viewports. Usually only called from interface panels.

xmsatfrq — Processing for Presat experiment (M)

Applicability *VnmrJ Walkup*

Description A macro to handle processing steps for the Presat experiment. It is optimized for use with water. Usually only called from other macros.

See also *VnmrJ Walkup*

Related `xmHprescan` Set up and process Proton prescans (M)

xmselect — Action when study queue node is selected (M)

Applicability *VnmrJ Walkup*

Description A macro to specify the action taken when a study queue node is selected by double-clicking on it. The action depends on the node status, which is Ready for acquisition, Executing, Completed, etc. The macro also runs the macros associated with selecting a study queue node, and saves the parameters of the current node before retrieving parameters of the selected node.

See also *VnmrJ Walku*

Related
`xmaction` Perform study queue action (M)
`xmactionw` Perform study queue action for walkup (M)
`xmrtpar` Retrieve parameters from a study queue node (M)

xmsetatts — Set an attribute for a study queue node (M)

Applicability *VnmrJ Walkup*, Imaging

Description Set an attribute for a study queue node.

See also *VnmrJ Walkup*, *VnmrJ Imaging User's Guide*

Related
`xmaction` Load colors for graphics window and plotters (M)
`xmactionw` Location of sample in tray (P)

`xmsetattr` Set an attribute for a study queue node (M)

Applicability *VnmrJ Walkup*, Imaging

Description Set an attribute for a study queue node.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related `xmaction` Load colors for graphics window and plotters (M)
`xmactionw` Location of sample in tray (P)

`xmshowdata` Show data from a study queue node (M)

Applicability *VnmrJ Walkup*, Imaging

Description A macro that retrieves data from a completed study queue node. In the Walkup liquids interface, data is also processed if *Process data on drag-and-drop* from locator is selected in the *System settings* dialog in the Utilities menu.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related `xmselect` Action when study queue node is selected (M)

`xmstartnightq` Start the night queue (M)

Applicability *VnmrJ Walkup*

Description Start the night queue. It also is used to initialize the night queue settings in the Utilities menu.

Examples `xmstartnightq` start the night queue
`xmstartnightq('at')` initialize the night queue settings.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related `walkup` Walkup automation (M)

`xmsubmit` Submit sample(s) to the study queue (M)

Applicability *VnmrJ Walkup*, systems with automation such as sample changer or LC-NMR.

Description Submit the sample or samples selected in the study queue tray. If the Submit DayQ button below the study queue area is selected, samples are submitted to the DayQ. If the Submit NightQ button is selected, samples are submitted to the NightQ.

See also *VnmrJ Walkup*

Related `xmsample` Write enterQ entry for a sample for study queue – automation (M)

xmtime Update the study queue time (M)

Applicability *VnmrJ Walkup*, systems with automation such as sample changer or LC-NMR.

Description Update the study queue time for both DayQ and NightQ. Usually only called from panels or other macros.

See also *VnmrJ Walkup*

Related

sqfilemenu	Study queue file menu commands (M)
startq	Start a chained study queue (M)
studytime	Study time (P)
xmsubmit	Submit sample(s) to the study queue (M)

xmtune Check tune parameter during automation (M)

Applicability Automation

Syntax `xmtune`

Description Check tune parameters in the study queue during automation and determine if tuning will occur. Macro is usually called from within automation and not from the command line.

See also *NMR Spectroscopy User Guide* and *VnmrJ Walkup*

Related

protune	Macro to start ProTune (M)
tunemethod	Method to use for tuning (P)
wtune	Specify when to tune (P)

xmwerr Recover from acquisition error in study queue (M)

Applicability *VnmrJ Walkup*, Imaging

Description Recover from an acquisition error in a study queue when not running automation. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related

acquire	Acquire data (M)
xmreact	Recover from error conditions during automation study (M)

xmwexp Processing macro for end of acquisition in study queue (M)

Applicability *VnmrJ Walkup*, Imaging

Description A processing macro; runs at the end of acquisition in the study queue and keeps track of study queue parameters and settings. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related		
	`acquire`	Acquire data (M)
	`xmreact`	Recover from error conditions during automation study (M)

`xmwritenode` Write study queue node attributes (M)

Applicability *VnmrJ Walkup*, Imaging

Description Write study queue node attributes. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related		
	`xmaction`	Load colors for graphics window and plotters (M)
	`xmactionw`	Location of sample in tray (P)
	`xmsetattr`	Set an attribute for a study queue node (M)

`xmwritesq` Write study queue node order (M)

Applicability *VnmrJ Walkup*, Imaging

Description Write the study queue node order. Usually only called from other macros.

See also *VnmrJ Walkup, VnmrJ Imaging User's Guide*

Related		
	`xmaction`	Load colors for graphics window and plotters (M)
	`xmactionw`	Location of sample in tray (P)

`xpol` Cross-polarization (P)

Applicability Systems with a solids module.

Description Selects cross-polarization or direct polarization in solid-state NMR experiments such asXPOLAR1.

Values `'n'` sets the experiment for direct polarization.
`'y'` sets the experiment for cross-polarization.

See also *User Guide: Solid-State NMR*

Related		
	`xpolar1`	Set up parameters for XPOLAR1 pulse sequence (M)

xpolar1 Set up parameters for XPOLAR1 pulse sequence (M)

Applicability: Systems with solids modules.

Description: Sets up the solid-state NMR cross-polarization experiment XPOLAR using the parameters. Otherwise, `xpolar1` contains the same functionality as `xpolar`.

See also: *User Guide: Solid-State NMR*

Related:

hsrotor	Display rotor speed for solids operation (P)
rotorsync	Rotor synchronization (P)

xy XY shim gradient (P)

Description: Holds current setting of the XY radial shim gradient.

Values: If shimset is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

Related:

shimset	Type of shim set (P)

xz XZ shim gradient (P)

Description: Holds current setting of the XZ radial shim gradient.

Values: If shimset is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

Related:

shimset	Type of shim set (P)

xz2 XZ2 shim gradient (P)

Description: Holds current setting of XZ2 radial shim gradient.

Values: If shimset is 2, 8: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

Related:

shimset	Type of shim set (P)

Y

y0	Y-zero position of HP pen plotter or Postscript device (P)
y1	Y1 shim gradient (P)
y3	Y3 shim gradient (P)
y4	Y4 shim gradient (P)
yz	YZ shim gradient (P)
yz2	YZ2 shim gradient (P)

y0 — Y-zero position of HP pen plotter or Postscript device (P)

Applicability Systems with a Hewlett-Packard pen plotter or a Postscript output device.

Description Adjusts the *y*-zero position on the chart. Use hpa to adjust y0 (and x0) to place numbers in a pleasing position when filled in on the blank lines. y0 is part of vnmrsys/global; therefore, it is common to all experiments.

Values Number, in mm.

See also *NMR Spectroscopy User Guide*

Related

hpa	Plot parameters on special preprinted chart paper (C)
x0	X-zero position of HP plotter or Postscript device (P)

y1 — Y1 shim gradient (P)

Description Holds current setting of the Y1 radial shim gradient.

Values If shimset is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current. If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

y3 Y3 shim gradient (P)

Description Holds current setting of the Y3 radial shim gradient.

Values If `shimset` is 1, 2, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

y4 Y4 shim gradient (P)

Description Holds current setting of the Y4 radial shim gradient.

Values –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

yz YZ shim gradient (P)

Description Holds current setting of the YZ radial shim gradient.

Values If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

yz2 YZ2 shim gradient (P)

Description Holds current setting of the YZ2 radial shim gradient.

Values If `shimset` is 2, 8: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

Z

z	Add integral reset point at cursor position (C)
z0	Z0 field position (P)
z1	Z1 shim gradient (P)
z1c	Z1C shim gradient (P)
z2	Z2 shim gradient (P)
z2c	Z2C shim gradient (P)
z2x2y2	Z2X2Y2 shim gradient (P)
z2x3	Z2X3 shim gradient (P)
z2xy	Z2XY shim gradient (P)
z2y3	Z2Y3 shim gradient (P)
z3	Z3 shim gradient (P)
z3c	Z3C shim gradient (P)
z3x	Z3X shim gradient (P)
z3x2y2	Z3X2Y2 shim gradient (P)
z3x3	Z3X3 shim gradient (P)
z3xy	Z3XY shim gradient (P)
z3y	Z3Y shim gradient (P)
z3y3	Z3Y3 shim gradient (P)
z4	Z4 shim gradient (P)
z4c	Z4C shim gradient (P)
z4x	Z4X shim gradient (P)
z4x2y2	Z4X2Y2 shim gradient (P)
z4xy	Z4XY shim gradient (P)
z4y	Z4Y shim gradient (P)
z5	Z5 shim gradient (P)

z5x	Z5X shim gradient (P)
z5y	Z5Y shim gradient (P)
z6	Z6 shim gradient (P)
z7	Z7 shim gradient (P)
z8	Z8 shim gradient (P)
zeroneg	Set all negative intensities of 2D spectra to zero (C)
zoom	Adjust display to given width (M)
zx2y2	ZX2Y2 shim gradient (P)
zx3	ZX3 shim gradient (P)
zxy	ZXY shim gradient (P)
zy3	ZY3 shim gradient (P)

z — Add integral reset point at cursor position (C)

Syntax *z*<(reset1,reset2,...)>

Description Resets the integral to zero at the point marked by the displayed cursor. The command cz removes all such integral resets and it should generally be used before starting to enter a series of integral zeros (resets). The resets are stored as frequencies and do not change if fn is changed.

Arguments reset1, reset2,... are reset points entered, in either Hz or ppm. The default is the cursor position). Reset points can be entered in any order.

Examples
```
z
z(7.5*sfrq,5*sfrq,2.5*sfrq,0.1*sfrq)
```

See also *NMR Spectroscopy User Guide*

Related

cz	Clear integral reset points (C)
dlni	Display list of normalized integrals (C)
ds	Display a spectrum (C)
fn	Fourier number in directly detected dimension (P)
nli	Find integral values (C)

z0 — Z0 field position (P)

Description Holds current setting of the Z0 setting. The value of z0 can be set by su. lockfreq can be used to find the lock signal or resonance. To use the lock frequency, deactivate z0 by typing the statement z0='n'. To activate z0, enter z0='y'.

Values If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `lockfreq` Lock frequency (P)
`su` Submit a setup experiment to acquisition (M)

z1 Z1 shim gradient (P)

Description Holds current setting of the Z1 axial shim gradient.

Values If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

z1c Z1C shim gradient (P)

Description Holds current setting of the Z1C axial shim gradient.

Values If `shimset` is 1, 2, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 5 or 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

z2 Z2 shim gradient (P)

Description Holds current setting of the Z2 axial shim gradient.

Values If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

z2c Z2C shim gradient (P)

Description Holds current setting of the Z2C axial shim gradient.

Values If `shimset` is 1, 2, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 5 or 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related `shimset` Type of shim set (P)

z2x2y2 Z2X2Y2 shim gradient (P)

Description Holds current setting of the Z2X2Y2 radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z2x3 Z2X3 shim gradient (P)

Description Holds current setting of the Z2X3 radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z2xy Z2XY shim gradient (P)

Description Holds current setting of the Z2XY radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z2y3 Z2Y3 shim gradient (P)

Description Holds current setting of the Z2Y3 radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3 Z3 shim gradient (P)

Description Holds current setting of the Z3 axial shim gradient.
Values If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*
Related `shimset` Type of shim set (P)

z3c Z3C shim gradient (P)

Description Holds current setting of the Z3C radial shim gradient.

Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3x Z3X shim gradient (P)

Description Holds current setting of the Z3X radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3x2y2 Z3X2Y2 shim gradient (P)

Description Holds current setting of the Z3X2Y2 radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3x3 Z3X3 shim gradient (P)

Description Holds current setting of the Z2X3 radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3xy Z3XY shim gradient (P)

Description Holds current setting of the Z3XY radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3y Z3Y shim gradient (P)

Description Holds current setting of the Z3Y radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z3y3 — Z3Y3 shim gradient (P)

Description: Holds current setting of the Z3Y3 radial shim gradient.

Values: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

z4 — Z4 shim gradient (P)

Description: Holds current setting of the Z4 shim gradient.

Values: If `shimset` is 1, 2, 8, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

Related: `shimset` Type of shim set (P)

z4c — Z4C shim gradient (P)

Description: Holds current setting of the Z4C shim gradient.

Values: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

z4x — Z4X shim gradient (P)

Description: Holds current setting of the Z4X shim gradient.

Values: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

z4x2y2 — Z4X2Y2 shim gradient (P)

Description: Holds current setting of the Z4X2Y2 radial shim gradient.

Values: –32768 to +32767, steps of 1, 0 is no current.

See also: *NMR Spectroscopy User Guide*

z4xy — Z4XY shim gradient (P)

Description: Holds current setting of the Z4XY radial shim gradient.

Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z4y Z4Y shim gradient (P)

Description Holds current setting of the Z4Y shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z5 Z5 shim gradient (P)

Description Holds current setting of the Z5 axial shim gradient.
Values If `shimset` is 2, 10: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*
Related `shimset` Type of shim set (P)

z5x Z5X shim gradient (P)

Description Holds current setting of the Z5X radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z5y Z5Y shim gradient (P)

Description Holds current setting of the Z5Y radial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z6 Z6 shim gradient (P)

Description Holds current setting of the Z6 axial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z7 — Z7 shim gradient (P)

Description Holds current setting of the Z7 axial shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

z8 — Z8 shim gradient (P)

Description Holds current setting of the Z8 shim gradient.
Values –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*

zeroneg — Set all negative intensities of 2D spectra to zero (C)

Description Sets to zero all negative intensities of 2D-J spectra.
See also *NMR Spectroscopy User Guide*
Related `foldj` Fold J-resolved 2D spectrum about f_1=0 axis (C)
`rotate` Rotate 2D data (C)

zoom — Adjust display to given width (M)

Syntax `zoom(width)`
Description Adjusts the display limits. It is useful in the display of powder patterns after `split` has been used. `zoom` both zooms in and out from the current display.
Arguments `width` is the total display width, in Hz. Display limits are set to ±`width/2`.
See also *NMR Spectroscopy User Guide*
Related `split` Split the difference between two cursors (M)

zx2y2 — ZX2Y2 shim gradient (P)

Description Holds current setting of the ZX2Y2 shim gradient.
Values If `shimset` is 2, 8: –2048 to +2047, steps of 1, 0 is no current.
If `shimset` is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.
See also *NMR Spectroscopy User Guide*
Related `shimset` Type of shim set (P)

zx3 ZX3 shim gradient (P)

Description Holds current setting of the ZX3 shim gradient.

Values –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

zxy ZXY shim gradient (P)

Description Holds current setting of the ZXY shim gradient.

Values If shimset is 2, 8: –2048 to +2047, steps of 1, 0 is no current.
If shimset is 3 to 7, 9: –32768 to +32767, steps of 1, 0 is no current.

See also *NMR Spectroscopy User Guide*

Related shimset Type of shim set (P)

zy3 ZY3 shim gradient (P)

Description Holds current setting of the ZY3 shim gradient.

Values –32768 to +32767, steps of 1, 0 as no current.

See also *NMR Spectroscopy User Guide*

Agilent Technologies

Printed in USA, Rev. B, August 2011

91000171