Abbreviated Command and Parameter List for the U400 and SPARC2

Table of Contents

Note: (C) identifies a command.
(M) identifies a macro command.
(P) identifies a parameter

Parameters are entered as variables or text strings in quotes followed by a return < >.
Examples: pw=21 < >
dm='nnn' < >

Commands and macros are executed by typing the command followed by a return < >.
Examples: jexp1 < >
dres < >

Setting up an Experiment

Parameters

alock **Automatic Lock Status (P)**

Description: Controls Autolock following the insertion of a sample, and following initiation of an acquisition with the commands *go*, *ga*, or *au*. It is possible to switch between simple (hardware) Autolock and simple lock with buttons in the interactive VNMR Acquisition window. Autolock does not work on the U400 since the software for the sample changer is not in place. Only two settings for alock should be used:
alock='*n*' sets Autolock to "not used".
alock='*u*' turns lock off (i.e., run experiment unlocked).

gain **Receiver Gain (P)**

Description: *gain=#* sets receiver gain, where # is a value from 0 to 59 (59 represents the highest possible receiver gain and Ø the lowest).
gain='*n*' enables Autogain and 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.
NOTE: Autogain is automatically enabled by the normal setup procedure.

load **Load Status of Displayed Shims (P)**

Description: *load*='*n*' allows the user to interactively shim with the values that are currently loaded in the shim DACs. This will ignore the old shim values stored in *dgs* of the current experiment. *load=´y´* allows the user to load shims from the current experiment or a stored shim file. *rts* will automatically set *load='y'* and load the desired shims. The user must perform a set up *su* and then set *load='n'* before entering the interactive VNMR Acquisition window to begin shimming.

lockgain **Lock Gain (P)**

Description: Lock gain is set by the user from 0 to 70 dB; normal values range from 40 to 60 dB. The value may be seen in the interactive VNMR Acquisition window or by the command line query, *lockgain?*

lockphase **Lock Phase (P)**

Description: Lock phase is set by the user from 0 to 360 degrees: normal range is probe and filter dependent. The lock phase should always be optimized for each sample after locking but before shimming. The value may be seen in the interactive VNMR Acquisition window or by the command line query, *lockphase?*

lockpower **Lock Power (P)**

Description: Lock power is set by the user in 1 dB steps, from 0 to 63 dB (63 is full power); the normal range is probe dependent but should not exceed 40 dB for any sample. The lock power should always be less than the lock gain. The value may be seen in the interactive VNMR Acquisition window or by the command line query, *lockpower?*

method **Autoshim Method (P)**

Description: *method*='*xyz*' selects method for automatic shimming, where *xyz* is the name of a file for one of the predefined shim methods in the /vnmr/shimmethods library. Standard methods include the following:
'*z1z2'* Selects shimming of the Z1 and Z2 gradients.
'*allzs*' Shims all spinning gradients (Z1 through Z4 or Z5, depending on the magnet type)
To display all available methods, type *ls*('*lvnmr/shimmethods*'*)*. Shim methods can also be stored in a user's shimmethods directory e.g., */home/username/vnmrsys/shimmethods*.
To initial shimming by the selected method, type the command line macro *shim*.

spin **Sample Spin Rate (P)**

Description: *spin=#* selects a regulated spin rate, where # is Ø (non-spinning) or a number from 5 to 39. *spin='n'* leaves the spin rate at the currently used value, and does not wait for regulated spinning before performing acquisition. A spin rate of 20 is best for this instrument. Entering a new value will not change the spin rate. The rate will change when a new sample is inserted or the commands *spin, to, ga,* or *au* are entered.

wshim **Conditions When Shimming is Performed (P)**

Description: Specifies when autoshimming will be used, according to the method specified by the parameter *method*. Some allowable values of *wshim*: are:

'*n*' Indicates no automatic shimming is performed. Even if *wshim='n'* the shimming procedure specified by the parameter *method* can be activated by using the *shim* command.

'*s*' or '*exp*' Indicates automatic shimming is done prior to data acquisition.

Commands and Macros

acqi **Start Up the Interactive Acquisition Display Process (C)**

Syntax: acqi

Description: Opens the interactive VNMR Acquisition window which allows interactive locking and shimming on the lock signal or FID. It is always necessary to type *acqi* following a UNIX logon and bootup to vnmr. If you can not connect to the LC VNMR acquisition window, LC MAIN, LC MORE, LC fixacq.

cexp **Create an Experiment (C)**

Syntax: cexp(*n*)

Description: Creates experiment *n*, where *n* is a number from 1 to 9. A maximum of 9 experiments can be created in any one user directory. Experiment 1 should be reserved for routine 1D experiemnts only. Experiment 5 is used as the add-subtract buffer and should not routinely be used for any other experiemnts.

Example: *cexp(3)*

See also: *delexp*

delexp **Delete an Experiment (C)**

Syntax: delexp(n)

Description: Deletes experiment *n*, where *n* is a number from 2 to 9 (experiment 1 cannot be deleted). It is always a good idea to delete and then create an experiment, load a standard 1D parameter set and then load 2D parameters on top of the 1D parameters. The user should never load a second 2D experiment on top of another 2D experiment.

Example: *delexp(3)*

See also: *cexp*

e or ***eject*** **Eject Sample (M, C)**

Syntax: eject

Description: Turns on the eject air and slow drop air to eject a sample from the probe. The macro *e* is equivalent in function to the command *eject.*

See also: *eject, i, insert*

explib **Display Experiment Library (M)**

Syntax: explib

Description: Displays a library of the currently available experiment files (exp1, exp2,...exp9). For each experiment, the name of the experiment, the current size of the experiment, the pulse sequence currently active in the experiment, and the first 50 characters of the text file in the experiment are displayed.

i or ***insert*** **Insert Sample (M, C)**

Syntax: insert

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 the command *insert*.

See also: *insert, e, eject*

jexp **Join an Existing Experiment (C)**

Syntax: jexp#

Description: Joins experiment #, where # is a number from 1 to 9 describing an existing experiment. After this command, all actions (including changes of parameters, acquisition of data, and display of data) apply to the parameter and data of experiment # until the next *jexp* command.

mp **Move Parameters Between Experiments (C)**

Syntax: mp(from,to)

Description: Moves text and current display, processing, and acquisition parameters from one experiment to another. No FID is transferred. If only one argument is given, parameters are moved from the currently active experiment to the number of the experiment given by the argument.

Examples: *mp(4)* Move parameters from current experiment to exp4
mp(2,3) Move parameters from exp2 to exp3

rt **Retrieve FID (C)**

Syntax: rt('filename')

Description: Retrieves all FIDs stored in the file *filename.fid* into the current experiment. If *filename.fid* does not exist, and *filename.par* does, it retrieves the parameters only from *filename.par*. If executed without an argument, *rt* will ask for a filename.

Examples: *rt*
rt('*lvnmr/fidlib/fidld*')

See also: *rtp*

rtp **Retrieve Parameters (C)**

Syntax: rtp('filename')

Description: Retrieves parameters from *filename.par*, if this file exists. If not, and if *filename.fid* exists, it retrieves the parameters only from *filename.fid*. If executed without an argument, *rtp* will ask for a filename.

Examples: *rtp*
rtp('*vnmr/fidlib/fidld*')

See also: *rt*

rts **Retrieve Shim Coil Settings**

Syntax: rts('filename')

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 the *su* command (or other related command). If the filename is an absolute path, *rts* uses it with no modifications. Otherwise, *rts* searches two different directories, as follows: First, it looks for a */home/directorynamevnmrsys/shims* subdirectory. If this does not work, *rts* searches for the file in the directory */vnmr/shims*.. If the command cannot find the shim file, it displays the directory names it searched.

Example: *rts*('*acetone*')

spin **Submit a Spin Setup Experiment to Acquisition (C)**

Syntax: spin

Description: Regulates sample spinning according to the parameter *spin*, using the acquisition computer. The *spin* command also sets the rf frequency, decoupler status, and temperature.

See also: *go, ga, au, su, lock, shim*

su **Submit a Setup Experiment to Acquisition (C)**

Syntax: su

Description: Sets up the hardware of the system to match the current parameters but does not initiate data acquisition. *su* is used to change the frequency of the system in preparation for probe tuning, to change the temperature of the sample in advance of beginning an experiment (or after a VT experiment is run), and to turn the decoupler on or off. *su* does *not* destroy any existing data in the current experiment (only *go, ga,* and *au* do that).

See also: *go, ga, au, lock, shim, spin*

svf **Save FIDs In Current Experiment (C)**

Syntax: svf('filename')

Description: Saves all FIDs in current experiment. A file named *filename.fid* is created, which contains the parameters, text, and FID data. *svf* does not remove the data from the current experiment; it merely saves a copy of that data in a different file. The complete data set can be retrieved with the *rt* command; parameters only can be retrieved with the *rtp* command. It is *not* possible to save only one (or several) FIDs from an arrayed experiment; all of them are saved together in one file.

An attempt to store data using a filename that already exists will result in a displayed message.

```
File exists, overwrite (enter y or n <return>)?
```

Examples: *svf*

svf('*/home/vnmrsys/fids/username/data.fid*')

svp **Save Parameters from Current Experiment (C)**

Syntax: svp('filename')

Description: Save parameters from current experiment. A file named *filename.par* is created, which contains the parameters and text. The parameter set can be retrieved with the *rtp* and *rt* commands. If no filename is given, the command will ask for one.

Examples: *svp*('*/home*')

svs **Save Shim Coil Settings (C)**

Syntax: svs('filename')

Description: Saves all shim coil settings except Z0 into the *filename* given. If the filename is an absolute path, *svs* uses it with no modifications. Otherwise, *svs* will save the shims in */home/vnmrsys/shims*.

Examples: *svs*('*acetone*')

See also: *rts*

Acquiring Data

Parameters

at **Acquisition Time (P)**

Description: Length of time during which each FID is acquired, in seconds. *at* can be entered directly or indirectly by using the parameter *np*.

bs **Block Size (P)**

Description: As data are acquired, the *bs* parameter tells the acquisition computer to store the data on the disk every so many scans. *bs*='*n*' disables this block size storage. bs should be set to a small number (16, 32, or 64) for nt < 500 but should be set to a larger number (100, 200, or 500) for nt > 1000. **If *bs*='*n*', data are stored on disk only at the end of the experiment. Data will be lost if the experiment is aborted prior to termination.**

Limits: '*n*', 1 to 32767

ct **Completed Transients (P)**

Description: An 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 which is updated every five seconds. The parameter *ct*, displayed in the acquisition parameter group by *dg* command, is only updated when *wft* or *ft* are performed 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.

d1 **First Delay (P)**

Description: Length of the first delay or preacquisition delay in the *s2pul* and most other pulse sequences. This is the delay used to allow recovery of magnetization back to equilibrium following a read pulse and acquisition.

Limits: 0 to 8190 seconds, entered in units of seconds; smallest value possible is 0.2 μs (2e-7); finest increment possible is 0.1 μs.

d2 **Second Delay (P)**

Description: Length of the second delay in *s2pul*.

Limits: 0 to 8190 seconds, entered in units of seconds; smallest value possible is 0.2 μs (2e-7); finest increment possible is 0.1 μs.

date **Date (P)**

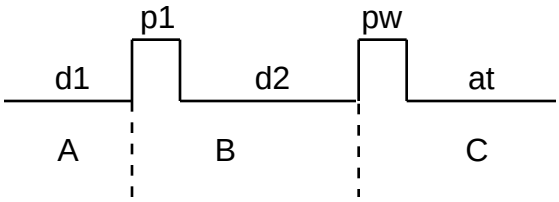
Description: An informational parameter taken from the UNIX-level calendar which cannot be entered by the user. Whenever data are acquired, the date is copied from UNIX and written into the acquisition parameters.

dfrq **Decoupler Transmitter Frequency (P)**

Description: Frequency in MHz of the decoupler transmitter frequency.

dm **Decoupler Mode (P)**

Description: Determines decoupler output, using '*y*' to turn on the decoupler and '*n*' to turn it off. Each pulse sequence is divided into periods and the decoupler status can be controlled separately during each period. EXAMPLE: *s2pul* sequence



A is the period during the delay *d1*, B the period during the delay *d2*, and C is the period during the acquisition time *at*. Three decoupler status modes can therefore be specified. For example, to turn the decoupler on during period A, off during period B, and on during period C, set *dm*='*yny*'. *dm*='*nny*' gives a decoupled spectrum without NOE, while *dm*='*yyn*' gives a coupled spectrum with NOE. If the decoupler status is constant for the entire pulse sequence, it can be entered as a single letter, i.e., *dm*='*n*' or *dm*='*y*'.

dmf **Decoupler Modulation Frequency (P)**

Description: Controls the modulation of the decoupler using the WALTZ-16 decoupling. After calibrating the decoupler field strength γH_2 (expressed in units of Hz), *dmf* should be set to equal $4*\gamma H_2$. In *e* (efficient) mode decoupling (see *dmm*, below), *dmf* is inactive.

Limits: 100 to 32700; for normal WALTZ decoupling should be between 5000 and 9000.

dmm **Decoupler Modulation Mode (P)**

Description: The decoupler can be set to the following modulation modes:

'*c*' Continuous or single-frequency decoupling
'*e*' Efficient decoupling (swept-square wave modulation)
'*s*' Super-efficient WALTZ-16 decoupling which requires a WALTZ board

Normally *dmm* is set to a single state in the *s2pul* sequence; however, this is not fixed. For example, *dmm = 'ccs'* gives single-frequency decoupling during the first part of the pulse sequence, and WALTZ-16 decoupling during acquisition. In any pulse sequence which uses the decoupler for hard pulses, *dmm* must be set to '*c*' during those periods of the pulse sequence.

dn **Decoupler Nucleus (P)**

Description: The decoupler nucleus *dn* is selected by the user such as '*H1*' or '*F19*' and sets the *dfrq* automatically.

See also: *tn*

dof **Decoupler Offset (P)**

Description: Controls the exact positioning of the decoupler. Higher numbers move the decoupler to higher frequency (to the left or "low-field" side of the spectrum). This parameter is set automatically for homonuclear decoupling by the *sd* or *sda* commands.

Limits: -100000 to 100000 Hz (approximate, depends on frequency) in steps of 0.1 Hz.

See also: *tof, sd, sda*

dp **Double Precision (P)**

Description: If *dp*='*n*', data are acquired in a 16-bit integer format. If *dp*='y', data are acquired in a 32-bit integer format. *dp*='*y*' allows continuous time averaging and does not allow ADC overflow when the digital convertor is full. *dp* should be set to '*y*' for all 2D experiments and almost all 1D experiments. However, *fn* and *np* cannot be greater than 64K if *dp='y'*.

Values: '*n*', '*y*'

dpwr **Decoupler Power with Linear Amplifiers (P)**

Description: The decoupler power has a range of 0 to 63 dB in units of 1 dB. For broadband WALTZ decoupling of protons, typical values range from 36 to 49 and **should never exceed 49 dB**.

fb **Filter Bandwidth (P)**

Description: Filter bandwidth in Hz used to set the audio filters which prevent noise of higher frequency than the spectral limits from "folding in" to the spectrum. The standard value of *fb* is 10% more than half of the spectral width *sw*. *fb* is automatically changed whenever the spectral width *sw* is changed. For example, typing *sw*=4000 will automatically set *fb*=2200, which is 10% more than 2000 Hz. After changing *sw*, *fb* may be changed to a different value.

Limits: 50 to 49500 Hz

homo **Homodecoupling Mode (P)**

Description: *homo*='*y*' is used for homonuclear decoupling, in which the receiver is gated off during the time the decoupler is on.
homo='*n*' is used for heteronuclear decoupling or for cases in which the decoupler is off during acquisition.

in **Interlock (P)**

Description: If *in* is set to '*y*', *and* if the lock level falls below a preset level (about 20 on the "lock meter"), acquisition is halted. Also, if *spin* is set to a particular value, and the spin rate goes out of regulation, acquisition is also halted. If *in*='*n*', acquisition continues irrespective of the lock channel or spin rate.

np **Number of Data Points (P)**

Description: 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*. *np* is constrained to be a multiple of 2.

nt **Number of Transients (P)**

Description: The number of transients to be acquired, i.e., the number of repetitions or "scans" performed to make up the experiment. To set up an indefinite acquisition, set *nt* to a very large number, such as 1e9. *nt* should always be a multiple of 2.

Limits: 1 to 1e9

p1 **First Pulse Width (P)**

Description: Sets the length of the first pulse in the *s2pul* sequence in µs. Never set higher than 100 µsec without consulting the MSL staff first.

Limits: 0, 0.2 to 8190 µs in steps of 0.1 µs.

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". However, if the temperature is arrayed and time must be allowed for equilibrium, pad can be set to 300 sec to allow five minutes equilibration at each temperature.

pw **Pulse Width (P)**

Description: Length of the read pulse in the *s2pul* sequence, in µs. In "normal" 1D experiments with just a single pulse per transient, this is the observe pulse width. Never set higher than 100 µsec without consulting the MSL staff first.

Limits: 0, 0.2 to 8190 µs in steps of 0.1 µs.

pw90 **90˚ Pulse Width (P)**

Description: Length of the 90˚ pulse, in µs. *pw90* is not used by pulse sequences directly, but is used by a number of commands to assist in setting up special experiments. Note that this parameter must be updated by the user and is not automatically determined or magically correct under all circumstances. The most recent 90° pulse widths for standard samples are found in the instrument logbook and are stored with the standard parameter directories. Never set higher than 100 µsec without consulting the MSL staff first.

Limits: 0 to 8190 µs in steps of 0.1 µs.

sfrq **Observe Transmitter Frequency (P)**

Description: Frequency in MHz of the observe transmitter. Set automatically when *tn* is entered.

solvent **Lock Solvent (P)**

Description: Selects one of a series of lock solvents. Standard values include:

DeuteriumOxide	CDCl3
D2O	Cyclohexane
Acetone	C6D12
CD3COCD3	Toluene
Benzene	C6D5CH3
C6D6	Acetic_Acid
DMSO	CD3COOD
MethyleneChloride	MethylAlcohol-d4
CD2CL2	CD3OD
Chloroform	

ss **Steady State Pulses (P)**

Description: The number of complete executions of the pulse sequence which occur prior to the acquisition of the real data (sometimes known as "dummy scans") This number of pulses is applied at the beginning of each experiment *only*. In particular, in an arrayed experiment, only the first member of the array will have the steady-state pulses.

Limits: 0 to 32767.

sw **Spectral Width (P)**

Description: The total width of the spectrum to be acquired, from one end to the other, in Hz.

Limits: 100 Hz to 100000 Hz

temp **Sample Temperature (P)**

Description: Temperature of sample, entered in degrees Celsius (˚C). *temp*='*n*' instructs the acquisition system not to set the VT controller and to ignore temperature regulation throughout the course of the experiment.

Limits: -150 ˚C to +200 ˚C, in steps of 0.1˚.

tn **Transmitter Nucleus (P)**

Description: Observe nucleus, used to extract a value from a look-up table coded by atomic weights. Typical values are '*H1*', '*C13*', '*P31*', etc. The nuclei are in the */vnmr/nuctables* directory.

Limits: Restricted to values entered in the */vnmr/nuctables* directory.

tof **Transmitter Offset (P)**

Description: The exact position of the transmitter is controlled by the transmitter offset position *tof*. Higher numbers move the transmitter to higher frequency (to the left or "low-field" side of the spectrum).

Limits: -100000 to +100000 Hz

tpwr **Transmitter Power (P)**

Description: The transmitter power can have values from 0 to 63 dB in units of one dB, where 63 is the maximum possible power. Normal values are 55 to 60 for all nuclei. Do not adjust this value without permission from MSL staff.

Limits: 0 to 63 dB

wbs **When Block Size (P)**

Description: Tells the computer what to do when a block size is complete. For example, *wbs='wft'* results in an automatic weighting and Fourier transformation after each *bs* transients. If the acquisition has already been started, the *wbs* command must be used to change this parameter.

Examples: *wbs='dg wft'*
wbs='wft'
wbs='mf(3)'

werr **When Error (P)**

Description: Specifies a macro that will take appropriate action when an error occurs during acquisition. If the acquisition has already been started, the *werr* command must be used to change this parameter.

wexp **When Experiment (P)**

Description: Invokes a single action to occur automatically after the experiment is finished. If the acquisition has already been started, the *wexp* command must be used to change this parameter.

Example: *wnt=' wft(' all')'*

wnt **When Number of Transients (P)**

Description: Invokes a single action to occur automatically after the FID is finished (*ct=nt*). The most common processing to occur after an FID is an automatic weighting and Fourier transformation, i.e., *wnt='wft'*. If the acquisition has already been started, the *wnt* command must be used to change this parameter.

Examples: wnt='*wft*'
wnt='*wftp1*'

Commands and Macros

aa **Abort Acquisition with Error (C)**

Description: Aborts an experiment that has been submitted to acquisition. If the experiment is active, it is aborted immediately. All current data for the experiment is discarded: any data collected from an earlier block size transfer is retained. Following the *aa* command, any queued experiments will begin. It is more preferrable to use *sa* which allows the experiment to be restarted if desired.

See also: *halt, sa, ra*

au **Submit Experiment to Acquisition and Process Data (C)**

Description: Performs the experiment described by the current acquisition parameters and checks the parameters *loc, spin, gain, wshim, load,* and *method* to determine the necessity to perform various actions in addition to simple data acquisition. *au* will also follow any commands given in *werr, wbs, wnt,* and *wexp*.

See also: *go, ga, su, lock, shim, spin*

banner **Display Message to Screen with Large Characters**

Syntax: banner('message')

Description: Displays message in large characters to the VNMR graphics window. Lines of text must be separated by two backslashes \\ between the lines.

Example: banner('HELLO') displays the message "HELLO" to the screen.

da **Display Acquisition Parameter Arrays (C)**

Description: Displays all arrayed acquisition parameters to the screen.

dg **Display Parameter Groups (Acquisition/Processing) (C)**

Description: Displays the acquisition and 1D/2D processing parameters.

See also: *dg1* (M), *dg2* (M), *dgs*(M),*da*

dg1 **Display Parameter Groups (Display) (M)**

Description: Displays the display and plotting parameters.

dgs **Display Parameter Groups (Special/Automation) (M)**

Description: Displays the "special" (parameters created for the current experiment) and shim parameters.

dps **Display Pulse Sequence (C)**

Syntax: dps<('filename')>

Description: Displays the pulse sequences specified by *filename*, or the current pulse sequence if no argument is entered.

ga **Submit Experiment to Acquisition and FT the Result (C)**

Description: Performs the experiment described by the current acquisition parameters, checking the parameters *loc, spin, gain, wshim, load,* and *method* to determine whether to perform various actions in addition to simple data acquisition. This may involve a single FID or an arrayed 1D experiment, but should not be used for 2D experiments; *au* or *go* are better choices for arrayed 2D experiments.

See also: *au, go*

movesw **Move Spectral Window According to Cursors (M)**

Syntax: movesw<(width)>

Description: Adjusts the transmitter offset *tof* and the spectral width *sw* to closely match the region controlled by the two cursor positions. Both cursors must be visible on the screen when *movesw* is typed. *movesw* by itself uses the distance between cursors to determine the new spectral width. The optional parameter allows *sw* to be explicitly defined in Hz.

Examples: *movesw*
movesw(*5000*)

See also: *movetof*

movetof **Move Transmitter Offset (M)**

Description: Changes the transmitter offset parameter *tof* to the frequency of the cursor.

See also: *movesw*

p1 **Enter Pulse Width *p1* in Degrees (C)**

Syntax: p1(p1degrees,<90degrees>)

Description: Calculates the pulse length in μsec when given the desired flip angle in degrees and the 90° pulse width in μsec. When no second argument is given, this command defaults to the *pw90* value.

Example: *p1(90,12.8)* Set *p1* to a 90˚ pulse and *pw90* to 12.8 μsec

See also: *pw*

pw **Enter Pulse Width *pw* in Degrees (C)**

Syntax: pw(pwdegrees,<90degrees>)

Description: Calculates the pulse length in μsec given a desired flip angle in degrees and the 90˚ pulse width in μsec. The calculated value is entered into the pulse width parameter *pw*. If no second argument is entered, *pw90* is taken as the 90˚ pulse.

Example: *pw(30)* Set *pw* to a 30˚ pulse using current *pw90*.

ra **Resume an Acquisition Stopped with *sa* Command (C)**

Description: Resumes an experiment acquisition that was stopped with the *sa* (stop acquisition) command. The parameters *dp* and *np* may not be altered before the experiment is resumed but *nt* may be increased or decreased.

See also: *sa*

sa **Stop Acquisition (C)**

Syntax: sa<('option')> or sa<(number)>

Description: Stops the experiment that is currently acquiring. Data is retained. The experiment can be stopped at several user-specified places during acquisition by entering one of the following options or an integer number as an argument.

'eos', '*ct*', '*scan*'	Stop at the next *ct* (completed transients).
'*eob*', '*bs*'	Stop at the next block size.
'*eof*', '*nt*', '*fid*'	Stop at the next complete FID.
'*eoc*', '*il*'	Stop at the next complete *il* (interleave) cycle; i.e., the latest block size has been complete for all FIDs in the interleave cycle.
number	Stop at the next *ct*, where the value of *ct* is a multiple of *number* (an integer number). This is useful when you wish to complete a phasecycle before stopping.

Examples: *sa* Stop at end of next data accumulation
sa('*ct*') Stop at end of next *ct*
sa(*4*) Stop when the value of *ct* is a multiple of 4

sd **Set Decoupler Frequency (M)**

Description: *sd* sets the decoupler offset parameter *dof* to the place in the spectrum specified by one cursor. This will work only if the parameter *tn* is the same as the parameter *dn*.

sda **Set Decoupler Frequency Array (M)**

Description: To set up an array of decoupler offset values, use *sd* for the first decoupler position and *sda* for subsequent decoupler positions. This will work only if the parameter *tn* is the same as the parameter *dn*.

setup **Set Up Parameters for Basic Experiments (M)**

Syntax: setup<('nucleus'<,'solvent'>)>

Description: Entering *setup* without arguments displays a menu with choices of nuclei. After choosing one, a choice of lock solvents is presented; one of these is selected. A parameter set is now returned to do the experiment requested, complete with positioning of the transmitter and decoupler.

Examples:

setup	Use menu system to set up experiment
setup('*H1*' , '*CDCl3*')	Set up routine proton experiment in chloroform
setup('*C13*' , '*DMSO*')	Set up routine carbon experiment in DMSO-d_6
setup('*C13*')	Set up routine carbon experiment in chloroform

text **Display Text or Set New Text for Current Experiment (C)**

Syntax: text<('string')>

Description: Associated with each experiment is a text file consisting of a block of text that may be used to describe the sample and the experiment. The command *text* by itself displays the text file in the current experiment; *text('string')* allows entry of new text. If more than one line of text is to be added, the double backslash character (\\) is inserted in the text to denote a new line.

Examples:
text
text('Sample 101 in CDCl3\\13 July 1990')

time **Display Experiment Time (M)**

Description: Without any arguments, *time* calculates and displays the approximate acquisition time for the experiment described by the current acquisition parameters. It is roughly accurate for most 1D and 2D experiments; however, for most 2D experiments, it is best to allow ~20% more time for completion of the experiment than the time macro says.

wbs **Specify Action When *bs* Transients Accumulate (C)**

Syntax: wbs('string')

Description: Specifies what action to take when *bs* transients accumulate. The command requires a single argument, a string containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. This command is used once the experiment has started.

See also: *werr, wnt, wexp*

werr **Specify Action When an Error Occurs (C)**

Syntax: werr('string')

Description: Specifies what action to take if an error occurs during acquisition. The command requires a single argument, a string containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. This command is used once the experiment is started.

See also: *wbs, wnt, wexp*

wexp **Specify Action When Experiment Completes (C)**

Syntax: wexp('string')

Description: Specifies what action to take when the experiment completes. The command requires a single argument, a string containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. This command is used once the experiment has started.

See also: *wbs, werr, wnt*

wft **Weight and Fourier Transform 1D Data (C)**

Syntax: wft<('options')>

Description: Performs a Fourier transform on one or more 1D FIDs with weighting applied to each FID. *wft* uses the same arguments as the command *ft*, and except for weighting, it functions the same as the *ft* command.

See also: *ft*

wnt **Specify Action When *nt* Transients Accumulate (C)**

Syntax: wnt('string')

Description: Specifies what action to take when *nt* transients accumulate. The command requires a single argument, a string containing the command or macro to be executed when this event happens. The string must be enclosed in single quotes. This command is to be used once the experiment has started.

See also: *wbs, werr, wexp*

Data Processing

Parameters

fn **Fourier Number (P)**

Description: Number of points to be Fourier transformed (must be a power of two). May be larger than *np* which results in zero-filling the FID or may be smaller than *np* which results in discarding some of the collected data.

Limits: 128 to 512K

lb **Line Broadening and Exponential Weighting (P)**

Description: A positive value gives the desired line broadening in Hz which is then used to calculate an exponential decay of the form *exp(-t*π*lb)*. A negative value gives resolution enhancement (increasing exponential) of the same form. *lb='n'* turns off line broadening. Normal values for proton are 0.1 to 0.5 Hz and for carbon are 1.0 to 3.0 Hz.

lp **Left Phase (P)**

Description: Specifies the left phase-correction angle or first-order phase correction.. Typical values are between -180° and +180°. It is sometimes necessary to reset *lp* and *rp* to zero and reapply *aph* if the spectrum will not autophase correctly.

Limits: -3600° to +3600°

rp **Right Phase (P)**

Description: Specifies the right phase-correction angle or zero-order phase correction. Typical values are between -360° and +360°. It is sometimes necessary to reset *lp* and *rp* to zero and reapply *aph* if the spectrum will not autophase correctly.

Limits: -360° to +360°

sb **Sinebell Constant (P)**

Description: A positive value applies a sinebell of the form *sin(t*π*/(2*sb))* to the raw FID before transformation. A negative value applies the squared sinebell function of the same form. *sb* is given in seconds. Typical value is *sb='n'*.

sbs **Sinebell Shift Constant (P)**

Description: This parameter works with *sb* and allows the origin of the sinebell function to be shifted according to the function *sin(t-sbs)*π/(2*sb))* before apodization. The square of this function is applied if *sbs* is negative. Typical value is *sbs='n'*.

Commands and Macros

aph **Automatically Adjust Left and Right Phase (C)**
Description: Automatically adjusts the phase parameters *lp* and *rp* to produce an absorption mode spectrum.

av **Select Absolute Value Mode in 1D (C)**
Description: Sets the parameter *dmg='av'*. Each point in the displayed 1D spectrum is now the square root of the sum of the squares of the each complex 1D data point. Application of *av* will "get rid of" any dispersive character of a spectrum by making each point positive.
See also: *dmg, ph, ft, wft*

cdc **Cancel Drift Correction (C)**
Description: Turns off drift correction from the *dc* command and resets the spectral drift correction parameters *lvl* and *tlt* to zero.
See also: *dc*

dc **Calculate Spectral Drift Correction (C)**
Description: Turns on a linear baseline correction which uses the edges of the displayed spectrum to denote flat places in the spectrum.
See also: *cdc*

df **Display a Single FID (C)**
Syntax: df<(index)>
Description: Displays a single FID. For an arrayed 1D experiment or for 2D experiments, a particular FID can be displayed by entering the index number.
Examples: *df*
df(4)

ds **Display a Spectrum (C)**
Syntax: ds<(index)>
Description: Displays a single transformed spectrum. For an arrayed 1D experiment of for 2D experiments, a particular spectrum (or slice) can be displayed by entering the index number.
Examples: *ds*
ds(4)

ft **Fourier Transform 1D Data (C)**
Syntax: ft<(index)>
Description: Performs a Fourier tranform on one or more 1D FIDs without a weighting function applied to the FIDs. Transformation of one FID in an array is accomplished by entering the index number.
Examples: *ft*
ft(4)

ph **Select Phased Mode in 1D**
Description: Sets the parameter *dmg='ph'*. Each real point in the displayed 1D spectrum is a linear combination of the the 1D complex data point. The coefficients for this linear combination are derived from the phase parameters *rp* and *lp*. This will produce a spectrum that allows dispersive character and this is the normal mode for 1D spectra.
See also: *dmg, av, ft, wft*

wft **Weight and Fourier Transform 1D Data (C)**
Syntax: wft<(index)>
Description: Performs a Fourier transform on one or more 1D FIDs with weighting applied to the FID. *wft* uses the same arguments as the command *ft*, and except for weighting, it functions the same as the *ft* command.
See also: *ft*

Displaying the Spectrum

Parameters

axis **Axis Label for Displays and Plots (P)**
Description: Specifies the units for the axis display and plot.
Examples: *axis='h'* Hertz
axis='p' ppm
axis='n' no scale

cr **Current Cursor Position (P)**
Description: Lists the current cursor position. The macro *rl* uses *cr* to set the reference line.

delta **Difference of Two Frequency Cursors (P)**
Description: The difference in Hertz between the two frequency cursors. The parameter *delta* is always stored in Hertz but is displayed on the screen in ppm if that is the axis of choice for the display. One may ask for the difference between the cursors in Hertz while *axis='p'* by the following commands:
r1=delta r1? < >

rfl **Reference Peak Position (P)**
Description: The actual position of the reference line in the spectrum, i.e., the distance from the right edge of the spectrum to the reference line in Hertz.

rfp **Reference Peak Frequency (P)**
Description: Sets the frequency of the reference line in Hertz. May be entered in ppm with the *p* suffix.
Examples: *rfp=0* reference line at 0 Hz
rfp=77p reference line at 77 ppm

th **Threshold (P)**
Description: Threshold for the printout of peak frequencies, in mm. Negative peaks greater in magnitude than *th* will also be printed.
Limits: 0 to 1e9

vp **Vertical Position of Spectrum (P)**
Description: Vertical position of the spectrum with respect to the bottom of the display or plotter, in mm.
Limits: -200 to +200

vs **Vertical Scale (P)**
Description: In the normalized mode (*nm*) of spectral display, *vs* is the height in mm of the tallest peak in the spectrum. In the absolute mode (*ai)* of spectral display, *vs* is a multiplier which is adjusted by the user to produce the desired spectral appearance on the screen prior to plotting.
Limits: 1e-6 to 1e9

Commands and Macros

ai **Select Absolute Intensity Mode (C)**
Description: Selects the absolute intensity mode in which the scale is kept constant for one spectrum to the next to allow comparison of peak height for one spectrum to another. It is important to set *ai* for arrayed spectra.
See also: nm

cdc **Cancel Drift Correction (C)**
Description: Turns off drift correction from the *dc* command and resets the spectral drift correction parameters *lvl* and *tlt* to zero.
See also: *dc*

crl **Clear Reference Line (M)**
Description: Cancels the stored reference line by setting *rfl* and *rfp* to zero.

dc **Calculate Spectral Drift Correction (C)**
Description: Turns on a linear baseline correction which uses the edges of the displayed spectrum to denote flat places in the spectrum.
See also: *cdc*

dll **Display Listed Line Frequencies and Intensities (C)**
Description: Displays a list of line frequencies and amplitudes that are above the threshold *th.* The output may be sent to the printer by the following commands:
printon dll printoff < >

dpf **Display Peak Frequencies Over Spectrum (C)**
Description: Displays peak frequencies in *axis* units above the spectrum on the screen. Height of the labels can be changed with *wc2*.
See also: *ppf, dpir, pir, dpirn, pirn*

dres **Measure Linewidth and Digital Resolution (C)**
Description: Measures linewidth at half height and displays linewidth and digital resolution (which is calculated on zero-filled spectrum, not raw FID). *dres* is usually preceeded by *nl* to assure that the cursor is correctly positioned on the desired line.

ds **Display a Spectrum (C)**
Syntax: ds<(index)>
Description: Displays a single transformed spectrum. For an arrayed 1D experiment or 2D experiments, a particular spectrum (or slice) can be displayed by entering the index number.
Examples: *ds*
ds(4)

dscale **Display a Scale (C)**
Syntax: dscale<(#)>
Description: Displays a scale set by parameter *axis* under a spectrum or FID. The scale is placed 5 mm below the position of the spectrum or FID. A single number will place the scale at a different position on the display.
Examples: *dscale* plots a scale defined by parameter axis 5 mm below spectrum
dscale(10) plots a scale 10 mm below spectrum

dsn **Measure Signal-to-Noise (C)**
Description: Measures signal-to-noise for a spectrum. The noise region is specified by placing two cursors on the screen. The tallest peak on the screen is used in the calculation.

f **Display Full Spectrum (C)**
Description: Sets *sp* and *wp* parameters to display the full spectrum.

full **Set Display Limits for a Full Screen (C)**
Description: Sets display parameters so that the entire screen is used for the display. The command string *f full* < > will display the full spectrum on the screen.

nl **Position the Cursor on the Nearest Line (C)**
Description: Moves the cursor to the nearest calculated line position.

nm **Select Normalized Intensity Mode (C)**

Description: Selects the normalized intensity mode in which the spectra are scaled so that the largest peak in the spectrum is *vs* mm high.

See also: ai

r **Recall Display Parameter Set (M)**

Syntax: r#

Description: Recalls the following parameters of the display parameter set #, where # ranges from 1 to 9: *sp, wp, sp1, wp1, sc, wc, sc2, wc2, ho, vo, vs, ai/nm.*

Example: *r2*

See also: *s*

rl **Set Reference Line (M)**

Syntax: rl(frequency)

Description: Assigns the line nearest the cursor as the reference line and assigns it the frequency *rfp*. If a frequency is entered, the macro will assign the reference line that frequency.

Examples:

rl	sets reference frequency to *rfp*
rl(0)	sets reference frequency to 0 Hz
rl(77p)	sets reference frequency to 77*sfrq Hz

s **Save Display Parameters as a Set (M)**

Syntax: s#

Description: Saves a copy of the current values of all display parameters as display parameter set #, where # ranges from 1 to 9. The set has only parameters and no data.

Example: *s3*

See also: *r*

vsadj **Adjust Vertical Scale (M)**

Syntax: vsadj<(height)>

Description: Automatically sets the vertical scale *vs* in the absolute intensity mode *ai* so that the largest peak is the requested height (default is 180 mm).

Example: *vsadj(150)*

Plotting the Spectrum and Printing the Parameters

Parameters

cutoff **Data Truncation Limit (P)**

Description: Defines the distance above and below the current vertical position *vp* at which spectra and/or integrals will not be plotted. *cutoff* does not exist in every parameter set and may be created by the following commands:
create('cutoff') setgroup('cutoff','display') < >
cutoff is not seen on the screen but will work for plotting.

maxpen **Maximun Number of Pens to Use (P)**

Description: Sets the maximum number of pens for a particular plotter. *maxpen=1* must be set for the Zeta plotter.

plotter **Plotter Device (P)**

Description: Sets the plotter for system use.

Options:

'Zeta'	2 page Zeta plot
'Zeta_L'	4 page Zeta plot
'LaserJet_150'	shortwise on 8$_{1/2}$ x 11 paper
'LaserJet_150R'	longwise on 8$_{1/2}$ x 11 paper

printer **Printer Device (P)**
Description: Sets the printer for system use.
Options: *'LaserJet_150'* is equivalent to the '*LaserJet_150R*' orientation given under plotter.

sc **Start of Chart (P)**
Description: Sets the start of plotting (the "chart") with respect to the right edge of the plotter, in mm.
Limits: 0 to wcmax

sp **Start of Plot (P)**
Description: Start of the displayed or plotted region of the spectrum. This parameter is always stored in Hz, but can be entered in ppm with the p suffix.
Examples: sp=100 start of plot is 1000 Hz
sp=0p start of plot is 0 ppm

wc **Width of Chart (P)**
Description: Width of the plotting area (the "chart") in mm.
Limits: 5 to wcmax

wcmax **Maximum Width of Chart (P)**
Description: Maximum width of a plot for a given plotting device.

wp **Width of Plot (P)**
Description: Width of the displayed or plotted region of the spectrum. This parameter is always stored in Hz, but can be entered in ppm with the *p* suffix.
Examples: *wp=1000* width of plot is 1000 Hz
wp = 6p width of plot is 6 ppm

Commands and Macros

page **Send Plotter Output to the Designated Plotter (C)**
Description: This command sends all output in the plotter buffer to the plotter. It begins the actual process of plotting the desired information. To plot a spectrum with a scale and parameter list it would be necessary to type the following commands:
pl pscale pap page < >

pap **Plot Out "All" Parameters (C)**
Description: Plots a parameter list in a set format, listing only the two- or three-character names and the values for all the parameters.

pl **Plot Spectra (C)**
Syntax: pl<(<start,finish,step>)>
Description: Plots one or more spectra.
Examples: *pl* plots spectrum as it appears on the display screen
pl(1,6,2) plots spectra 1 to 6 in steps of 2. It will plot spectra 1, 3, and 5
pl('all') plots all spectra of an arrayed data set; plots a stacked plot as it appears on the screen

pll **Plot Line Frequencies (M)**
Description: Plots the line frequencies in Hz and ppm in column format. Must be used with *page* and will output to the selected plotter.

pltext **Plot a Text File (C)**
Syntax: pltext<('filename')>
Description: Plots a text file. When no filename is given, *pltext* plots the current experiment text file in the upper left hand corner of the page. *pltext* may also be used to send text files to the plotter when a full filename is given.
Examples: *pltext* plots the experiment text file on the spectrum
pltext('textfile') plots the file textfile on the designated plotter

ppa **Plot a Parameter List in "English" (M)**
Description: Plots the most important parameters in an "English-language" format, with a full explanation of each parameter.

ppf **Plot Peak Frequencies Over Spectrum (M)**
Description: Plots peak frequencies in the units of the parameter *axis*. Only peaks with heights greater than *th* will be plotted. The height of these labels can be adjusted with *wc2*.
See also: *dpf, dpir, pir, dpirn, pirn*

printoff **Stop Sending Text to the Printer and Start Print Operation (C)**
Description: Stops the flow of data to the printer and begins the print operation. Flushes the print buffer.
See also: *printon*

printon **Direct Text Output to Printer (C)**
Description: Sends whatever commands follow it to the printer designated by the parameter *printer*.
Examples: *printon dll printoff* < > prints list of line frequencies
printon dg dg1 dgs printoff < > prints data groups
See also: *printoff*

pscale **Plot a Scale (C)**
Syntax: pscale<(#)>
Description: Plots a scale under a spectrum or FID. The default is 5 mm below the spectrum or FID. If a number is given, the scale will be plotted that number of mm below the spectrum.
Examples: *pscale*
pscale(10)

ptext **Print Out a Text File (M)**
Syntax: ptext('filename')
Description: Prints out the text file filename on the printer specified by the parameter *printer*. No *printon* or *printoff* is necessary with this command.

Integration

Parameters

ins **Integral Normalization Scale (P)**
Description: The value selected by the user for the largest integral in the normalized integration mode.

intmod **Integral Display Mode (P)**
Description: Controls display and plotting of the spectral integral.
Examples: *intmod='off'* no integrals displayed or plotted
intmod='partial' every other integral region is plotted (like integral blanking on XL200)
intmod='full' all integral regions are displayed and plotted

io **Integral Offset (P)**
Description: Height that the integral is set above the spectrum, in mm.
Limits: 0 to 200

is **Integral Scale (P)**
Description: An integer multiplier used to adjust the integral scale.
Limits: 1 to 1e9.

Commands and Macros

cz **Clear Integral Reset Points (C)**
Description: Removes all currently defined integral zero points.

dli **Display List of Integrals (C)**
Description: Displays a list of integrals with the defined integral reset points. When *intmod='partial'*, only the integrals corresponding to the displayed integral regions are listed. The output may be sent to the printer by the following commands:
printon dli printoff < >

dlni **Display List of Normalized Integrals (C)**
Description: Displays integrals in a normalized format. The value ins is the value of the sum of all the integrals. When *intmod='partial'*, only the displayed integrals are summed. The output may be sent to the printer by the following commands:
printon dlni printoff < >

dpir **Display Integral Amplitudes Below Spectrum (C)**
Description: Displays integral amplitudes below the appropriate spectral region. *vp=12* must be set first.
See also: *dpf, ppf, pri, dpirn, pirn*

dpirn **Display Normalized Integral Amplitudes Below Spectrum (C)**
Description: Equivalent to the command *dpir* except the sum of the integrals is normalized to the value of *ins*.
See also: *dpf, ppf, dpir, pir, pirn*

isadj **Adjust Integral Scale (M)**
Syntax: isadj<(#)>
Description: Adjusts the integral scale so that the largest integral is # mm high. If no number is given, this defaults to *wc2max*/2.
Example: *isadj(60)*

pir **Plot Integral Amplitudes Below Spectrum (C)**
Description: Plots integral amplitudes below the appropriate spectral regions. *vp=12* must be set first.
See also: *dpf, ppf, dpir, dpirn, pirn*

pirn **Plot Normalized Integral Amplitudes Below Spectrum (C)**
Description: Equivalent to the command *pir* except the sum of the integrals is normalized to the ins parameter.
See also: *dpf, ppf, dpir, pir, dpirn*

z **Add Integral Reset Point at Cursor Position (C)**
Description: Resets the integral to zero at the point marked by the displayed cursor.
See also: *cz*

Storing and Retrieving Data

Commands and Macros

cd Change Working Directory (C)

Syntax: cd<('directoryname')>

Description: Attempts to change the working directory to directoryname. If no directory name is given, *cd* will set the working directory to the user's home directory.

Examples:

cd	Changes to home directory such as /home/user1d
cd('/home/vnmr/maclib')	Changes to given directory

cp Copy a File (C)

Syntax: cp(<'option',>'fromfile','tofile')

Description: Makes a copy of a file by using the UNIX command *cp*. The option '*-r*' will copy a directory recursively. All *.fid "files" are actually directories and must be copied recursively.

Examples:

cp('/home/vnmr1/maclib/hello','/home/user1d/vnmrsys/maclib/hello')

cp('-r','/home/practice_spectra','/home/user1d/practice_spectra')	copies directory recursively

dir List Files in Current Directory (C)

Syntax: dir<('directoryname')>

Description: Lists the files in the current working directory if no directory name is give. Otherwise, *dir* will list the files in the named directory.

Examples:

dir	lists files in current directory
dir('/home/vnmr/maclib')	lists files in named directory

ls List Files in Current Directory (C)

Syntax: ls<('directoryname')>

Description: Lists the files in the current working directory if no directory name is give. Otherwise, *ls* will list the files in the named directory. *ls* is identical to *dir*.

Examples:

ls	lists files in current directory
ls('/home/vnmr/maclib')	lists files in named directory

mkdir Create a New Directory (C)

Syntax: mkdir('newdirectory')

Description: Attempts to create a new UNIX directory. This directory is placed in the current working directory unless an absolute pathname is specified.

Examples:

mkdir('tests')	makes a directory called *tests* in current working directory
mkdir('/home/vnmr1/tests')	makes a directory called *tests* in */home/vnmr1* no matter what the working directory is

mv Move (or Rename) a File (C)

Syntax: mv('fromfile','tofile')

Description: Renames a file or directory.

Examples:

mv('junk.fid','good.fid')	renames *junk.fid* directory as *good.fid* directory

pwd Display Current Working Directory (C)

Description: Displays the pathname of the current working directory.

rm Delete a File (C)

Syntax: rm(<'option',>'filename')

Description: Removes one or more files from the file system, like the UNIX command *rm*. *rm* will allow use of the wildcard character * and the recursive option '*-r*' .

Examples: *rm('-r','junk.fid')* removes *junk.fid* directory recursively from current working directory

rm('-r','.fid')* removes all *.fid* directories recursively from current working directory.....USE THIS WITH CAUTION!!!

rmdir **Remove a Directory (C)**

Syntax: rmdir('directoryname')

Description: Removes an empty directory (only empty directories can be removed).

Example: *rmdir('/home/user1d/trash')* removes the empty directory *trash* from */home/user1d*

rt **Retrieve FID (C)**

Syntax: rt('filename')

Description: Retrieves all FIDs stored in the file *filename.fid* into the current experiment. If *filename.fid* does not exist, and *filename.par* does, it retrieves the parameters only from *filename.par*. If executed without an argument, *rt* will ask for a filename.

Examples: *rt*

rt('lvnmr/fidlib/fidld')

See also: *rtp*

rtp **Retrieve Parameters (C)**

Syntax: rtp('filename')

Description: Retrieves parameters from *filename.par*, if this file exists. If not, and if *filename.fid* exists, it retrieves the parameters only from *filename.fid*. If executed without an argument, *rtp* will ask for a filename.

Examples: *rtp*

rtp('*vnmr/fidlib/fidld*')

See also: *rt*

rts **Retrieve Shim Coil Settings**

Syntax: rts('filename')

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 the *su* command (or other related command). If the filename is an absolute path, *rts* uses it with no modifications. Otherwise, *rts* searches two different directories, as follows: First, it looks for a */home/vnmrsys/shims* subdirectory. If this does not work, *rts* searches for the file in the directory */vnmr/shims*.. If the command cannot find the shim file, it displays the directory names it searched.

Example: *rts*('*acetone*')

svf **Save FIDs In Current Experiment (C)**

Syntax: svf('filename')

Description: Saves all FIDs in current experiment. A file named *filename.fid* is created, which contains the parameters, text, and FID data. *svf* does not remove the data from the current experiment; it merely saves a copy of that data in a different file. The complete data set can be retrieved with the *rt* command; parameters only can be retrieved with the *rtp* command. It is *not* possible to save only one (or several) FIDs from an arrayed experiment; all of them are saved together in one file.

An attempt to store data using a filename that already exists will result in a displayed message.

```
File exists, overwrite (enter y or n <return>)?
```

Examples: *svf*

svf(‘*/home/vnmrsys/fids/username/mydatafile*’)

svp **Save Parameters from Current Experiment (C)**

Syntax: svp('filename')

Description: Save parameters from current experiment. A file named *filename.par* is created, which contains the parameters and text. The parameter set can be retrieved with the *rtp* and *rt* commands. If no filename is given, the command will ask for one.

Examples: *svp*('*/home/directoryname/vnmrsys/fids/username/testdata*')

svs **Save Shim Coil Settings (C)**

Syntax: svs('filename')

Description: Saves all shim coil settings except Z0 into the *filename* given. If the filename is an absolute path, *svs* uses it with no modifications. Otherwise, *svs* will save the shims in */home/vnmrsys/shims*.

Examples: *svs*('*acetone*')

See also: *rts*

Macros for Use on SPARC2 Workstation Only

directory **List and Print Floppy Disk Directory (M)**

Description: This macro will change the working directory to the user's home directory and will print the comtents of the */tmp* directory which is the mount point for the floppy disk.

ejectf **Eject Floppy Disk (M)**

Description: This macro umounts and ejects the floppy disk.

formatf **Format Floppy Disk (M)**

Description: This macro formats a floppy disk to the correct format for the SPARC2.

mountf **Mount Floppy Disk (M)**

Description: This macro mounts the floppy disk at the point */tmp* in the user's home directory.

sizes **List and Print Current Directory Recursively with Sizes of Files (M)**

Description: This macro will list and print the contents of the current working directory recursively and will include the sizes of the files and directories. This is useful in calculating how many .fid directories will fit on a floppy disk or a streaming magnetic tape.

Buttons

readtape **Reads the Contents of Streaming Magnetic Tape to Current Working Directory**

Description: This button will copy the contents of a streaming magnetic tape onto the hard disk in the current working directory. Once this button is activated, no other activity can occur on the workstation until this operation is complete. This button is accessed by the following sequence:

LC Main Menu LC file LC next LC readtape

tapedir **Displays a Directory of a Streaming Magnetic Tape to the Screen**

Description: This button will read a streaming magnetic tape and display the contents of the tape to the screen. This button does not actually copy any of the contents of the tape to the hard disk. Once this button is activated, no other activity can occur on the workstation until this operation is complete. This button is accessed by the following sequence:

LC Main Menu LC file LC next LC tapedir

writetape **Writes the Contents of Current Working Directory to Streaming Magnetic Tape**

Description: This button will write the contents of the current working directory to a streaming magnetic tape and display the contents of the tape to the screen. This button does not distroy any data on the hard disk, it just copies it. Once this button is activated, no other activity can occur on the workstation until this operation is complete. This button is accessed by the following sequence:

LC Main Menu LC file LC next LC writetape